

Sentient Enterprise

For the Windows®, Linux, and Macintosh operating systems

Software Version: 11.00

Reference Guide

Document Release Date: September 2022

Software Release Date: September 2022

Legal Notices

Warranty

The only warranties for products and services are set forth in the express license or service agreements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Persistent Systems shall not be liable for technical or editorial errors or omissions contained herein. The information contained herein is subject to change without notice.

Restricted Rights Legend

Confidential computer software. Valid license from Persistent Systems or its licensors required for possession, use or copying. No part of this manual may be reproduced in any form or by any means (including electronic storage and retrieval or translation into a foreign language) without prior agreement and written consent from Persistent Systems.

Copyright Notice

© Copyright 2022 Persistent Systems Systems, its licensors, and Hewlett-Packard Development Company, LP.

Trademark Notices

Adobe™ is a trademark of Adobe Systems Incorporated.

Android™ is a trademark of Google Inc.

IOS is a registered trademark of Cisco in the U.S. and other countries and is used under license by Apple.

Microsoft® and Windows® are U.S. registered trademarks of Microsoft Corporation.

UNIX® is a registered trademark of The Open Group.

Acknowledgements

This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>).

This product includes cryptographic software written by Eric Young (eay@cryptsoft.com).

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (<http://www.openssl.org/>).

This product includes software written by Tim Hudson (tjh@cryptsoft.com).

This product includes software written by Daniel Stenberg (daniel@haxx.se).

This product includes OVAL language maintained by The MITRE Corporation (oval@mitre.org).

Documentation Updates

The title page of this document contains the following identifying information:

- Software Version number, which indicates the software version.
- Document Release Date, which changes each time the document is updated.
- Software Release Date, which indicates the release date of this version of the software.

To check for recent updates or to verify that you are using the most recent edition of a document, go to:

<https://support.accelerite.com/hc/en-us/articles/9349435630349--Sentient-Endpoint-Manager-Radia-11-0>

This site requires you to register on the Persistent Systems Support Portal. To know more about registration process at Persistent Systems Support portal, visit the [New User Registration Process](#) page.

Support

Persistent Systems Software support provides customer self-solve capabilities. It provides a fast and efficient way to access interactive technical support tools needed to manage your business. As a valued support customer, you can benefit by being able to:

- Search for knowledge documents of interest
- Submit and track support cases and enhancement requests
- Submit enhancement requests online
- Download software patches
- Look up Persistent Systems support contacts
- Enter into discussions with other software customers
- Research and register for software training

To access the Self-solve knowledge base, visit the [Persistent Systems Support home page](#).

Most of the support areas require that you register on the Persistent Systems Support Portal. Many also require a support contract. To register an account at the Persistent Systems Support Portal, visit the [Persistent Systems Support portal](#).

To know more about registration process at Persistent Systems Support portal, visit the [New User Registration Process page](#).

Contents

About the Guide	45
Documentation Map	45
Abbreviations and Variables	46
Administrator Tools	47
Configuring Services Using Advanced Capabilities	47
Configuring Applications in the Machine/User Context	47
Setting the Context of Components (ZCONTEXT)	48
Setting the Service Mode (ZSVCMODE)	49
Enhanced Machine and User Services for Sentient Application Self-service Manager	51
Deploying a Machine or User Service	51
Preparing Versioned Applications	52
Versioned vs. Non-versioned Applications	52
Version Group Editor	53
Creating a Version Group	53
Creating a Version Instance	54
Assigning Version Instances to the Version Group	54
Preparing a Version Group for Deployment	55
Editing a Version Group	56
Version Group (VGROUP) Class	56
Versions (VERSION) Class	58
Application (ZSERVICE) Attributes	58
Reporting Attributes in ZSERVICE	64
Catalog Processing on Agent Machine	65
Creating Internal Users and Groups in Configuration Server Database	65
Creating Users and Groups	66
Assigning Users to Groups	66
Connecting Services to Groups	67
Creating Timers in Configuration Server Database	68
Creating a Timer	68

Scheduling (TIMER) Class Attributes	68
Configuring the Timer	72
Specifying the Timer Expiration (ZSCHDEF)	73
Start Time and End Time Parameters	73
Limit Time Parameter	73
ZSCHFREQ=RANDOM	74
Deploying Applications over a Period of Time	78
Specifying the Command Line (ZRSCCMDL)	78
Connecting the Timer to a Service	79
Testing the Timer Deployment	79
Removing the Timer Object	80
Viewing the Timer Object	80
Experimenting with Timers	80
Timer Logs	81
Sentient Administrator Tools Access for External Users	81
Naming Conventions to Organize Content in the Configuration Server Database	82
Categorizing Information	83
Naming Conventions for the POLICY Domain	83
Naming Conventions for the SOFTWARE Domain	83
Modifying Component Instances(Non-Production only)	84
Accessing the Add, Edit, and Replace Component Commands	85
Editing a Component	85
Using the Editor Selector Dialog Box	86
Replacing Component Data	87
Adding a Component	88
Publishing using Native Packaging	89
Supported Native Package Types	89
Sentient Native Packaging Command-Line Interface	89
Sentient Native Packaging Options File	93
Publishing with Sentient Native Packaging	96
Publishing with Interactive Mode	96
Wrapped Native Packages	97

Automatic Inclusion of Required Packages	100
Publishing and Deploying of Native Packages	101
Publishing	101
Deployment	102
Event Reporting	103
Viewing Event Details	104
Inventory Management	105
The AUDIT Domain	105
AUDIT Domain Defined	105
RIMOPTS Class	107
REGISTRY Class (Applicable for Windows Systems)	110
Software and Hardware Auditing	111
CIM Schema and Inventory Collection	111
Auditing Types	113
File Auditing	113
Audit.FILESCAN	121
WBEM Auditing	122
Manual Scanning Using RIMWBEM	124
NVDCIM.TKD	124
WBEM Object Processing	126
Disabling Remnant Configuration Server instances for WBEM Object Processing	127
Hardware Auditing	127
Configuring Auditing	131
Sample Auditing	131
Configuring a Sample Audit	134
Audit Scanning Processes	135
Audit Scan Results	137
Configuring Timers for Audit Collection	138
ALWAYS The Scheduling (TIMER) Class	138
Creating a Timer Instance	141
Specifying Timer Settings	143

Specifying ZSCHDEF	143
Specifying ZSCHTYPE	144
Specifying ZSCHFREQ	144
Specifying ZRSCCMDL	144
Specifying ZNOPING, PINGDLAY, and PINGCNT	144
Connecting the Timer to a Service	145
Audit Execution Configuration	146
Customizing the RIMOPTS Instance	148
Patch Management	151
Patch Acquisition Process	152
Overview	153
About Patch Descriptor (XML) Files	153
Patch Management Vendor Settings for Microsoft Data Feeds	155
Microsoft Office and Microsoft Update Catalog	155
About Microsoft Automatic Updates	155
About Red Hat Patch Acquisition	156
Creating a Red Hat systemid file	157
Creating Custom Patch Descriptor Files	158
Settings the Manage Installed Bulletins (mib) Option	159
Patch Assessment and Analysis	160
Product Discovery and Analysis	160
Detecting and Managing Microsoft Office Security Bulletins	161
Best Practices for Managing Microsoft Office Security Bulletins	161
Windows Installer 3.1 Requirement	162
Options for Updating Microsoft Office Products	162
When to use Patch Management to Deploy Microsoft Office Updates	163
Client Automation Management Features that are Disabled when using Patch Management	163
Microsoft Update Catalog Supports Office XP, Office 2003, and Office 2007	164
Microsoft Office Service Packs	164
About Patch Management and Microsoft Update Catalog	164

Enabling Microsoft Office Updates in Patch Management (Versions 3.0.2 and later)	165
About Patch Objects Used for Device Compliance Reporting	166
Managing Vulnerabilities	167
Instance Naming Convention for SuSE 10 and 11 Bulletins	168
For SuSE 10	168
For SuSE 11	169
Entitle the FINALIZE_PATCH Service	169
Deploying Automatic and Interactive Patches	170
Customizing Reporting Options	170
Disabling Vulnerability Detection and Deployment	172
Controlling Patch Deployment (PATCHARG)	173
Removing a Patch	173
Supported XML Tags for Patch	174
Descriptor Files	174
Bulletin Node	174
Products Node	176
Product Node	176
Releases Node	176
Release Node	176
Patch Node	177
Patch Signature Node	179
FileChg Node	179
RegChg Node	180
HPFileset Node	181
Patch.cfg Parameters	181
Patch Management Server Configuration Parameters	181
Patch Acquisition Parameters	186
Database Synchronization Parameters	188
Patch Agent Update Parameters	189
OS Management	191
Restoring Operating Systems	191

Pre-requisites	191
Recovering the operating system	191
Deploying OS Images Using CSDB Editor	192
About Policy	193
Assigning OSs to Devices and Groups	193
Advanced Topic: Assigning OSs by Using Policy	193
Advanced Topic: Preparing Content Using the CSDB Editor	195
Logging On	198
About the OS Management Classes	198
About System Behaviors	200
Creating a Manufacturer or Model Instance	204
Assigning Operating Systems	204
Defining Drive Layouts	205
Adding Partitions	220
Assigning Drive Layouts	222
Using an Override Sysprep File	222
Advanced Topic: Configuring ROM Objects for OS Management	224
Prerequisites	225
Syntax	225
Working with the Configuration File	226
Disk Encryption	230
Prerequisites	231
Encryption Support Mode Parameter (ENCMODE)	231
Using Microsoft BitLocker	232
Reserved Space – RSVDSPCE in DRIVEMAP class	232
Local Service Boot and OSM Client Method Updates	233
Partitioning Notes (DRIVEMAP class)	233
Introduction to Hardware Configuration Management	233
Perform a One-Time Hardware Maintenance Operation	234
Using Hardware Configuration Management	235
Using Shadow Hardware Configuration Elements (OS.LME)	235
Creating and Configuring the HWCEs	237

Creating the Hardware Configuration Object (OS.LDS)	240
Viewing the Assigned Hardware Configuration Objects	242
Modifying the Hardware Configuration on a Device	242
Publishing Hardware Configuration Elements	242
About the Hardware Configuration Element Class	243
About the Instance Number and Apply Method Attributes	249
About the Hardware Configuration Class	250
Multicast and OS Management	251
Prerequisites	252
Requirements	252
Configuring Multicast for OS Management	252
Improving Performance and Reliability for Multicast with OS Management	253
Terminology	253
About the Multicast Parameters	254
How the Parameters Influence Multicast Data Transfer	257
Understanding Inter-packet Delay	257
About the Buffer Settings	257
Handling Special Packets	258
Handling the End of Image	258
Auto Throttle	259
Analyzing Problems	259
About the Logs	259
Poor Performance	260
Client Time-Out	261
Total Image Transfer Time-Out	261
Network Inactivity Time-Out	261
Buffer Overflow	262
Slow Client	262
Missing Data	263
Test Modules	264
Using GDMCSEND	264
Using GDMCRECV	267

Customizing OS Deployment by Using Exit Points and Add-Ons	269
User Exit Points	270
Add-On Methods	270
Publishing Add-On Methods	271
Agent Execution of Add-On Methods	271
Agent Execution of Add-On Methods – Important Information	272
OS Deployment Processing Using User Exits	273
Pre-OS Deployment Phase	273
OS Deployment Phase	273
Customizing the Windows Answer File	275
Customizing the unattend.xml File	275
ProductKey	276
Retail Editions	276
Business Editions	276
64-Bit Platforms	277
TimeZone	278
RegisteredOwner and RegisteredOrganization	278
JoinDomain	279
MetaData	280
XML File Processing in Sentient	281
About the .subs	283
Example of Substitution	284
Building a Windows PE Service OS	285
About the Windows PE Service OS Screen	285
Adding Drivers to the Windows PE Service OS	286
About the Custom Build Script	287
Prerequisites	287
Process Knowledge	287
Administrator Machine	288
Media	288
Files and Directories	289
Support for Other Languages	289

Advanced Option	290
Building a Custom Windows PE Service OS	290
Using the Script	291
Get the Script	291
Run the Script	291
Additional Information	294
Using Customized build.config Files (Advanced Option)	295
Using the Sentient OS Manager Deployment Toolkit	296
To Build Custom WinPE Wim Files	296
To Create Image Capture ISO and Image Deploy ISO Files	297
Configuration Parameters in Deployment Toolkit	299
Supported Locales	301
Supported Languages	301
Changing the Locale	301
Setting the System Language Parameter	303
Double-Byte Support for Sysprep or unattend.txt files	303
AppEvents	303
User Messages	306
About the Boot Server	308
Prerequisites	309
Setting Default Service OS	310
Usage Management	313
Usage Data Collection Processing	313
Collection Point Destinations	313
Collection Destination Point Unavailable	313
Agent Processing of Data Collection Request	313
Executable Inventory Scan and Options	314
Initiating Collection of Agent Data	314
Database Performance and Maintenance	314
Using Standard Materialized Views and Filtered Materialized Views	314
Applying the Scripts for SMV or FMV	315
Optional Utility Scripts	315

Miscellaneous Scripts for Oracle	316
Enhancing Reporting Performance	316
Using Views	316
Utility Scripts	317
Miscellaneous Scripts for Oracle	317
Concept of Current Computer	318
Aggregation of Usage Data Files with Tier-Level Store and Forward	318
Aggregation	319
Implementation	319
Configuring Usage Data Files Store and Forward	320
Configuring Aggregation	321
About the Sections in the USAGE.DDA.CFG File	321
Performance	323
Application Usage Manager Agent	323
The USAGE Domain Defined	323
Significance of Collection Instances	324
Filters	325
Filter Criteria Class (UMFLTCRI)	325
Filter Rule Class (UMFLTRUL)	325
Filter Set Class (UMFLTSET)	325
Using Filters	326
Filter Set Class Instances	326
Applying Filters	327
Applying a Filter to a Collection Class	327
Creating Filters	327
Creating a Filter Criteria Instance	327
Creating the Filter Rule Instance	329
Creating the Filter Set Instance	329
Handling Renamed Device	330
Internet Explorer Usage Counts for Windows Vista	330
Setting Search Criteria using AUM Administrator	330
Accessing the Application Usage Manager Administrator	331

Application Usage Manager Admin Search Function	331
Creating Criteria, Rules, Rule Sets, and Rule Set Groups	331
Operators AND versus OR	331
Criteria Tab	332
Creating a Criterion	332
Rules Tab	333
Creating a New Rule	333
Rule Sets Tab	333
Creating a New Rule Set	333
Rule Set Groups Tab	333
Creating a New Rule Set Group	334
Security and Compliance Management	335
Vulnerability Management	335
Compliance Management	336
Security Tools Management	339
How Accelerite Content Network Content is Updated	340
Adding new Products for Security Management	343
Customize the STMLibrary.xml File	343
Replace the STMLibrary.xml File	344
STM Library Schema	344
Product Tags Definitions	345
Operation Tags	346
<Registry>	347
<Process>	348
<RunCommand>	348
<FileProperties>	349
<FileRead>	350
<DirRead>	351
<XMLRead>	351
<String>	352
<Integer>	353
Logical Connector Tags	353

Sample STMLibrary.xml	354
Anti-virus McAfee VirusScan Enterprise 8.8	354
Anti-spyware McAfee VirusScan Enterprise 8.8	357
Microsoft Windows Firewall 7	360
Updating Accelerite Content Network Content Through Command Line	363
Required Settings	363
Optional Settings	364
Stored Settings	365
Examples	366
Move Accelerite Content Network Content from a Test Environment to a Production Environment	366
More Information about Security and Compliance Management	368
OOB Devices Management	369
OOBM Features	369
Discover	369
Hardware Assets	369
Software Assets	369
Heal	370
Remote Operations	370
Event Management	371
Protect	372
System Defense	372
Agent Presence	375
Network Outbreak Containment Heuristics	376
Setting Up vPro and DASH Devices	379
Setting up vPro Device	380
SCS Provisioning of vPro Devices	380
Sentient and SCS on Different Systems	380
Setting Up TLS Mode	380
Setting up SCS Server	386
Provisioning vPro Devices	388
Viewing the vPro Device	394

Changing the Authentication Mode	394
Setting Up DASH Device	395
DASH Configuration	395
OOBM Configuration	396
Information about Configuration Parameters	396
Reconfiguring the SCS Path	396
Reconfiguring Client Automation Web Services	396
Configuring IDE-R Drives	396
Configuring SOL Ports	397
Configuring the SNMP Port	397
Configuring the IDE-R and SOL Time-out Values	397
Configuring Web Service Time-out Value	398
Configuring the Cache Size for DASH Devices	398
Configuring Security Parameters	398
Configuring Watchdog Settings	399
Configuration Settings Used for Debugging	399
Setting Configuration Parameters	399
Configuring Secure Access between OOBM Service and SCS	406
Disabling Secure Access between OOBM Service and SCS	407
To Disable HTTPS	408
Importing the Root Certificate into the Java Key Store	408
To Import the Root Certificate into the Java Key Store	408
Converting Certificates to PEM Format	409
Converting the Root Certificate to PEM Format	409
Converting the Client Certificate to PEM Format	410
Use Case Scenarios	410
Hardware Failure and Replacement	410
Overview	411
Use Case Process	411
Operating System Failure and Reboot	412
Overview	412
Use Case Process	412

Virus Infection Detection and Quarantine	413
Overview	413
Use Case Steps	414
Device Quarantine and Remediation	416
Overview	416
Use Case Steps	417
Monitoring Critical Software	419
Overview	419
Use Case Steps	420
Worm Infection and Containment	426
Overview	426
Use Case Steps	427
Delayed Remote Configuration of the vPro Device	428
Transitioning to Setup Mode	428
Remote Configuration Provisioning Process	429
Delayed Remote Configuration through Sentient Console	431
Trusted Certificates	432
TLS Server Authentication	433
TLS Mutual Authentication	433
RCA Application Manager Agent	435
Sentient Application Management Agent	435
Agent Directories	436
Agent Objects	437
ZCONFIG (Hardware Configuration Information)	438
SYNOPSIS (Client Operation Profile Summary)	442
SAPSTATS (Service Access Profile Status)	444
PREFACE (RADSKMAN Execution)	446
SMINFO (Systems Management Information)	447
Controlling Default Permissions for Directories and Objects	448
Logs	449
RADSTATE (Diagnostic Module)	450
Method Dispatching	452

Lockdown Mode	453
Entitlement Settings	454
Machine Defaults (MACHDEF.EDM)	454
User Defaults (USERDEF.EDM)	455
Directory Structure	455
Directory Structure in Non-Lockdown Mode	455
Directory Structure in Lockdown Mode	456
Features	458
Limitations	458
Retrieving Sentient Agent Information using Sentient WMI	459
VBScript Example - How to retrieve Sentient Agent root path using Sentient WMI provider	459
Messages and Codes	460
Message Types	461
Message Logs	461
Message Categories	462
Receiving a Message	462
Catalog Processing Messages	463
Transmission Error Messages	464
External Data-Download Return Codes (Multicast and Staging Requestor)	465
User Exception Messages	465
Invalid Data Errors	466
Sentient Agent Processing Error Messages	466
User Interface Errors	468
Verification Error Messages	469
Server Errors	470
Method Execution Errors	470
SSL Errors	471
Sentient Internal Error Messages	472
API Errors	476
SAP Error Messages	476
NVDAgent Provider Errors	476

RADSKMAN Command Line Parameters	477
Core Parameters	477
Machine or User Connect parameters	480
Operations	482
Operations Parameters	483
Process parameters	484
RADSKMAN Examples	485
Local AIP Support for the MSI Redirector	488
Resolution Process	489
Configuration Server	495
Configuration Server Management	495
Configuration Server Directories	495
Backing up Configuration Server	496
Configuration Server as a Windows Service	496
Windows Service Options	497
Using the Windows Event Viewer with the Configuration Server	497
Accessing the Event Viewer	497
Filtering for Configuration Server Messages	497
Configuration Server Messages in Event Viewer	498
Tuning the Configuration Server	498
Viewing Configuration Server Settings	500
Editing Configuration Server Settings	500
EDMPROF File	501
MGR_ACCESS	504
MGR_ATTACH_LIST	505
MGR_CACHE	508
MGR_CLASS	511
MGR_DB_VERIFY	514
MGR_DIAGNOSTIC	515
MGR_DIRECTORIES	516
MGR_DMA	519
MGR_ERROR_CONTROL	520

MGR_LOG	521
MGR_MESSAGE_CONTROL	528
MGR_METHODS	529
MGR_NOTIFY	530
MGR_OBJECT_RESOLUTION	532
MGR_POLICY	533
MGR_POOLS	534
MGR_RESOLUTION_FILTERS	537
Example	538
Performance and Usage Considerations	538
MGR_RETRY	538
Example	539
Performance and Usage Considerations	539
MGR_RIM	539
Example	540
MGR_RMP	540
Example	540
MGR_ROM	541
Example	541
MGR_SMTP_MAIL	542
Examples	543
Performance and Usage Considerations	543
MGR_SNMP	543
Example	545
Performance and Usage Considerations	545
MGR_SSL	546
Example	547
Performance and Usage Considerations	547
MGR_STARTUP	547
Example	549
Performance and Usage Considerations	550
MGR_TASK_LIMIT	551

Example	551
Performance and Usage Considerations	552
MGR_TIMEOUT	552
Example	553
Performance and Usage Considerations	553
MGR_TPINIT	553
Example	554
Performance and Usage Considerations	554
MGR_TRACE	554
MGR_USERLOG	558
Performance and Usage Considerations	560
User Log Naming Conventions	560
OBJECT_SIZES	560
Example	561
RCS_TUNING_CONTROL	561
Example	563
SECTION_DELIMITERS	563
Examples	564
Managing Configuration Server Processing	564
Configuration Server Operations	564
Customizing Configuration Server Processing	565
Configuration Server REXX Programs	565
REXX Directories	565
Customizing the Sentient REXX Programs	566
Event Points	566
REXX Programs	567
ZSTARTUP	567
ZPCACHE	567
ZINIT	567
ZTASKSTA	567
ZTASKEND	567
ZNFYxSTA	568

ZNFYxEND	568
ZLOGSWCH	568
ZLOGWRAP	569
ZSHUTDWN	569
Configuration Server Self-Tuning Tool	569
The MGR_CLASS Section	569
The MGR_CACHE Section	570
Reporting Files	571
Sentient REXX Functions	571
EDMGET	572
EDMGETV	574
EDMSET	575
EDMSETV	575
EDMRESO	576
Configuration Server Database Object and non-ASCII Object Names	576
Additional Functions	576
NvdCurrentObjects	577
NvdObjectInfo	577
NvdObjectInfoEX	578
NvdL2U	579
NvdU2L	579
ZCVT and ZTCBG Objects	579
ZCVT Table of Variables	579
ZTCBG Table of Variables	589
Configuration Server Methods	593
Overview	593
The Affects of Configuration Server Methods	594
Method Naming Standard	595
"Must Run" Methods	595
Configuration Server Methods	596
EDMMAILQ	596
EDMMCACH	597

EDMMDB	597
EDMMGNUG	598
EDMMPUSH	599
EDMMPUTD	600
EDMMRPRO	601
EDMMULOG	603
EDMSIGN	604
EDMSIGNR	604
ZDCLASS	605
ZDELINS	606
ZDELOBS	607
ZDELPF	607
ZEXIST	608
ZGETPF	608
ZNFYT	609
ZOBJCMPR	610
ZOBJCOPY	611
ZOBJDELI	611
ZOBJDELV	612
ZOBJSORT	612
ZPROMANY	613
ZPUTHIST	613
ZPUTPF	613
ZSIMRESO	614
ZTOUCH	615
ZVARDEL	615
ZVARGBL	615
ZVARLOG	616
ZUPDPF	616
ZXREF	617
EDM Access Method Services (EDMAMS)	617
Overview	618

Terminology	618
Invoking the EDMAMS Verbs	618
Using the EDMAMS Verbs	619
Input Files	620
EXPORT Verbs	620
Multiple Verbs	621
Wildcards	621
LOGFILE	621
Internationalization Consideration for Exporting/Importing Database Decks	622
EXPORT Verb Considerations	622
IMPORT Verb Considerations	623
Codepage and Locale Defaults	623
ZEDMAMS Utility	624
ADD_FIELD	627
BUILD_PATCH	627
BUILD_STAGING_POINT	627
Resource Naming	628
CHANGE_FIELDNAME	629
CHANGE_FLD_VALUE	629
CHANGE_INST_DATA	630
CHANGE_INS_FIELD	631
CHECK_RESOURCES	631
CLONE_INSTANCE	632
COPY_CLASS	633
COPY_DOMAIN	633
Copying a Domain and its Contents	634
COPY_FIELD	634
COPY_INSTANCE (COPY_RESOURCE)	635
COPY_NEW_SUFFIX	636
CREATE_INSTANCES	636
DELETE_CLASS	637

DELETE_COMP_ORPHS	638
DELETE_DOMAIN	638
DELETE_FIELD	638
DELETE_INSTANCE	639
DELETE_ORPHANS	640
DELETE_RESOURCE	640
EDIT_CLASS_PREFIX	640
EXPORT_CLASS	641
EXPORT_INSTANCE	642
EXPORT_RESOURCE	644
IMPORT_CLASS	645
IMPORT_INSTANCE	646
Verb History	647
Syntax	647
IMPORT_RESOURCE	654
LIST_CLASSES	655
LIST_CONNECTS	655
LIST_CONS_VARS	656
LIST_DOMAINS	656
LIST_FLAGS	656
LIST_INST_DATA	657
LIST_INSTANCE	657
LIST_PACKAGE	658
LIST_PREFIX	658
LIST_RESOURCES	658
LIST_ZRSC_FIELDS	659
MATCH_RESOURCES	659
PACKAGE_UNMATES	660
REFRESH_DMA	660
RENAME_INSTANCE	661
SEARCH_INSTANCES	662
SORT_OBJECT_ID	662

SYNC_CLASS	663
UPDATE_INSTANCES	663
UPDATE_MGRIDS	664
VERIFY_CLASS	665
VERIFY_DATABASE	665
ZRSOURCE_UNMATES	666
Configuration Server Database Utility (RadDBUtil)	667
Running RadDBUtil from a Command Line	668
Implementation Details	668
Sentient Enterprise Patch Manager Considerations	668
IMPORT	669
EXPORT	669
EDMPROF File Settings	669
RadDBUtil Verbs	669
General Syntax	669
VERSION	670
LOG	670
IMPORT	671
EXPORT	675
DELETE	677
RCS	679
Return Codes	679
IMPORT and EXPORT Examples	680
Performing a Simple Import	680
Performing a Simple import and Replacing the Old Instances	680
Importing to a Domain Other Than SOFTWARE	680
An Easy Method to Create Export Files	681
Using the -walk Command	681
Using the -walk and -data Commands	681
Exporting and Importing the PRIMARY File (ZEDMAMS Used to Import the Class)	682
Exporting, Deleting, and Importing a Bulletin	682

The DELETE Verb	682
The Configuration Server Database	683
Configuration Server Performance	683
Performance Issues	683
General Performance and Usage Considerations	684
CPU Requirements	684
CPU and Object Resolution	684
Total Number of Available CPU Seconds	684
Total Number of CPU Seconds Required Per User	685
Total Number of Possible User Resolutions	685
Memory	685
MGR_CACHE	686
MGR_CLASS	687
Networking	687
Bandwidth Throttling	687
Distributed Configuration Server	688
DCS Components	688
Source Component	689
Tcl Web Server	689
Destination Component	689
A Synchronization Pair	690
Configuration Server Eligibility	690
Improving Performance using MANAGER_TYPE	690
Domain Ownership	691
Domain Naming Considerations	691
One Owner vs. Multiple Owners	691
One Owning Configuration Server	691
Multiple Owning Configuration Servers	692
Domain Eligibility	692
Selecting Domains	693
Domain Eligibility Rules	693
Configuring DCS	693

DCS: Batch Mode	694
Synchronization Logs	694
Simultaneous Synchronizations	694
The EDMPROF File and DCS Security	694
Editing the EDMPROF File	694
MGR_STARTUP Section	695
MGR_ID	695
MGR_DMA Section	696
DCS Options and DMABATCH	696
DCS Configuration Files	696
dcs.cfg	696
dmabatch.rc	696
Download	699
Maximum Deletions	700
Configuration Object Equivalents	700
DMASTATS	701
ZUSERID	702
DMABATCH Command-line Options	703
Reset	703
Deferred Commit	703
DCS Objects	704
DCS Files	704
.MK	704
.DAT	704
.IDX	705
DMABATCH Scripting Commands	705
DMABATCH Line Commands	705
Results of DMABATCH	706
Configuration Server Messages	707
Introduction	707
Configuring Configuration Server Log Location and Settings	707
MGR_LOG Section	708

Examples	709
Performance and Usage Considerations	710
MGR_MESSAGE_CONTROL Section	711
MGR_TRACE Section	712
A Sample Configuration Server Activity Log	712
Startup	712
Agent Connect	712
Object Resolution	713
End of Agent Connect	713
Shutdown	713
Viewing the Configuration Server Log	713
Reading the Configuration Server Log	713
Example	714
Message Number	714
Time Stamp	717
Task Name	718
Task Type	719
Message Type	719
Message	719
Configuration Server Messages	720
Numeric Ranges	720
Configuration Server Messages	724
Configuration Server Response Codes	857
Return Codes	857
Reason Codes	858
CSDB Domains and Classes	860
Introduction	860
The Configuration Server Database	860
How to Read the Class Reference Tables	861
The ADMIN Domain	863
Administrator IDs (ADMINID) Class	863
Administrator Types (ZADMIN) Class	864

Attribute Verify (ZVERA) Class	865
Class Verify (ZVERC) Class	866
Connection Rules (ZCONNECT) Class	867
Console Privileges (ZCONUSER) Class	870
Console User IDs (CONUSEID) Class	871
Domain Verify (ZVERD) Class	871
File Verify (ZVERF) Class	872
Function Verify (ZVERFDCI) Class	873
Instance Verify (ZVERI) Class	873
Name Lists (8) (ZLIST) Class	874
Name Lists (32) (ZLIST32) Class	875
The AUDIT Domain	875
Application (ZSERVICE) Class	875
Audit Packages (PACKAGE) Class	888
Behavior Services (BEHAVIOR) Class	890
Client Methods (CMETHOD) Class	896
Desktop (DESKTOP) Class	897
File (FILE) Class	899
File Scanner (FILESCAN) Class	903
File Scanner Filters (FILTER) Class	904
Inventory Options (RIMOPTS) Class	905
Inventory Scanners (SCANNER) Class	906
Path (PATH) Class	908
Registry (REGISTRY) Class	908
Scheduling (TIMER) Class	910
UNIX Permissions (UNIXPERM) Class	915
Virtual Mgr Location (MGRVLOC) Class	915
WBEM (WBEM) Class	916
The CLIENT Domain	918
Alert Management (RADALERT) Class	918
Connect Defer Prefer (CDFCFG) Class	919
Core and Satellite Servers (SERVER) Class	923

Settings (SETTINGS) Class	924
Custom Exception (EXCEPT1-EXCEPT6) Class	932
Diagnostics (DIAGS) Class	934
Hardware Scan Config (RADHWCFG) Class	936
Dynamic Scanning	939
Load Balancers (BALANCER) Class	940
Machine Exception (MACHINE) Class	941
Network Locations (LOCATION) Class	942
Notify Security (NTFYSEC) Class	946
Sentient Extended Security	948
Restricting Notify to Self Notify Only	949
Usage Note	949
Reboot Defer Prefer (RBOOTCFG) Class	949
RepObjs Differencer (REPTDIFF) Class	951
RSM UI Preferences (RADUICFG) Class	952
Server Access Profile (SAP) Class	958
Server Connect Pool (SVRPOOL) Class	964
Subnet (SUBNET) Class	965
Subnet For IPV6 (SUBNET6) Class	966
The MOBILE Domain	967
Android Black List (ABLCKLIS) Class	967
Android Policies (ANDRDPOL) Class	968
Application (ZSERVICE) Class	970
Application Packages (PACKAGE) Class	976
Async Class	978
File Resources (FILE) Class	979
iOS Profile (IOSPRFIL) Class	982
Mail Class	986
MDM (MDM) Class	988
SCEP (SCEP) Class	989
Scheduling (TIMER) Class	990
Settings (SETTINGS) Class	990

VPN Class	992
Wifi Class	993
Windows Profile (WINPOL) Class	995
The PRDMAINT Domain	996
Application (ZSERVICE) Class	996
Application Packages (PACKAGE) Class	1005
File Resources (FILE) Class	1007
Mac File Resources (MACFILE) Class	1012
Maintenance Location (MAINTLOC) Class	1016
Path (PATH) Class	1017
Product Maintenance (PRDMAINT) Class	1018
Scheduling (TIMER) Class	1018
Unix File Resources (UNIXFILE) Class	1023
The PATCH Domain	1027
Patch (PATCH) Class	1028
The PATCHMGR Domain	1029
Application Package (PACKAGE) Class	1030
AUTOPKG (AUTOPKG) Class	1031
vBulletin (BULLETIN) Class	1032
HTTP Codes (CODES) Class	1035
Client Method (CMETHOD) Class	1035
Dialog Services (DIALOG) Class	1036
DOMVER (DOMVER) Class	1039
File Change (FILECHG) Class	1040
File Resources (FILE) Class	1041
HP-US Fileset (HPFSET) Class	1043
Location (LOCATION) Class	1044
Message Catalogs (MSGCAT) Class	1046
Metadata Resources (METADATA) Class	1048
Options (OPTIONS) Class	1051
Patch (PATCH) Class	1051
Patch Arguments (PATCHARG) Class	1054

Patch Auth. Store (PASTORE) Class	1055
Patch Signature (PATCHSIG) Class	1057
Path (PATH) Class	1058
Product (PRODUCT) Class	1058
Product Association (PG2PR) Class	1059
Publisher Error (PUBERROR) Class	1060
Registry Resources (REGCHG) Class	1061
Release (RELEASE) Class	1062
Services Packs (SP) Class	1063
Services (ZSERVICE) Class	1064
The POLICY Domain	1068
Country / Region (COUNTRY) Class	1068
Departments (DEPT) Class	1069
Machine Manufacturer (MANUFACT) Class	1070
Machine Models (MODEL) Class	1070
Machine Roles (ROLE) Class	1071
Machine Subnets (SUBNET) Class	1071
Mobile Device Config (MBLCONFIG) Class	1071
Multicast (MULTICAST) Class	1073
PDACONFIG (PDACONFIG) Class	1076
POLPRMS (POLPRMS) Class	1077
Server Stagers (STAGER) Class	1077
Users (USER) Class	1080
Workgroups (WORKGRP) Class	1082
The SECURITY Domain	1083
Alert/Defer (ALERTDEF) Class	1083
Application (ZSERVICE) Class	1085
Application Control Class (APPCNTRL)	1093
Application Packages (PACKAGE) Class	1095
Auto Run (EXECUTE) Class	1097
Behavior Services (BEHAVIOR) Class	1097
Class Defaults (METACLAS) Class	1102

Desktop (DESKTOP) Class	1103
Dialog Services (DIALOG) Class	1105
File Resources (FILE) Class	1109
HTTP Proxy (HTTP) Class	1115
IBM AIX Packages (AIX) Class	1116
Install Options (INSTALL) Class	1119
Linux RPM Packages (RPM) Class	1122
Mac File Resources (MACFILE) Class	1125
Mobile File Resources (RMMFILE) Class	1129
MSI Basic Resources (MSIBASIC) Class	1132
MSI Features (MSIFEATS) Class	1135
MSI Properties (MSIPROPS) Class	1136
MSI Resources (MSI) Class	1136
Panel Services (PANEL) Class	1141
Path (PATH) Class	1146
Registry Resources (REGISTRY) Class	1147
Replace Application (REPLACE) Class	1151
Scheduling (TIMER) Class	1152
SD Packages (SD) Class	1157
Solaris Patches (SOLPATCH) Class	1162
SVR4 Packages (SVR4) Class	1165
TPM Configuration (TPMCFG) Class	1170
Unix File Resources (UNIXFILE) Class	1171
Version Groups (VGROUP) Class	1175
Versions (VERSION) Class	1177
WTS App Comp Scripts (WTSACS) Class	1178
WTS Control (WTCTRL) Class	1179
The SOFTWARE Domain	1181
Alert/Defer (ALERTDEF) Class	1182
Application (ZSERVICE) Class	1183
Application Packages (PACKAGE) Class	1192
Auto Run (EXECUTE) Class	1194

Behavior Services (BEHAVIOR) Class	1195
Class Defaults (METACLAS) Class	1200
Desktop (DESKTOP) Class	1201
Dialog Services (DIALOG) Class	1203
File Resources (FILE) Class	1207
HTTP Proxy (HTTP) Class	1212
IBM AIX Packages (AIX) Class	1213
Install Options (INSTALL) Class	1216
Linux RPM Packages (RPM) Class	1220
Mac Alias (MACALIAS) Class	1223
Mac File Resources (MACFILE) Class	1224
Mobile File Resources (RMMFILE) Class	1229
MSI Basic Resources (MSIBASIC) Class	1232
MSI Features (MSIFEATS) Class	1234
MSI Properties (MSIPROPS) Class	1235
MSI Resources (MSI) Class	1236
Panel Services (PANEL) Class	1241
Path (PATH) Class	1246
Registry Resources (REGISTRY) Class	1247
Replace Application (REPLACE) Class	1251
Scheduling (TIMER) Class	1252
SD Packages (SD) Class	1256
Solaris Patches (SOLPATCH) Class	1262
SVR4 Packages (SVR4) Class	1265
TPM Configuration (TPMCFG) Class	1270
Unix File Resources (UNIXFILE) Class	1271
Version Groups (VGROUP) Class	1275
Versions (VERSION) Class	1277
WTS App Comp Scripts (WTSACS) Class	1278
WTS Control (WTSCtrl) Class	1279
The SYSTEM Domain	1280
Application Manager (ZCOMMAND) Class	1280

Consoles (ZCONSOLE) Class	1281
DB Version (DBVER) Class	1282
LICENSE (LICENSE) Class	1282
Methods (ZMETHOD) Class	1282
RCA Intent Class (ZINTENT) Class	1283
RCA Processes (PROCESS) Class	1284
RCA Processes (ZPRODUCT) Class	1285
SQLTABLE (SQLTABLE) Class	1285
System Files (ZFILE) Class	1287
Application Management Profiles	1289
Environment Variables	1289
List of Environment Variables	1289
Application Management Profiles Standalone Agent	1291
Policy Management	1293
Adding Client Automation Policy Attributes	1294
Adding the nvObject Class	1295
Modifying classes with nvObject	1295
Connection to LDAP	1296
The Policy Server Configuration File	1296
Support for Multiple LDAP Connections	1296
LDAP Configuration Files	1297
Options for DsConfig Section	1297
Options for DsConfigOverride Section	1298
Policy Section in the Configuration Server Profile	1299
Configuring the LDAP Method	1299
Connecting to the LDAP Method	1301
Policy Scope and Resolution	1302
Managing Policy Scope	1304
Controlling Policy Scope Globally	1304
Optimization for Single-Service Policy Resolution	1305
LDAP	1306

Sentient Policy Server and LDAP	1306
Substitution	1306
Expressions	1307
The LDAP Extension URL Namespace	1308
Domain Filtering	1308
Report Management	1311
Sentient Database Configurations	1311
Sentient Oracle Database	1311
Sentient SQL Server Database	1311
Enhancing Reporting Performance	1312
Using Views	1312
Utility Scripts	1313
Miscellaneous Scripts for Oracle	1313
Reporting Server Configuration Files	1313
Application Management Profiles Configuration (AMP)	1314
Client Configuration Manager Configuration	1315
Configuration Baseline Auditor Configuration	1315
Compliance Management Configuration	1315
DDM Inventory Configuration	1316
Sentient Management Reports Configuration	1317
Inventory Manager Configuration	1317
Mobile Device Configuration	1318
Patch Manager Configuration	1318
Reporting Server Configuration	1319
Security Tools Management (STM) Configuration	1320
Settings Management Configuration	1321
Virtualization Management Configuration	1321
Vulnerability Management Configuration	1321
Application Usage Manager Configuration	1322
LDAP Configuration	1323
Configuring Reporting Server	1324
Normalizing Case Sensitive Databases	1324

Customizing Reports	1325
Extracting Report Packs	1325
Reporting Object Files	1326
Backing up Your Reporting Object Files	1327
Understanding the Reporting Object Files	1327
Reporting Object File Construction	1327
Reporting Object File Types	1327
View Group Objects and View Objects	1327
Filter Group Objects and Filter Objects	1329
Window Objects	1330
Modifying Reporting Object Files	1331
Modifying View Group Objects	1332
View Group Object Global Section	1332
View Group Object Views Section	1332
Modifying View Objects	1333
View Object Global Section	1333
View Object Header Section	1334
View Object Windows Section	1334
Modifying Filter Group Objects	1335
Filter Group Object Global Section	1335
Filter Group Object Filters Section	1336
Modifying Filter Objects	1337
Filter Object Global Section	1337
Windows Toolbar Functions Sub-Section	1338
Modifying Window Objects	1339
Window Object Global Section	1339
Window Object Detail Section	1340
Query Filename and Field ID Sub-Section	1341
Detail Mode Columns Sub-Section	1341
Window Object Graph Section	1343
Generating Reports in CSV Format	1344
Report Name and Window File Name Mapping	1345

Report Filters	1351
Scheduling CSV Format Reports	1355
Exporting Large Reports to CSV Format	1355
RCA Backend Services Management	1357
Portal Services	1357
Portal Architecture	1357
Portal Zone	1358
The Zone Directory Structure	1358
About Object Names in a Zone	1359
Administrative Functions	1360
Setting Configuration Parameters	1360
Configuring for a Custom LDAP Policy Extension Prefix	1362
Customizing Domain Filters (DNAMES) for Policy Resolution	1363
Managing the Portal Zone Directory	1364
Terms for Database Recovery	1364
Managing Management Agent Signal Processing	1364
Managing the Portal Web Services Token	1366
Routing Messages to Portal	1366
Messaging Services	1367
Features and Benefits	1367
Messaging Server Processing on the Sentient Server	1368
Data Delivery Agents	1370
Routing Inventory Data	1371
Creating SQL Tables for the Inventory Database	1371
Using Store and Forward	1371
Configuring and Tuning Messaging Server	1372
Messaging Server Processing	1372
Getting Agent Information	1372
Patch Methods for Collection	1373
ZTASKEND REXX Method	1373
ZTASKEND Calls to QMSG	1374
Processing Phase-Dependent Objects	1374

Processing CORE Objects by Phase	1374
Adding Items to a Critical Object List	1376
Processing Always Objects	1376
Adding Custom Objects to the BuildAlways Object List	1377
QMSG Method Syntax	1377
How Priority Establishes Messaging Server Processing Order	1378
Configuring the Messaging Server	1379
Editing the Configuration Files	1379
RMS.CFG File	1388
Additional Sections in the RMS.CFG File	1389
CORE.DDA.CFG File	1390
ODBC Settings for CORE, INVENTORY and WBEM Objects	1391
PATCH.DDA.CFG File	1392
ODBC Settings for PATCH Objects	1393
Additional Tuning Topics	1393
Configuring the Poll Interval and Post Quantity	1393
Configuring for Retry Attempts	1394
Configuring for Failover	1394
Configuring the Log Level, Log Size and Number	1395
Changing the Logging Level	1395
Changing the Size and Number of Log Files	1395
Configuring the Messaging Server to Discard or Drain Messages	1396
Example: Discarding Messages for the Portal	1396
Example: Draining a Message Queue	1396
Configuring the Messaging Server to Route Portal Messages	1396
About the Portal Data Queue (rmpq) in CORE.DDA.CFG	1396
Restoring Routing for Portal Messages	1397
Enhancing Messaging Server Performance	1398
Disabling Processing of Messages in a Queue	1399
Modifying the Priority in which Messages are Processed	1400
Configuring Messaging Server to Report Differenced Objects	1400
Proxy Server Management	1403

Introduction	1403
Proxy Server Processing	1404
When to use a Proxy Server	1405
Preloader and Desired-State Policy Resolution	1405
Dynamic Cache Management	1406
Proxy Server Configuration File	1407
Sample rps.cfg File After an Installation	1407
RPS.CFG Configuration Parameters	1409
Editing the RPS.CFG File	1413
Syntax Notes	1414
Configuring the Dynamic Cache Parameters	1414
Preloading Deployment Options	1415
Preloading Using Multicast	1416
Preloading using Multicast with Dynamic Windows	1418
Session Logs for a Preload Using Multicast	1419
Preloading Windows Installer Enabled Applications	1419
Model 1: Create one Application (ZSERVICE) that includes a ZSTOP expression on the ACP Package	1419
Model 2: Create separate Applications (ZSERVICES) for MSI Packages and ACP Packages	1420
Configuring HTTPS data download from Linux Proxy	1420
Enhancing Proxy Server Performance	1423
Proxy Server Administration	1424
Starting and Stopping the Proxy Server service	1424
Purging the Proxy Server Dynamic Cache	1424
The Date-Based Purge of the Dynamic Cache	1425
Saving the Index File	1425
Specifying a Size-Based Purge of the Dynamic Cache	1426
SSL Configuration	1429
Configuring SSL	1429
The Certificate Generation Utility	1429
Locating the Certificate Generation Utility	1429

Setting up a Certificate	1429
Using an Existing Private Key	1430
Generating a Signed Certificate	1430
Server Names	1430
Option 1: Generating a Self-Signed Certificate	1430
Option 2: Generating a Re-usable Certificate Signed by a Generated CA ...	1432
Option 3: Generating a Certificate Signed by a Trusted CA	1433
Additional Information about Certificates and Keys	1434
Installing the Private Key File	1434
Command Line Options for the Certificate Generation Utility	1435
Multicast Transmission	1437
Unicasting vs. Multicasting	1437
Multicast Process	1439
Advanced Multicast Configuration Options	1445
MCELIGBL	1445
MCORDER	1446
MWINDOW	1446
BYPASCON	1446
Burst Mode	1447
Using Burst Mode	1447
Parameter Values Selection	1448
Setting DELAYBP and Burst	1448
Testing the Multicast Server	1448
Sentient Agent vs. Administrator Testing	1449
Sentient Agent Testing	1449
Administrator Testing	1449
Notify and TIMER	1449
Network Test Modules	1450
RADCSSEND and RADCRECV	1450
Syntax	1451
RADCRECV	1451
RADCSSEND	1451

BROADCAST Tests	1452
MULTICAST Tests	1453
Implementation and Diagnostics	1454
Preliminary Parameter Calculation	1455
Calculate DELAYBP and MWINDOW	1455
MWINDOW1	1455
Test the Network	1455
Examine the Test Logs	1456
RADCRECV	1456
RADCSEND	1456
Pre-Multicast Session Checklist	1457
Multicast Program-Call Schematic	1457
Smart Card Authentication	1459
Smart Card Login Process	1459
Microsoft Products Supported	1461
We appreciate your feedback!	1467

Chapter 1

About the Guide

The purpose of this guide is to describe the advance capabilities of various Sentient Enterprise features. This information must be used by expert Sentient administrators. You must have the knowledge of configuring and working with Sentient basic features before you customize your environment based on the content described in this guide. Any customizations done without having proper knowledge may have undesirable results in your environment.

You must read the Sentient Core Console Online Help before reading and implementing content explained in this guide.

Documentation Map

The following table lists all documents available for Sentient Enterprise 11.00.

Sentient 11.00 Documentation Set

Guide Name	Description
Sentient Release Notes	Describes what is new in the Sentient current release.
Sentient Installation and Upgrade Guide	Describes step-by-step installation and upgrade procedures for all Sentient components. It also describes the features and components of Sentient.
Sentient Console Online Help	Describes the tasks that you can perform through Sentient Core Console for all Sentient features. After you install the Sentient Core Server, read the online help to configure your environment and use any of the Sentient features.
Sentient System Administrators Guide	This is a replica of Sentient Console Online Help.
Sentient Troubleshooting Guide	Describes various problems that you may encounter in your Sentient environment and their solutions. This guide also describes how to enable trace levels for various Sentient components and respective log files.
Sentient Reference Guide	Describes the advance capabilities of various Sentient features. This information must be used by expert Sentient administrators. You must have the knowledge of configuring and working with Sentient basic features before you customize your environment based on the content described in this guide. You must read the Sentient Console Online Help before reading and implementing content explained in this guide.
Sentient Application Self-	Describes how to use the Application Self-Service Manager on Linux and Macintosh platforms.

Service Manager Online Help (Linux and Macintosh)	
Sentient Application Self- Service Manager Online Help (Windows)	Describes how to use the Application Self-Service Manager on Windows platform.
Sentient Application Manager and Configuration Server REXX Programming Guide	Describes how to create Sentient REXX methods to customize processing for your Sentient-managed computing environment.

Abbreviations and Variables

Abbreviations Used in this Guide

Abbreviation	Definition
RCA	Sentient Enterprise
Core and Satellite	Sentient environment consisting of one Core server and one or more Satellite servers.
CSDB	Configuration Server Database

Variables Used in this Guide

Variable	Description	Default Values
<i>InstallDir</i>	Location where the Sentient server is installed	For a 32-bit OS: C:\Program Files\PSL\RCA For a 64-bit OS: C:\Program Files (x86)\PSL\RCA
<i>SystemDrive</i>	Drive label for the drive where the Sentient server is installed	C:

Chapter 2

Administrator Tools

The *Sentient Enterprise Console Online Help* describes the Sentient Administrator tools for the Windows, Linux, and Macintosh operating systems and how to use these tools to administer the Sentient Enterprise product.

Sentient Administrator contains the following tools that you use to manipulate the contents of the Sentient Configuration Server Database (CSDB), prepare applications for management, view Sentient agent objects, package and publish software into the CSDB, and customize environments:

- Sentient Administrator Agent Explorer
- Sentient Administrator Application Management Profiles (AMP) Editor
- Sentient Administrator CSDB Editor
- Sentient Administrator Publisher

Note: For information on these tools, see *Sentient Enterprise Console Online help*.

Configuring Services Using Advanced Capabilities

This chapter describes how to configure applications in machine and user context. This chapter also describes how to prepare versioned applications to be deployed on agent computers, and describes attributes of the Application (ZSERVICE) instance in Sentient Administrator CSDB Editor.

Configuring Applications in the Machine/User Context

Your enterprise might require that applications be configured to accommodate multiple users, or that one computer always has available the same applications regardless of the user. A service might be considered multi-context if it has components to be installed in the machine context and in the user context.

To complete the installation of a multi-context service, the Sentient agent computer will need to connect to the Configuration Server twice: once to install the machine components and once to install the user components. The machine components are installed first. If the machine portion does not successfully complete, installation of the user component will not take place. The machine portion can be invoked through Notify and timers. The user portion should be embedded in a logon script, desktop shortcut, or batch file that is local to the user.

Complete the following steps to configure your service for machine or user installations.

- Use either Sentient Administrator Publisher or Sentient Administrator CSDB Editor to specify whether a package component is to be installed in the User or Machine context. For more information, see "[Setting the Context of Components \(ZCONTEXT\)](#)" below.
- Use Sentient Administrator CSDB Editor to configure your service to deploy under either the system or user account. For more information, see "[Setting the Service Mode \(ZSVCMODE\)](#)" on the next page.
- Use Sentient Administrator CSDB Editor to set the deployment methods for the service. For more information, see "[Deploying a Machine or User Service](#)" on page 51.

Setting the Context of Components (ZCONTEXT)

Use Sentient Administrator Publisher to select machine and user components separately. Set the context of a component using either Sentient Administrator Publisher during the publishing process, or using Sentient Administrator CSDB Editor after the publishing process.

Note: To take advantage of the Machine/User context, previously packaged applications must be repackaged.

To set the context in Sentient Administrator Publisher:

1. Right-click the files or directories and select **Set Properties**. The **Instance Properties** dialog box opens.
2. In the **Agent Management** tab select the appropriate context for the component.

If you have already completed the publishing process, use Sentient Administrator CSDB Editor to set the ZCONTEXT attribute in the component's instance.

To set the context in Sentient Administrator CSDB Editor:

1. Click **Start > Programs > Sentient Enterprise Administrator > Sentient Enterprise Administrator CSDB Editor**. The **Sentient Administrator CSDB Editor Security Information** dialog box opens.
2. Enter the user ID and password, and click **OK**. The default user ID is `admin`. The default password is `secret`. The Sentient Administrator CSDB Editor window opens.
3. Double-click **PRIMARY**.
4. Double-click **SOFTWARE**.
5. Double-click **Application Packages (PACKAGE)**.
6. Double-click the component's class.
7. Double-click the component.
8. Double-click **ZCONTEXT**.

9. Type the appropriate value based on the information in the following table.

Component Context (ZCONTEXT) options

Setting	Explanation
blank	Leave ZCONTEXT blank if the component is independent of context. This component is installed during machine and user connects.
U	Type U to indicate that this component will be deployed only to the user logged on when the application is initially deployed.
M	Type M to indicate that the file should be deployed to all users of the computer.
User Specified	<i>This option is for future use.</i>

10. Click **OK** to complete the changes.
11. Click **Yes** to confirm the changes and return to Sentient Administrator CSDB Editor.

Note: As a rule, the component will be processed if its ZCONTEXT attribute matches the value of the context parameter in the RADSKMAN command line or if the component's ZCONTEXT attribute is blank.

Now that you have set the component's context, specify that the ZSERVICE instance for the application has machine and user components.

Setting the Service Mode (ZSVCMODE)

The ZSVCMODE attribute in the ZSERVICE class is used to determine if the machine/user context is relevant to the package's deployment. If you need to create the ZSVCMODE attribute, add it to the class template for the ZSERVICE class. Create it as a variable of length 3. Accept the other default properties. The possible values are M, U, MU, blank, and EMU.

Caution: It is recommended that you back up the CSDB prior to making class template changes.

Values for ZSVCMODE in the ZSERVICE Class

Value	Explanation
Blank	Use this when you want the components to install whether the client is logged on in the machine or user context. The application will only be installed using the available context.
EMU	Enhanced Machine/User service: If the Sentient agent connect is being made in the user context, but the machine side of the application has not yet been installed, this will force the machine connect. After the machine connect completes successfully, the user connect is initiated to install the user components. Use this for optional applications that the user controls through

Value	Explanation
	Sentient Application Self-service Manager.
EMU:AD=N	Enhanced Machine/User service: If the Sentient agent connect is run in the user context, but the machine side of the application are not yet installed, this will force the machine connect. After the machine connect completes, the user connect is initiated to install the user components. Use this for optional applications on a shared computer that the user controls through the Application Self-service Manager. The addition of AD=N prevents a user from being able to remove the machine components of the application. Other users of the agent computer may still need the machine components of the application. The valid application events are: AI = application install AD = application delete AU = application update AR = application repair VA = version activation VD = version deactivation The default for each event is Y. Separate multiple events with a comma.
M	Machine service only Set ZSVCMODE to M if the service has only machine components. This service will be ignored if context is set to u on the RADSKMAN command line.
MU	Machine/User service Set ZSVCMODE to MU if the service has both machine and user components. The user connect will verify that the machine components have been installed before installing the user components. The user components will not be installed if the machine components are not present.

To set ZSVCMODE in ZSERVICE:

1. Click **Start > Programs > Sentient Enterprise Administrator > Sentient Enterprise Administrator CSDB Editor**. The **Sentient Administrator CSDB Editor Security Information** dialog box opens.
2. Enter the user ID and password, and click **OK**. The default user ID is `admin`. The default password is `secret`. The Sentient Administrator CSDB Editor window opens.
3. Double-click **PRIMARY**.
4. Double-click **SOFTWARE**.
5. Double-click **Application (ZSERVICE)**.
6. Double-click the appropriate service.
7. Double-click **ZSVCMODE** in the list view.
8. Type the appropriate values based on the values discussed in ["Values for ZSVCMODE in the ZSERVICE Class" on the previous page](#).
9. Click **OK** to complete the changes.
10. Click **Yes** to confirm the changes and return to Sentient Administrator CSDB Editor.

Enhanced Machine and User Services for Sentient Application Self-service Manager

When an application has machine and user components, Sentient Application Self-service Manager agent needs elevated privileges on the Sentient agent in order to properly install machine components, and the user components need a user logged on to access the user's settings. A machine component might be a file or registry key while a user component might be a desktop shortcut. If context is set to **U**, the user will not be able to install the machine side of the application, and the user connect will fail. By setting ZSVCMODE to **EMU**, when the user selects the application to install, Sentient Application Self-service Manager agent will check to see if the machine components are already installed. If they are not, Sentient Application Self-service Manager agent will install the machine components *for that service only*, and then, install the user components. In other words, two separate Sentient agent connects will run, one in machine context and one in user context. Set the CONTEXT tag to **U** in the `args.xml` file to enforce EMU behaviors. `args.xml` is the Application Self-Service Manager arguments file created from information stored in the [ARGS] section of `Install.ini`. This file is stored in `IDMLIB` directory, `<InstallDir>\Agent\Lib\`, on the Sentient agent computer and controls the behavior of the Sentient Application Self-Service Manager. For more details about the `args.xml` file, see *Sentient Enterprise Installation and Upgrade Guide*.

Note: ZSVCMODE must be changed to 3 bytes and EMU must be one of the choices.

If ZSERVICE.ZSVCMODE is set to **EMU**, and Sentient Application Self-service Manager agent encounters a service that is marked as EMU, it first checks to see if the machine catalog exists, and if the requested service exists in the machine catalog:

- If the service exists, and is marked installed in the machine catalog, the service is installed for the user.
- If the service is not installed on the machine side, but the catalogs are synchronized, then the client will first install the machine side of the service and then install the user piece of the service.
- If the machine catalog is missing (because a machine connect never ran), then the client gets the machine catalog, and installs the service on the machine side. Then, if the service was successfully installed on the machine side, the user side of the service will be installed.

Deploying a Machine or User Service

The Sentient agent will need to connect to the Configuration Server twice to complete the installation of services that are marked with ZSVCMODE set to **MU**. The first connect will be in the machine context. In the RADSKMAN command line, add a parameter `context=M`. This will set the `startdir=SYSTEM` by default. Do this by using either a Timer or a Notify command.

The second connection will install the user components. Do this in a logon script, batch file, or desktop icon since the user needs to be logged on. Use the RADSKMAN command line with `context=U` parameter. If ZSVCMODE is **MU**, the user components will only be installed if the machine connect has finished successfully.

Preparing Versioned Applications

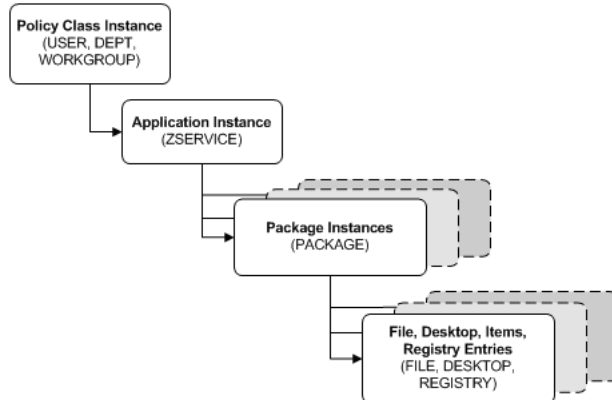
Typically, when you deploy an application to a Sentient agent computer, using either Sentient Scheduler or Sentient Notify, it is activated immediately. There is another option, **Version Groups**, which enables you to roll out a new version of an application to users; it gives you the options of having it activate upon delivery or at a pre-determined time. If the installation of the new version fails, Sentient will automatically roll back to the previous version. If problems occur in the new version after installation, you can deactivate the new version and roll back to the previous version for some, or all, users.

After versioning is configured, the compressed files are stored on the Sentient agent computer, and the versioning takes place. The roll forward/roll back can be entirely local, not requiring any data to be transferred at the version change time. It can also be configured to be partially local, with a minimum of data being transmitted.

Versioned vs. Non-versioned Applications

Versioned and non-versioned applications adhere to different connection models within the CSDB. For non-versioned applications, one Application instance (ZSERVICE) connects to one or more package instances.

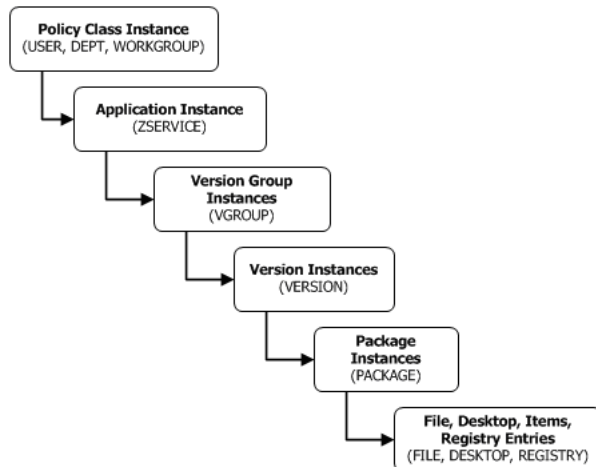
CSDB path for non-versioned application deployments



Versioned applications adhere to a different connection model than non-versioned applications. For versioned applications, an Application instance (ZSERVICE) connects to a single Version Group (VGROUP) instance.

Note: If you want to use multiple Version Groups, you must create one service for each Version Group.

The Version Group instance connects to one or more Version instances that connect to one or more Package instances. A Version instance (which represents one version of a software application) contains one Sentient package. Each Sentient package is represented in the CSDB by an instance of the PACKAGE Class.

CSDB path for deployments of versioned application

To prepare versioned applications:

1. Use Sentient Administrator Publisher to package the application.
2. Right-click **ZSERVICE** class and select **New Instance**.
3. In the **Create Instance** dialog box, type a display name and an instance name, and click **OK**.

To finish creating the Version Group, see the following section, "[Version Group Editor](#)" below.

Version Group Editor

Use Version Group Editor, in Sentient Administrator CSDB Editor, to create, edit, and delete instances for each version of an application, as well as manage the deployment of a version group. A version group contains all versions of an application.

To manage a versioned application, in the Version Group class create an instance that represents the set of versions for the application. Then, use the Version Group Editor to create instances for each version of the application.

Creating a Version Group

In the following example, Sentient Administrator CSDB Editor is used to create a new instance in the Version Group (VGROUP) class.

To create a Version Group:

1. Click **Start > Programs > Sentient Enterprise Administrator > Sentient Enterprise Administrator CSDB Editor**. The **Sentient Administrator CSDB Editor Security Information** dialog box opens.
2. Enter the user ID and password, and click **OK**. The default user ID is `admin`. The default password is `secret`. The Sentient Administrator CSDB Editor window opens.
3. Navigate to **PRIMARY.SOFTWARE**, right-click **Version Group (VGROUP)**.
4. Select **New Instance**. The **Create Instance** dialog box opens.

5. In the text field of the **Create Instance** dialog box, type a name (such as Amortize) for the Version Group, and click **OK**. The **Editing Version Group** dialog box opens.

Creating a Version Instance

Now that you have created a Version Group (VGROUP) instance, it is necessary to learn how to create an instance for each version of your application.

To create a version instance:

1. In the Version Group Editor, click **Add**. The **Create Version** dialog box opens.
2. Type a suffix that identifies the version. For example, `Version 1`.
3. Click **OK**. The **Version Editor** dialog box opens.
The Version Editor dialog box contains a list of Application Package (PACKAGE) instances that are stored in the Configuration Server Database. Use this dialog box to connect the new Version (VERSION) instance to an Application Package (PACKAGE) instance. There is a one-to-one correspondence between these two instances.
4. Click the appropriate Application Package (PACKAGE) instance.
5. Click **OK**. The Version instance appears in the **Unassigned Version** list.
Add a Version instance for each version of the application that will be available to subscribers through this Version Group.

To delete a version instance:

1. In the **Version Group Layout** list, select the version that you want to delete.
2. Click **Delete**.
The instance for the version appears in the **Unassigned Versions** list. The icon is in a dimmed state, ready for deletion.
To restore the instance, click the instance in the **Unassigned Versions** list and then click **Un-delete**.
The version instance will not be deleted until you close the Version Group Editor.
3. Click **OK** to close the Version Group Editor.

Assigning Version Instances to the Version Group

After creating your Version instances, you must assign them to the Version Group.

To assign Version instances to the Version Group:

In the **Unassigned Versions** list, click a Version instance and drag it over a connection labeled <no assigned version> in the Version Group Layout list.

Assign each of the Version instances that you created to the Version Group.

To remove a Version instance assignment:

In the Version Group Layout, click a Version instance and drag it to the Unassigned Versions area. Then, release the mouse button.

Preparing a Version Group for Deployment

Now that you have created the Version instances and assigned them to your Version Group, you are ready to specify how you want to deploy the versions. Use the Version Group Information area in the Version Group Editor to define the deployment of the versions.

To prepare a Version Group for deployment:

1. In the Versioning Controlled By drop-down list, select:
 - **Manager** if you want to control the version to be deployed.
 - **Client** if you want the user to control the version to be deployed.
This is used with Sentient Application Self-service Manager only.

Note: If you want the ability to schedule the version deployments with the Configuration Server you must select **MANAGER** from the **Versioning Controlled By** drop-down list.

2. In the **Initially Active** drop-down list, select the version that you want to activate on the Sentient agent computer the next time the user connects to the Configuration Server. You can also select from the versions that appear in the **Version Group Layout** list. The selected version appears in bold text in the **Version Group Layout** list, as shown in the next figure.
3. Select the **Activate on or After** check box to access additional controls that can be used to delay activation of a version until a specific date and time.

Note: If you select **MANAGER** in the **Versioning Controlled By** drop-down list, you must select the **Activate On or After** check box so that the Configuration Server knows when to activate the next version.

4. In the **Next Version to Activate** drop-down list, select the version of the application that you want to activate after the initial version.
5. In the **Time (hh/mm)** drop-down list, select when you want the version to be activated.
6. Use the Calendar controls to set the date of deployment for the next version.

Caution: If you use the Time and Calendar controls to schedule the deployment of a version, consider the following:

- You can schedule deployments of versions only if the Configuration Server controls the versions
- If you selected **MANAGER** in the **Versioning Controlled By** drop-down list, you must select the **Activate On or After** check box to let the Configuration Server know when to activate the next version.
- If you delete a VGROUP instance, the associated TIMER instance will be deleted.

7. Click **OK** to save the information in the Version Group Editor.
8. Click **Yes** to confirm your changes.

The Version Group instance appears in the Version Groups (VGROUP) Class. If you scheduled the next version to activate, Sentient creates an instance in the Scheduling (TIMER) Class and automatically connects the timer to the Version Group.

To connect the Version Group to the Service:

1. In Sentient Administrator CSDB Editor, navigate to **PRIMARY.SOFTWARE.ZSERVICE**.
2. Right-click the appropriate service (such as Amortize) and select **Show Connections**. The **SOFTWARE.ZSERVICE Connections** dialog box opens.
3. Click **Version Groups (VGROUP)** and click **OK**.
The Version Group instances appear in the list view of Sentient Administrator CSDB Editor.
4. Click **Amortize** in the list view and drag it to the appropriate Application (ZSERVICE) instance (in this example, Amortize). When your cursor changes to a paper clip, release the mouse button. The **Select Connection Attribute** dialog box opens.
5. Click **Copy**.
6. Click **Yes** to confirm that you want to connect the Amortize Version Group to the Amortize service.
7. Click **OK** when you receive a confirmation message.

Note: Be sure to connect the ZSERVICE instance to the POLICY instance for the subscribers to whom you want to deploy this.

The next time the Sentient agent connects to the Configuration Server, the initial version of the application is activated and the compressed files for the next version will be stored on the Sentient agent computer.

Editing a Version Group

After you create a version group and its instances, you might want to return to the Version Group Editor to make changes.

To edit a Version Group:

1. In Sentient Administrator CSDB Editor, navigate to the Version Group instance located in **PRIMARY.SOFTWARE.VGROUP**.
2. Right-click the appropriate Version Group instance and select **Version Group Editor**. The Version Group Editor opens.
3. Edit the Version Group as necessary.
4. Click **OK** to save your changes.
or
Click **Cancel** to close the Version Group Editor without saving your changes.

Version Group (VGROUP) Class

Each instance of the Version Group (VGROUP) Class defines a set of versions for an application, and contains connections to the Versions (VERSION) Class. The following table describes the

VGROUP Class attributes.

Version Group (VGROUP) Class Attributes

Attribute	Description
ZSTOP00n	In ZSTOP attributes, expressions that evaluate to true cause resolution of the instance to be skipped. If left blank, the instance is not skipped and resolution continues. This is useful for assigning a version to a set of users. Use Sentient Administrator CSDB Editor to set this attribute.
CONTROL	Indicates whether the Sentient administrator (MANAGER) or the user (CLIENT) controls which version to activate on the Sentient agent computer. Use the Versioning Controlled By drop-down list in the Version Group Editor to set this option. Note: Sentient agents support Sentient administrator-controlled version activation, but do not support subscriber-controlled activation.
INITIAL	Indicates which version to activate on the Sentient agent computer. Use the Initially Active drop-down list in the Version Group Editor to set this option.
ROLLBACK	Indicates whether to automatically roll back to the previously activated version when deployment of a new version fails. The default is <i>Y</i> .
REQACTDT	The earliest date on which a version in this Version Group will be activated on any Sentient agent computer. If this attribute is blank, the version identified by the INITIAL attribute will be activated at the end of the Sentient agent connect that causes the version to be transferred to the Sentient agent computer. Use the calendar controls in the Version Group Editor to set REQACTDT.
REQACTTM	The earliest time, on the date specified by the REQACTDT attribute, after which a version in the Version Group will be activated on any Sentient agent computer. The version identified by the INITIAL attribute will be activated during the next Sentient agent connect. Use the Time (hh/mm) drop-down lists in the Version Group Editor to set REQACTTM.
NAME	The friendly name for the VGROUP instance. This is set when you create the instance using Sentient Administrator CSDB Editor.
ACTDATE	This data is set and maintained, by the Sentient agent, in the VGROUP object on the Sentient agent computer. <i>Do not alter its value.</i>
STATUS	This data is set and maintained, by the Sentient agent, in the VGROUP object on the Sentient agent computer. <i>Do not alter its value.</i>
CURVERS	This data is set and maintained, by the Sentient agent, in the VGROUP object on the Sentient agent computer. <i>Do not alter its value.</i>
NEXTVERS	This data is set and maintained, by the Sentient agent, in the VGROUP object on the Sentient agent computer. <i>Do not alter its value.</i>
SOURCE	This data is set and maintained, by the Sentient agent, in the VGROUP object on the Sentient agent computer. <i>Do not alter its value.</i>

Attribute	Description
TIMERCON	If you specify a “next version to activate,” Sentient Administrator CSDB Editor automatically creates a timer and stores the connection to that timer in this attribute.
VERCON0n	Connects to each version in the Version Group. Each VERCON0n attribute contains a connection to one instance of the VERSION Class. This is set when you assign a version to the Version Group in the Version Group Editor.

Versions (VERSION) Class

Each instance of the VERSION Class defines one version of an application to be deployed and managed by Sentient. Use the Version Group Editor to create VERSION Class instances and assign them to a Version Group. The following table describes the VERSION Class attributes.

Versions (VERSION) Class Attributes

Attribute	Description
ZSTOP00n	In ZSTOP attributes, expressions that evaluate to true cause resolution of the instance to be skipped. If left blank, the instance is not skipped and resolution continues. This is useful for assigning a version to a set of users. Use Sentient Administrator CSDB Editor to set this attribute.
NAME	The friendly name for the VERSION instance. This is set when you create the instance using Sentient Administrator CSDB Editor.
PACKAGE	Connects to a PACKAGE Class instance, which represents the packaged software for this version.

Application (ZSERVICE) Attributes

This section describes the attributes of the Application (ZSERVICE) instance in Sentient Administrator CSDB Editor. Many of the values for these attributes are set when using the Sentient Administrator tools and the New Application Wizard in Sentient Administrator CSDB Editor. You can also use Sentient Administrator CSDB Editor to modify the values of these attributes in the SOFTWARE.ZSERVICE class.

You might notice that some attributes do not have values, or their values are not displayed in Sentient Administrator CSDB Editor. The Sentient agent uses these attributes.

Modifiable SOFTWARE.ZSERVICE Attributes

Attribute	Description
ZSTOPnnn	Stops resolution if the expression evaluates to TRUE. Example: WORDPOS (EDMGETV (ZMASTER, ZOSTYPE) , 'WIN32_NT WIN64_NT') =0This example expression will stop resolution (the application will not be installed) on the instance if the Sentient agent's operating system is neither Windows 32-bit nor Windows 64-bit.
ZSVCNAME	Use this attribute to set the name of the service that will display in Sentient

Attribute	Description
	Application Self-service Manager interface. This value is initially set in the Short Description field in the New Application Wizard.
ZSVCTTYP	This indicates for which Sentient agent this application was packaged. - Specify A for Sentient Application Manager. - Specify S for Sentient Application Self-service Manager. This value is initially set in the New Application Wizard.
ZSVCMO	Use this attribute to designate a service as mandatory or optional. - When using Sentient Application Manager, services are typically designated as mandatory (M); - When using Sentient Application Self-service Manager, services are typically designated as optional (O). If both Sentient agents are being used you could also specify mandatory then optional (ZSVCMO=MO), or optional then mandatory (ZSVCMO=OM). - The first character indicates how the application should be handled before installation. - The second character indicates how the application should be handled after installation. Note: You might need to edit the ZSERVICE class template to allow the ZSVCMO=OM setting. For information on editing a class, see <i>Sentient Enterprise Console Online Help</i>. To process mandatory applications using Sentient Application Self-service Manager, add <code>enterprisemanagement=auto</code> to the <code>args.xml</code> file. This value is initially set based on the setting for the application target type (ZSVCTTYP) in the New Application Wizard.
ZSVCPRI	Use this attribute to set the priority level for the service. Services are created based their priority. The lower the number, the higher the service's priority. - A service with ZSVCPRI=01 would have the highest priority. - A service with ZSVCPRI=99 would have the lowest priority.
ALWAYS	Any method that you specify for this attribute is unconditionally executed when this instance is resolved.
ZCREATE	A method that runs when the service is installed.
ZINIT	A method that runs when the service is initialized.
ZDELETE	A method that runs when the service is deleted.
ZUPDATE	A method that runs when the service is updated.
ZVERIFY	A method that runs when the service is verified.

Attribute	Description
ZREPAIR	A method that runs when the service is repaired.
PUBDATE	<i>Reserved for future use.</i>
UPDDDATE	<i>Reserved for future use.</i>
AUTHOR	The name of the author of the service. This appears in the extended information area in Sentient Application Self-service Manager interface. This value is initially set in the Author field in the New Application Wizard.
DESCRIPT	A description of the service. This appears in the properties for the service in the Service List. This value is initially set in the Long Description field in the New Application Wizard.
VENDOR	The name of the vendor of the service. This appears in Sentient Application Self-service Manager interface. This value is initially set in the Vendor field in the New Application Wizard.
URL	The web address where the subscriber can find additional information about the service. This appears in the properties for the service in Sentient Application Self-service Manager interface. This value is initially set in the Web URL field in the New Application Wizard.
CATGROUP	Use this attribute to group a set of applications. You can display applications based on their group in Sentient Application Self-service Manager interface.
PRICE	Specify the price of an application. This will be displayed to subscribers in the extended information area in Sentient Application Self-service Manager interface.
SCHEDOK	<i>For Sentient Application Self-service Manager only.</i> - Specify Y to allow the user to locally change the update schedule. - Specify N to retain control on the Configuration Server.
VERSION	The version of the software. This appears in the properties for the service in Sentient Application Self-service Manager interface. The value is initially set in the Version field in the New Application Wizard.
NAME	This name appears in the properties for the service in Sentient Application Self-service Manager interface. The value is initially set in the Short Description field in the New Application Wizard.
OWNER	<i>Reserved for future use.</i>
RUNDLG	Specify Y to enable the processing of DIALOG Class instances during the installation of the service; specify N (the default) to disable this processing.
REBOOT	This attribute is used to restart the Sentient agent computer based on an application event. Specify the action by equating an application event to a reboot type, panel, or connect. Event on which to Restart: AI = Install; AD = Deinstall; AU = Update; AR = Repair; AV = Verify Type of Panel; Q = No panel;

Attribute	Description
	A = OK button only; Y = OK and Cancel buttons. Type of Reboot: S = Soft reboot (default of type Y panel); H = Hard reboot (default of type A panel); N = No reboot Type of Connect: None specified = Reboot on machine connect (context=m); U = reboot on user connect (context=u); MU = reboot when machine and user parts of the service have been installed.
EVENTS	Set this attribute to indicate on which events to report. Specify your event by equating an application event to an event type. AI=Application Install; AD=Application Deinstall; AU=Application Update; AR=Application Repair; AV=Application Verify; VA=Version Activation; VD=Version Deactivation; UM=Upgrdmaint Maintenance What to Report on: S=Success; F=Failure; B=Both (success and failure); N=None The default is AI=B, AD=B, AU=F, AR=N, VA=F, VD=F, UM=B
ERTYPE	Set this attribute to send an APPEVENT object. Currently, this supports the Object format only, so the default is ERTYPE=O.
ADAPTIVE	Set this attribute to indicate whether the installed package is dependent on Sentient agent settings that must be monitored periodically. - Specify Y for Yes. - Specify N for No. If the settings change, the Sentient agent must reconnect to the Configuration Server to get new or different components.
LREPAIR	Set this attribute to enable local repair of broken applications. If an application is broken due to missing files, the files (locally stored) can be used to repair the application. - Specify Y for Yes. - Specify N (the default) for No.
REMOVAL	This attribute controls how the application is managed when a service is removed. - REMOVAL=A (Abandon) will delete the service's objects on the Sentient agent, but leave its components. The service will no longer be managed by Sentient. - REMOVAL=D (Delete) will delete the service's objects and components. The service will still be managed by Sentient. - This is the default. - REMOVAL=U (Unmanage) will stop management of the service by Sentient. Neither the objects nor the components will be deleted. This applies only to optional applications (ZVSCMO=O) that are removed based on entitlement policy. Note: If a user removes an optional application, the service's objects are also removed, regardless of the REMOVAL setting.

Attribute	Description
RECONFIG	Set this attribute to indicate whether an application can be relocated after it has been installed. For example, you can move an application from the C drive (on which it was installed) to the D drive without having to remove and re-install the application. - Specify Y for Yes. - Specify N for No.
ZSVCCAT	Set this attribute to indicate whether the service is visible in Sentient Application Self-service Manager catalog. For optional applications, the default is Y; for mandatory applications, the default is N. These defaults can be overridden.
UIOPTION	Set this attribute to indicate whether the Service Status window will be displayed. Possible values are: UIOPTION=NONE: No interface displayed. UIOPTION=FULL: Interface displayed; Cancel button is available. UIOPTION=INFO: Interface displayed; no Cancel option.
UIOPTMSI	This attribute provides control over the MSI status window, to the user. The possible values are: UIOPTMSI=NONE: No interface displayed. UIOPTMSI=FULL: Displays what is happening on the agent computer and provides the subscriber with controls to make changes.. UIOPTMSI=INFO: Displays what is happening on the agent computer, but disables all the controls so that the subscriber cannot make any changes. See the Setup log files and Event Viewer for warnings, errors, successes, failures, and other details.
CACHE	This attribute enables element caching. - Specify Y for Yes. - Specify N (the default) for No.
CACHELOC	<i>For Windows Installer applications only.</i> Specify the location of the folder, on the Sentient agent computer, that is used to cache the compressed application files that are needed for the product. The default is _UNDEF_. Sentient support for Windows Installer tags the PRODGUID value to this value in order to create the folder. For example, if CACHELOC=C:\progra~1\PSL\Sentient\Agent and PRODGUID=12345_XXXX, the cache folder would be C:\progra~1\PSL\Sentient\Agent\12345_XXXX\cache. Note: The folder \cache is automatically appended to PRODGUID. If you are not deploying a Windows Installer-enabled application, the files will be cached in IDMDATA.
CACHELIM	<i>For Windows Installer applications only.</i> Specify a number between 000 and 100 to indicate the cache limit—defined as “the percentage of used drive

Attribute	Description
	space.” If the percent of used space is greater than the cache limit, all of the product’s cached files are removed and the cache folder is deleted. This is checked after every file is cached on the disk.
ZDISCONN	Set this attribute to indicate whether the Sentient agent is allowed to disconnect from the Configuration Server if there is an open session with the Configuration Server. - Specify Y to disconnect the Sentient agent from the Configuration Server. - Specify N (the default) to keep the Sentient agent connected to the Configuration Server.
ZSYSACCT	Set this attribute to indicate whether to install the service under the system account or the user’s account. - Specify Y to install the application using the system rights. - Specify N (the default) to install the application using the rights of the user that is logged on.
MCELIBGL	Indicates whether the application is eligible for multicasting. - Specify Y (the default) for Yes. - Specify N for No.
RSTRSIZE	Use this attribute in the appropriate ZSERVICE Class instance to control which files are enabled for check-point restart based on the amount of data (in bytes) that are being downloaded.
ZSVCMODE	Set ZSVCMODE to: M if the service has machine components only. This service will be ignored if <code>context=u</code> on the RADSKMAN command line. U if the service has user components only. This service will be installed if <code>context=u</code> or is left blank on the RADSKMAN command line. You might want to use this setting if the application consists only of user registry changes or user desktop shortcuts. MU if the service has machine and user components. The user connect will verify that the machine components have been installed before installing the user components. You will need to run two RADSKMAN connects—one with <code>context=m</code> and one with <code>context=u</code>. EMU if the Sentient agent connect is being made in the user context but the machine side of the application has not yet been installed, because this will force the machine connect. After the machine connect completes, the user connect is initiated to install the user components. Use this for optional applications that the user controls through Sentient Application Self-service Manager. Leave ZSVCMODE blank to treat the service as single mode that can be installed independently by the machine or the user. In other words, install the entire service ignoring the component’s ZCONTEXT.

Attribute	Description
ZBITARCH	Use this attribute to specify the bit size-based architecture to which the service can be deployed. - Set to 32 to deploy to 32-bit architectures. - Set to 64 to deploy to 64-bit architectures. - Leave blank to deploy to all architectures.

Reporting Attributes in ZSERVICE

Some of the attributes in the ZSERVICE Class are calculated. They are updated when the service is installed, verified, updated, repaired, or deleted and reported in the Sentient agent computer's service objects.

Note: These attributes should not be modified using Sentient Administrator CSDB Editor.

Calculated ZSERVICE Attributes – DO NOT MODIFY

Attribute	Description
ZSVCCSTA	This status code for the service is used to determine why files for a service might not be correctly deployed. Values range from 000–999.
SIZE	The uncompressed size of the application that is displayed to the users in the extended information area in Sentient Application Self-service Manager interface. This “calculated” field is the cumulative value of the SIZE that is defined in the PACKAGE class; <i>do not modify it</i> .
COMPSIZE	The compressed size of the application that is displayed to the users in the extended information area in Sentient Application Self-service Manager interface. This “calculated” field is the cumulative value of the COMPSIZE that is defined in the PACKAGE class; <i>do not modify it</i> .
ZAVIS	The Sentient agent manages this attribute to show, in the catalog, the different states of the application. The four states are: Available indicates whether a service is available from the Configuration Server. Verified indicates whether a service has been verified. Installed indicates whether the service has been installed. Synchronized indicates whether the installed service has all of the latest changes from the Configuration Server. The valid values for each state are: Y (Yes), N (No), and X (unknown).
VERDATE	This attribute indicates (in local time, in the format of MMM DD,YYYY HH:MM:SS) when the application was last verified on the Sentient agent computer. This is displayed to the users in the extended information area in Sentient Application Self-service Manager interface. The Sentient agent manages this attribute.

Attribute	Description
UPGDATE	This attribute indicates (in local time, in the format of MMM DD,YYYY HH:MM:SS) when the application was last updated on the Sentient agent computer. This is displayed to the users in the extended information area in Sentient Application Self-service Manager interface. The Sentient agent manages this attribute.
INSTDATE	This attribute indicates (in local time, in the format of MMM DD,YYYY HH:MM:SS) when the application was installed on the Sentient agent computer. This is displayed to the users in the extended information area in Sentient Application Self-service Manager interface. The Sentient agent manages this attribute.
DELDATE	This attribute indicates (in local time, in the format of MMM DD,YYYY HH:MM:SS) when the application was removed on the Sentient agent computer. The Sentient agent manages this attribute.

Catalog Processing on Agent Machine

Local catalog processing reduces the network bandwidth and the number of connects required to the Configuration Server to manage applications. Use a Timer or a Notify command to make the machine connect with `context = m` on the RADSKMAN command line.

Note: The ZSVCMODE attribute of the application must be M or MU to use this feature.

If you have only one user for a computer or multiple users on one computer with the same entitlements, local catalog processing allows you to make one connection to the Configuration Server for the machine and user components of a service. During the machine connect, the required configuration information will be downloaded to the agent computer, machine and user components of the services will be downloaded in a compressed format, and the machine components will be installed. For the user connect, set `cat = m`, `local = y`, and `context = u` on the RADSKMAN command line. When the user connect is made, the user's services will be resolved based on the machine's service list, and the already downloaded resources will be added, modified, or deleted as needed.

If an Sentient agent computer has two or more users with distinct entitlements, you will need to create policies for each machine so that all components for all of the services for all users of that Sentient agent computer will be downloaded. Contact Professional Services for implementation.

Creating Internal Users and Groups in Configuration Server Database

There may be times when you need to create individual users or groups in Sentient. For example, you might want to create a lab environment that is used to test the distribution and management of your data. To create a simple environment, use the Sentient Administrator CSDB Editor to create several users, assign them to groups, and then assign services to the groups. Use the Sentient Administrator password to log on to the Sentient Administrator CSDB Editor. By default, the user name is `ADMIN` and the password is `secret`.

Creating Users and Groups

In this section, you will learn how to create a user in the USER Class of the POLICY Domain. You can follow the same steps to create a new WORKGRP or DEPT instance by substituting the appropriate Class name.

In the following example, you will use the Sentient Administrator CSBD Editor to create a new user (SSampson) in the USER Class.

To create a new user:

1. Click **Start > Programs > Sentient Enterprise Administrator > Sentient Enterprise Administrator CSDB Editor**. The **Sentient Administrator CSDB Editor Security Information** dialog box opens.
2. Type the user ID and password, and click **OK**. The Sentient Administrator CSDB Editor window opens.
3. Double-click **PRIMARY**.
4. Double-click **POLICY**.
5. Right-click **Users (USER)**.
6. Select **New Instance**.
7. In the **Create Instance** dialog box, type a display name (up to 25 characters) and an instance name (up to 25 characters).
8. Click **OK**.

The user instance, SSampson, is created.

Assigning Users to Groups

If you have created several users, you might want to assign them to one or more groups. In the following example, the Sentient Administrator CSDB Editor is used to assign the user SSampson to the Sales department.

To assign a user to a group:

1. Click **Start > Programs > Sentient Enterprise Administrator > Sentient Enterprise Administrator CSDB Editor**. The **Sentient Administrator CSDB Editor Security Information** dialog box opens.
2. If necessary, type a user ID and password, and click **OK**. The RCA Administrator CSDB Editor window opens.
3. Double-click **PRIMARY**.
4. Double-click **POLICY**.
5. Double-click **Users (USER)** to open the list of all user instances.
6. Right-click the USER instance (for example, SSampson) and select **Show Connections** from the shortcut menu. The **POLICY.USER Connections** dialog box opens and displays a list of classes to which you can connect the selected instance.

7. Select **Departments (DEPT)**, and then click **OK**.
The DEPT Class instances appear in the list view. This enables you to easily make a connection between an instance in the DEPT Class and an instance in the USER Class.
8. Select the **Sales** instance from the list view and drag it to the appropriate USER instance (for example, SSampson). When your cursor turns into a paper clip, release the mouse button. The **Select Connection Attribute** dialog box opens.

Note: The Sales instance, shown in the Departments (DEPT) Class, might not appear in your Configuration Server Database. To add this instance (or instances that are appropriate to your organization), follow the procedure ["To create a new user:" on the previous page](#). However, instead of right-clicking USER, right-click the appropriate Class, such as Departments (DEPT).

9. Click **Copy** to create the connection from Users.SSampson to Department.Sales.
10. Click **Yes** to confirm the connection.
11. Click **OK** when you receive the confirmation that "SSampson has been connected to Sales." SALES is now listed under the SSAMPSON user instance, indicating that SSampson is part of the Sales department.

Connecting Services to Groups

Whether you are using an external policy source or managing policy within Sentient, you will need to define the services that your subscribers will receive.

In this section, you will learn how to connect users and groups to the services that Sentient will manage. In the following example, the Sentient Administrator CSDB Editor is used to authorize the WinZip application for all users in the Sales department.

To connect the WinZip application to the Sales department:

1. Click **Start > Programs > Sentient Enterprise Administrator > Sentient Enterprise Administrator CSDB Editor**. The **Sentient Administrator CSDB Editor Security Information** dialog box opens.
2. If necessary, type a user ID and password, and click **OK**. The Sentient Administrator CSDB Editor window opens.
3. Double-click **PRIMARY**.
4. Double-click **POLICY**.
5. Double-click **Departments (DEPT)** to open the Departments Class.
6. Right-click the **Sales** instance (in the tree view) and select **Show Connections**. The **POLICY.DEPT Connections** dialog box opens. This dialog box displays a list of classes to which you can connect the selected instance.
7. From the **Show connectable classes for domain** drop-down list, select **SOFTWARE**.
8. Click **Application (ZSERVICE)**, and then click **OK**. The instances in the ZSERVICE Class appear in the list view.

9. Select the **WinZip** Instance from the list view and drag it to the appropriate Departments instance (in this example, Sales). When your cursor turns into a paper clip, release the mouse button. The **Select Connection Attribute** dialog box opens.
10. Click **Copy** to create the connection from DEPT.SALES to Application.WinZip.
11. Click **Yes** to confirm the connection.
12. Click **OK** when you receive the confirmation that “Sales has been connected to WinZip.” In the Sentient Administrator CSDB Editor, note the following:
 - WinZip is listed under DEPT.SALES, which indicates that the entire Sales department is authorized to receive the WinZip application.
 - SSampson is listed under the USER Class, as is the SALES Instance, indicating that SSampson is part of the Sales department.Therefore, based on these two conditions, Sentient will manage the WinZip application on SSampson’s computer.

Whether you are using an external policy store, or managing policy within Sentient, you can quickly modify the services that individuals are authorized for by manipulating the connections between services and groups, adding users to groups, or removing users from groups.

Creating Timers in Configuration Server Database

To schedule the deployment of a service, configure a timer in the Configuration Server Database. When the Sentient agent computer connects to the Configuration Server, the timer information is transferred to the Sentient agent computer in the ZTIMEQ object.

Creating a Timer

Use Sentient Administrator CSDB Editor to create a Scheduling (TIMER) instance in the SOFTWARE Domain.

To create a TIMER instance in the SOFTWARE Domain:

1. Click **Start > Programs > Sentient Enterprise Administrator > Sentient Enterprise Administrator CSDB Editor**. The **Sentient Administrator CSDB Editor Security Information** dialog box opens.
2. Enter the user ID and password, and click **OK**. The default user ID is `admin`. The default password is `secret`. The Sentient Administrator CSDB Editor window opens.
3. Navigate to **PRIMARY.SOFTWARE.Scheduling (TIMER)**, and right-click TIMER Instance.
4. Select **New Instance**. The **Create Instance** dialog box opens.
5. Type a name, for example, `Mandatory Apps Timer`, and click **OK**.

Scheduling (TIMER) Class Attributes

The TIMER Class attributes are described in the following table. They contain information that is needed to execute the timer on the Sentient agent computer.

Scheduling (TIMER) Class Attributes

Attribute	Usage
<code>_ALWAYS_</code>	Stores connections to other instances.
<code>NAME</code>	The friendly name for this instance.
<code>PINGDLAY</code>	If <code>ZNOPING=N</code> , <code>PINGDLAY</code> specifies the time (in milliseconds) between pings. The default is 2000 .
<code>PINGCNT</code>	If <code>ZNOPING=N</code> , <code>PINGCNT</code> specifies number of ping attempts. The default is 3 .
<code>RETRYINT</code>	Specify the number of minutes to wait between command executions. Note: This is ignored if <code>RETRYFLG=N</code> .
<code>RETRYLMT</code>	Specify the number of times to retry the command. Specify <code>0</code> to retry until the command succeeds. Note: This will be ignored if <code>RETRYFLG=N</code>.
<code>NETAVAIL</code>	Use to check the availability of the network. - Specify <code>Y</code> to check for network availability before executing the <code>TIMER</code> instance. If the network is not available, network availability will be checked every time the timer wakes up until the network is available. - Specify <code>N</code> (the default) and the <code>TIMER</code> instance will be executed without checking for network availability. - Specify <code>W</code> to check for network availability before executing the <code>TIMER</code> instance. If the network is not available and the time window's end limit has been reached, the timer will wait until the next time window before checking for network availability again.
<code>RETRYFLG</code>	Use to specify the retry activity. - Specify <code>Y</code> to retry the command up to <code><RETRYLMT></code>, ignoring the end time for the timer. - Specify <code>W</code> to retry the command up to up to <code><RETRYLMT></code>, but stop retrying after the specified limit time has passed. - Specify <code>N</code> to not retry. Note: A return code other than 200 will indicate success, and stop the retries.
<code>RETRYRC</code>	Specify return codes that qualify for the retry logic. If this variable does not exist or is blank, <code>RETRYRC</code> will default to 200 which means there was a fatal error due to a network connection failure with the Configuration Server. If you populate this attribute and a return code of 200 qualifies for a retry, be sure to specify 200 in the list. Example: <code>RETRY=200, 202, 209</code>

Attribute	Usage
ZNOPING	Use this attribute to control automatic sensing of a network connection between the Sentient agent computer and the Configuration Server. The default is <code>Y</code>. - Specify <code>Y</code> to prevent the Scheduler service from pinging the Configuration Server. This is especially useful for mobile users. - Specify <code>N</code> to have the Scheduler service to ping the Configuration Server. - Specify <code>W</code> if you are specifying an end limit in the ZSCHDEF attribute. The Scheduler will ping the Configuration Server before executing the command. If the Configuration Server is unavailable, the ZPENDING flag will be set to <code>W</code>. If the ZSCHDEF has a limit time, then, when that time passes, the ZPENDING flag will be set to <code>N</code> and the Scheduler will not attempt to execute the command until its next scheduled time. - Specify <code>S</code>, if you want the Scheduler service to ping SAP entries according to their priorities, along with the Configuration Server. - Specify <code>P</code>, if you are specifying an end limit in the ZSCHDEF attribute. The Scheduler pings the Configuration Server and SAP entries as per their priorities before executing the command. If the Configuration Server and all SAP entries are unavailable, the ZPENDING flag is set to <code>W</code>. If the ZSCHDEF has a limit time, then when that time passes, the ZPENDING flag is set to <code>N</code>, and the Scheduler will not attempt to execute the command until its next scheduled time. An expired timer continually evaluates whether communications with the Configuration Server can be established. When communications are established, the command line that is associated with the timer is executed. After executing the command line, the Scheduler service resumes normal evaluation of whether the timer has expired again. If this attribute is not present in the ZTIMEQ object, the Scheduler service will not ping the Configuration Server. If the Configuration Server is successfully pinged, the command in ZRSCCMDL executes and the ZPENDING attribute—in the Sentient agent's ZTIMEQ object—is set to <code>N</code>, to indicate that the Scheduler service does not need to ping the Configuration Server again. If the Configuration Server is not successfully pinged, the timer is not processed any further, and the ZPENDING attribute value remains <code>Y</code>, to indicate that the next time the Scheduler service “pops,” it should ping the Configuration Server again.
ZRSCCMDL	Use this attribute to specify the command line that is executed on the Sentient agent computer when the timer expires. Use RADSKMAN to verify and update Sentient-managed mandatory applications. For a list of the parameters and examples, see appendix C, "RADSKMAN Command Line Parameters" on page 477.
ZSCHDEF	Use this attribute to specify when the timer expires. The syntax for this attribute varies depending on the frequency, which can be DAILY, HOURLY, WEEKLY, MONTHLY, INTERVAL, MONTHDAY, NUMDAY, WEEKDAY, or STARTUP. For instructions on how to set ZSCHDEF, see "Specifying the

Attribute	Usage
	Timer Expiration (ZSCHDEF)" on page 73.
ZSCHFREQ	Use this attribute to specify how often the timer should expire. - Specify <code>ONCE</code> to have the timer to expire one time. - Specify <code>PERIODIC</code> to have the timer to expire repeatedly. - Specify <code>RANDOM</code> to have the timer to expire in random intervals. For more information, see "Deploying Applications over a Period of Time" on page 78.
ZSCHTYPE	This attribute is valid only when <code>ZSCHFREQ=PERIODIC</code>. Valid values are <code>IMMEDIATE</code> and <code>DEFERRED</code>. Specify <code>DEFERRED</code> to indicate that the first time an event is attempted to be launched, it will be deferred until the next scheduled time, regardless of when the timer instance is evaluated. This was designed so that events that are scheduled for off-peak hours will not launch while a user is working. Example 1 Assume a timer with <code>ZSCHDEF=DAILY (&ZSYSDATE, 04:00:00)</code>. - If <code>ZSCHTYPE=IMMEDIATE</code> and it is: - Earlier than 4:00 a.m., the command in the instance will be executed the same day at 4:00 a.m. - Later than 4:00 a.m., the command in the instance will be executed immediately. - If <code>ZSCHTYPE=DEFERRED</code> and it is: - Earlier than 4:00 a.m., the command in the instance will be executed the same day at 4:00 a.m. - Later than 4:00 a.m., the command in the instance will be executed the next day at 4:00 a.m. Example 2 Assume a timer with <code>ZSCHDEF=WEEKDAY (FRIDAY, 04:00:00)</code>. - If <code>ZSCHTYPE=IMMEDIATE</code> and it is: - Either not Friday, or earlier than 4:00 a.m. on Friday, the command in the instance will be executed on Friday at 4:00 a.m. - Later than 4:00 a.m. on Friday, the command in the instance will be executed immediately. - If <code>ZSCHTYPE=DEFERRED</code> and it is: - Not Friday, the command in the instance will be executed on the next occurring Friday, at 4:00 a.m. - Earlier than 4:00 a.m. on Friday, the command in the instance will be deferred one week and executed a week later on the following Friday, at 4:00 a.m. - Later than 4:00 a.m. on Friday, the command in the instance will be executed a week later on Friday at 4:00 a.m.

Attribute	Usage
ZSTOP	Expressions evaluating to “true” in ZSTOP attributes cause resolution of the instance to be skipped. If left blank, the instance is accepted and resolution continues. This is useful if you want to set conditions on which of your subscribers receive the timer.
<i>The values for the following attributes are inherited from the <code>_BASE_INSTANCE_</code> of the <code>TIMER</code> Class; they should not be edited.</i>	
RUNSYNC	Specifies whether synchronous timer execution will take place. The default value is <code>Y</code> .
ZOBJPRI	The deployment priority level (relative to the other elements being deployed during the Sentient agent connect) of the ZTIMEQ object. A value of 90 is inherited from the <code>_BASE_INSTANCE_</code>. Note: Elements with priority levels that are lower than this value will be deployed before this ZTIMEQ object.
ZSCHMODE	This attribute is specific to Sentient Application Self-service Manager and is used when a ZTIMEQ Instance is run. Its value, <code>Default</code> , should not be changed.
ZSVCID	The object ID of the Application instance to which this Scheduling instance is connected.
ZCHNNAME	The Configuration Server Database domain that contains the Application instance to which this Scheduling instance is connected.
ZPRVNAME	The name of the Configuration Server to which the subscriber that is receiving this timer instance is connected. The value is inherited from the <code>_BASE_INSTANCE_</code> .
ZCREATE	The Scheduler “create” method that runs on the Sentient agent computer. The value is inherited from the <code>_BASE_INSTANCE_</code> .
ZVERIFY	The Scheduler “verify” method that runs on the Sentient agent computer. The value is inherited from the <code>_BASE_INSTANCE_</code> .
ZUPDATE	The Scheduler “update” method that runs on the Sentient agent computer. The value is inherited from the <code>_BASE_INSTANCE_</code> .
ZDELETE	The Scheduler “delete” method that runs on the Sentient agent computer. The value is inherited from the <code>_BASE_INSTANCE_</code> .

Configuring the Timer

This section offers a review of the syntax that is used to configure the attributes of the `TIMER` instance. Following that, in the section, ["Deploying Applications over a Period of Time" on page 78](#), is a sample exercise on how to configure a `TIMER` instance to deploy mandatory applications during off-peak hours.

Specifying the Timer Expiration (ZSCHDEF)

Use ZSCHDEF and ZSCHFREQ attributes to specify when and how often a timer will expire.

- ZSCHDEF indicates when the timer will expire;
- ZSCHFREQ indicates how often the timer will expire.

The syntax of the ZSCHFREQ attribute will influence the settings of the ZSCHDEF attribute. Use ["ZSCHFREQ=RANDOM" on the next page](#) to determine the appropriate syntax for the value of ZSCHDEF. Before configuring the ZSCHDEF attribute, review the following syntax-formatting considerations.

- The value of WEEKDAY must be UPPERCASED and will accept only the days of the week: MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY, and SUNDAY.
- In all ZSCHDEF attribute syntax, the time value must be expressed in base-24 time, in the format **HH:MM:SS**. Valid values are **00:00:00–23:59:59**.
- In all ZSCHDEF attribute syntax, the *date* value must be expressed in the format **YYYYMMDD**.

Start Time and End Time Parameters

When `ZSCHFREQ=RANDOM` is specified the time parameter is automatically replaced with two parameters, **start time** and **end time**. The start time and end time parameters are used to specify the time limit between which the command can be chosen to be executed. For example, the command

```
DAILY (20070707,10:00:00,12:00:00[,20:00:00])
```

will choose to execute in between the start time parameter (10 a.m.) and the end time parameter (12 p.m.) on July 7, 2007. However, it will not execute after the limit time parameter (8 p.m.). So, if the target machine is not powered on (or the Scheduler is not running) during this time, the command will not execute. It will be rescheduled for the next occurrence to start in between 10 a.m. and 12 p.m." and end before 8 p.m, in this case. As it is a "daily" command, it will execute on the next day, July 8, 2007.

Limit Time Parameter

The **limit time** parameter is used to specify a time after which the command will not be executed. For example, the command

```
DAILY (20070707,18:00:00[,20:00:00])
```

will execute any time between 6 p.m. and 8 p.m. on July 7, 2007, but it will not execute after 8 p.m. So, if the target machine is not powered on (or the Scheduler is not running) during this time, the command will not execute. It will be rescheduled for the next occurrence of "between 6 p.m. and 8 p.m." which, in this case, because it is a "daily" command, will execute on the next day, July 8, 2007.

ZSCHFREQ=RANDOM

When ZSCHFREQ=RANDOM is specified the **time** parameter is automatically replaced with two parameters, **start time** and **end time**.

Be sure to not use the limit time parameter as an *end time* indicator; their functionalities are different and doing so will result in a malformed ZSCHDEF command.

- If ZSCHFREQ=RANDOM and the limit time parameter is *NOT* specified:
The end time parameter can span midnight (it can be the next day). For example, the commands
`DAILY (20070707, 20:00:00, 06:00:00)` and
`NUMDAYS (20070707, 20:00:00, 06:00:00, , 14)`
will execute at random times between 8 p.m. on July 7, 2007 and 6 a.m. on July 8, 2007. Note that even though the limit time parameter is not specified in either command, NUMDAYS still requires the third comma in order to be considered a valid argument; DAILY does not.
- If ZSCHFREQ=RANDOM and the limit time parameter *IS* specified:
The end time parameter cannot span midnight. If it does, the RADTIMEQ create method will log a warning and set the start time to midnight (00:00:00).

ZSCHDEF Attribute Syntax

Attribute	Description
HOURLY	The timer will expire and run hourly, starting any time after the specified time but not later than the specified limit time, based on the system's date. Syntax: <code>&SYSDATE,time[,limit time]</code> Example: <code>ZSCHDEF=HOURLY (&ZSYSDATE, 04:30:00)</code> Note: If ZSCHFREQ=RANDOM, the start time and end time parameters are activated. Syntax: <code>&SYSDATE,start time,end time[,limit time]</code> Example: <code>ZSCHDEF=HOURLY (&ZSYSDATE, 04:30:00, 09:00:00)</code> For more information, see "Specifying the Timer Expiration (ZSCHDEF)" on the previous page.
DAILY	The timer will expire and run daily at the specified time (but not later than the specified limit time), based on the system's date. Syntax: <code>&SYSDATE,time[,limit time]</code> Example: <code>ZSCHDEF=DAILY (&ZSYSDATE, 12:00:00)</code> Note: If ZSCHFREQ=RANDOM, the start time and end time parameters are activated. Syntax: <code>&SYSDATE,start time,end time[,limit time]</code>

Attribute	Description
	Example: ZSCHDEF=DAILY (&ZSYSDATE,12:00:00,14:00:00,18:00:00) For more information, see "Specifying the Timer Expiration (ZSCHDEF)" on page 73.
WEEKLY	The timer will expire and run at the specified time (but not later than the specified limit time) on every seventh day, based on the system's date. Syntax: &SYSDATE,time[,limit time] Example: ZSCHDEF=WEEKLY (&ZSYSDATE,08:00:00) Note: If ZSCHFREQ=RANDOM, the start time and end time parameters are activated. Syntax: &SYSDATE,start time,end time[,limit time] Example: ZSCHDEF=WEEKLY (&ZSYSDATE,08:00:00,12:00:00,14:00:00) For more information, see "Specifying the Timer Expiration (ZSCHDEF)" on page 73.
INTERVAL	The timer will expire and run every <i>n</i> minutes starting at the specified time (but not later than the specified limit time), based on the system's date. Once the limit time has been reached, the command will stop executing. The same command will execute the next day, again, within the same time limits. Syntax: &SYSDATE,time,[limit time],interval Example: ZSCHDEF=INTERVAL (&ZSYSDATE,04:00:00,06:00:00,30) Note: INTERVAL must be specified in minutes. The third comma is required regardless of whether a third argument is specified. If ZSCHFREQ=RANDOM, the time parameter is automatically replaced with two parameters, start time and end time. Syntax: &SYSDATE,start time,end time,[limit time],interval Example: ZSCHDEF=INTERVAL (&ZSYSDATE,04:00:00,23:00:00,06:00:00,30) For more information, see "Specifying the Timer Expiration (ZSCHDEF)" on page 73.
WEEKDAY	The timer will expire and run at the specified time (but not later than the specified limit time) on the specified weekday, every week. Syntax: WEEKDAY,time[,limit time] Example: ZSCHDEF=WEEKDAY (TUESDAY,01:00:00)

Attribute	Description
	Note: If ZSCHFREQ=RANDOM, the time parameter is automatically replaced with two parameters, start time and end time. Syntax: WEEKDAY, start time, end time[, limit time] Example: ZSCHDEF=INTERVAL (TUESDAY, 04:00:00, 06:00:00, 10:00:00) For more information, see "Specifying the Timer Expiration (ZSCHDEF)" on page 73.
MONTHDAY	The timer will expire and run at the specified time (but not later than the specified limit time) on the specified weekday in the week of the month that is indicated by the 4th parameter. Syntax: WEEKDAY, time, [limit time], week of the month Example: ZSCHDEF=MONTHDAY (TUESDAY, 01:00:00, , 2) Note: The valid values for the 4th parameter are 1–5. If this argument is not specified, the timer will expire during the first week of the month. The third comma is required regardless of whether a third argument is specified. If ZSCHFREQ=RANDOM, the time parameter is automatically replaced with two parameters, start time and end time. Syntax: WEEKDAY, start time, end time, [limit time], week of the month Example: ZSCHDEF=MONTHDAY (TUESDAY, 01:00:00, 04:00:00, , 2) Important Note: Consider the consequences of specifying an end time that spans midnight (occurs on the following day). For more information, see "Specifying the Timer Expiration (ZSCHDEF)" on page 73.
MONTHLY	The timer will expire and run at the specified time (but not later than the specified limit time) on the nth of every month, starting in the specified month and year. Syntax: date, time[, limit time] Example: ZSCHDEF=MONTHLY (20040215, 01:00:00, 05:30:00) Note: If ZSCHFREQ=RANDOM, the time parameter is automatically replaced with two parameters, start time and end time. For more information, see "Specifying the Timer Expiration (ZSCHDEF)" on page 73. Syntax: date, start time, end time[, limit time]

Attribute	Description
	Example: ZSCHDEF=MONTHLY (20040215,01:00:00,05:30:00,07:00:00) Note: This attribute reschedules differently than other ZSCHDEF attributes; it will reschedule by adjusting the month (but retaining the date) for which it was originally scheduled, rather than adjusting the date based on when it eventually ran. Example: Assume ZSCHDEF=MONTHLY (20040116,05:30:00) and that the Sentient agent device was powered off on January 16th and that the timer didn't execute until January 18th. The new schedule would automatically revise to MONTHLY (20040216,05:30:00) rather than MONTHLY (20040218,05:30:00).
NUMDAYS	The timer will expire and run at the specified time (but not later than the specified limit time) on the specified date, then again on every nth day (as specified by the 4th parameter). Syntax: date,time,[limit time],number of days Example: ZSCHDEF=NUMDAYS (20040803,18:00:00,21:30:00,14) Note: The third comma is required regardless of whether a third argument is specified. If ZSCHFREQ=RANDOM, the time parameter is automatically replaced with two parameters, start time and end time. Syntax: date,start time,end time,[limit time],number of days Example: ZSCHDEF=NUMDAYS (20040803,18:00:00,21:30:00,22:00:00,14) For more information, see "Specifying the Timer Expiration (ZSCHDEF)" on page 73.
STARTUP	When the Scheduler starts on the Sentient agent device, it will immediately execute all Timer instances that have ZSCHDEF=STARTUP specified. It will check for special conditions such as NETAVAIL, ZNOPING, and RETRYFLG. After executing all the STARTUP instances, RADSCHEM will return to its regular timer loop. It will execute STARTUP instances in the regular timer loop only if the ZPENDING flag on that instance was set (because NETAVAIL or ZNOPING could not get through or RETRYFLG is on and the return code was 200 during startup run).

Deploying Applications over a Period of Time

Applications can be deployed over a period of time in order to balance the workload on the Configuration Server and alleviate network congestion.

To do this, configure the timer for “random” expiration and use ZSCHDEF to specify the period of time during which the applications should be deployed. The time-period options are detailed in ["ZSCHFREQ=RANDOM" on page 74](#).

In the following example, a timer will be configured to deploy mandatory applications on a weekly basis. The deployments will be scheduled to run between 5:00 p.m. and 7:00 p.m. in order to alleviate network congestion.

To specify when the timer expires:

1. In Sentient Administrator CSDB Editor, navigate to the timer instance and double-click **ZSCHFREQ**. The **Editing Instance** dialog box opens.
2. In the **Frequency** drop-down list, select **RANDOM**.
3. Click **ZSCHDEF**.
4. In the **Timer Parameter** text box, type `WEEKLY (&ZSYSDATE, 17:00:00, 19:00:00)`.
5. Click **ZSCHTYPE**.
6. In the **Type (Immediate/Deferred)** drop-down list, select **IMMEDIATE**.
7. If you are done editing the attributes for the timer instance, click **OK** and then click **Yes** when prompted to confirm your changes.
or
Select another attribute to edit.

Specifying the Command Line (ZRSCCMDL)

When the timer expires, it executes on the Sentient agent computer any command line that you've specified.

Note: To see how timers work, create a timer that runs a command line such as `SystemDrive:\Notepad.exe`. Remember to configure the timer to immediately expire, and attach it to a service. Then, deploy the service. When the timer expires on the Sentient agent computer, the Notepad application opens.

To specify a command line:

1. In Sentient Administrator CSDB Editor, navigate to the timer instance and double-click **ZRSCCMDL**. The **Editing Instance** dialog box opens.
2. In the **Command line to execute** text box, type the appropriate command line to execute the program.
3. Click **OK**, and then click **Yes** when prompted to confirm your changes.

At the beginning of this section, it is indicated that new mandatory applications would be deployed on a weekly basis. The following procedure will demonstrate how to specify a command line that will update all mandatory services and perform self-maintenance.

Connecting the Timer to a Service

Once you have created the timer, you must connect it to a service. Each subscriber that receives the service with which the timer is associated will receive the timer information in the ZTIMEQ object the next time his/her Sentient agent connects to the Configuration Server.

In the example in this section, a timer is created and intended to deliver mandatory applications. The timer is then connected to a service and assumed that all subscribers are receiving it.

Note: For the following exercise, assume a service named ProDraw. The steps are identical for the services in your database.

To connect the timer to a service:

1. In Sentient Administrator CSDB Editor, navigate to **PRIMARY.SOFTWARE.Application (ZSERVICE)**, and right-click **ProDraw**.
2. In the menu that opens, click **Show Connections**. The **SOFTWARE.ZSERVICE Connections** dialog box opens.
3. Select **Scheduling (TIMER)** and click **OK**. The TIMER Class instances appear in the list view.
4. In the list view, click **Mandatory Apps Timer** and drag it to ProDraw. When the cursor changes to a paper clip icon, release the mouse button. The **Select Connection Attribute** dialog box opens.
5. Click **Copy**.
6. Click **Yes** to confirm that you want to connect the ProDraw service to the Mandatory Apps Timer.
7. Click **OK** to close the confirmation message.

Testing the Timer Deployment

The first time that an Sentient agent computer connects to the Configuration Server after the timer has been created, the timer information is transferred to the Sentient agent computer in the ZTIMEQ object, ZTIMEQ.EDM.

In the exercise that follows, you will force the Sentient agent computer to connect to the Configuration Server so that you can view the ZTIMEQ object.

To connect to the Sentient Configuration Server:

1. On the Sentient agent computer, go to a command prompt and change the directory to the location of RADSKMAN. The default location is *<InstallDir>\Agent*.
2. Type `radskman ip=manager_ip,port=mgr_port`. Be sure to specify a valid IP address and port for the Configuration Server.

Note: For information about RADSKMAN and the above parameters, see appendix C, "RADSKMAN Command Line Parameters" on page 477.

3. Press **Enter**.

After the Sentient agent connect completes, you can view the ZTIMEQ object on the Sentient agent computer, as described in the next section.

Note: If you plan to do additional testing, consider creating a batch file that contains the command line. Save the file in `IDMSYS` (by default `<InstallDir>\Agent`) on the Sentient agent computer. Then, create a shortcut on the desktop of the Sentient agent computer.

Removing the Timer Object

After the timer expires, the ZTIMEQ object will be removed from the Sentient agent computer during its next connect to the Configuration Server. This is dependent on the expiration settings in the `TIMER.ZSCHFREQ` attribute.

- If the timer is configured to expire once, the ZTIMEQ object will be removed immediately after the timer expires, during the next Sentient agent connect to the Configuration Server.
- If the timer is configured to expire more than once, the ZTIMEQ object will be removed after the timer expires for the last time, during the next Sentient agent connect to the Configuration Server.

Viewing the Timer Object

After having forced the Sentient agent to connect to the Configuration Server and retrieve the ZTIMEQ object, it can be viewed (and modified) using Sentient Administrator Agent Explorer, which was installed as part of the Sentient Administrator.

The ZTIMEQ object contains one instance for each Scheduling (TIMER) instance in the Configuration Server Database. Therefore, if two services have associated timer instances there will be two instances in the ZTIMEQ object.

To view the ZTIMEQ object on the Sentient agent computer:

1. Click **Start > Programs > Sentient Enterprise Administrator > Sentient Enterprise Administrator Agent Explorer**.
2. Double-click the **ZTIMEQ** object. The ZTIMEQ object opens.

Experimenting with Timers

If you want to experiment with timers, you can modify the `ZRSCCMDL`, `ZSCHDEF`, `ZSCHFREQ`, and `ZSCHTYPE` variables in the ZTIMEQ object on the Sentient agent to see what happens in various situations.

To edit a variable in ZTIMEQ:

1. Click **Start > Programs > Sentient Enterprise Administrator > Sentient Enterprise Administrator Agent Explorer**.
2. Double-click the **ZTIMEQ** object. The ZTIMEQ object opens.
3. Double-click the variable that you want to edit. The **Change Variable** dialog box opens.
4. Type the new value.
5. Click **Save/Exit**.

To quickly determine whether the timer expires, change ZRSCCMDL to run any executable, such as Notepad. When the timer expires, Notepad should open, confirming that the timer expired.

Timer Logs

Timer events are tracked in three logs—RADSCHED.LOG, RADSHIST.LOG, and RADTIMEQ.LOG—that are stored in the IDMLLOG directory by default *<InstallDir>\Agent\Log*).

Timer Logs

Log File	Description
RADSCHED.LOG	Lists the results of the most recent Scheduler expiration. The Scheduler, RADSCHED, runs in the background. It wakes up once per minute and examines the ZTIMEQ object to determine whether a timer has expired. This log contains information from only the most recent expiration.
RADSHIST.LOG	Lists all of the programs that were dispatched because a timer instance expired. It reflects all activity that has taken place since RADSCHED was last started.
RADTIMEQ.LOG	Lists the events that occurred during the last execution of the RADTIMEQ method. RADTIMEQ executes when the application to which the timer is attached is created, updated, verified, or deleted. Only the last execution's events appear in the log, with an indication of what activity took place regarding the application.

Sentient Administrator Tools Access for External Users

This section explains how to enable external users to access Sentient Administrator tools.

You can configure external user accounts specified in Directory Services to log on to the Sentient Administrator tools. Configure the Directory Services using the Sentient Core Console and then enable external users to access Sentient Administrator tools. For more information on configuring Directory Services, see the Directory Services explained in the section Infrastructure Management in the *Sentient Enterprise Console Online Help*.

If a connection to a particular Directory Service is stopped using the Sentient Core Console, the external users specified in that Directory Service cannot access the Sentient Administrator tools.

To enable external users to access Sentient Administrator tools:

1. Click **Start > Programs > Sentient Enterprise Administrator > Sentient Enterprise Administrator CSDB Editor**. The logon dialog box opens.
2. Type your User ID and Password. By default, the user name is ADMIN and the password is `secret`.
3. Click **OK**. The Sentient CSDB Editor window opens.
4. Navigate to PRIMARY.ADMIN.ADMINID._NULL_INSTANCE_.
5. Set the ALWAYS attribute value to ZADMIN.MASTER_ADMIN(&ZADMFUNC). This provides all external users administrative rights to access Sentient Administrator tools.

To enable specific external users to access Sentient Administrator tools:

1. Click **Start > Programs > Sentient Enterprise Administrator > Sentient Enterprise Administrator CSDB Editor**. The logon dialog box opens.
2. Type your User ID and Password. By default, the user name is ADMIN and the password is `secret`.
3. Click **OK**. The Sentient CSDB Editor window opens.
4. Navigate to PRIMARY.ADMIN.ADMINID.
5. Add a new instance to the ADMINID class for the external user you want to provide access from the Directory Service. For example, `<username>@domain name`, `username` is the name of the external user. For more information on how to add an instance to a class, see *Adding an Instance* section in the *Sentient Enterprise Console Online Help*.
6. Set the `_ALWAYS_` attribute to ZADMIN.MASTER_ADMIN(&ZADMFUNC). This will provide the external user administrative rights to access Sentient Administrator tools.

Note: To access Sentient Administrator tools with the external user account, the username is `<username>@<ds common name>`, where `<username>` is the name of the external user and `<ds common name>` is the Directory Service common name.

Naming Conventions to Organize Content in the Configuration Server Database

This section discusses the use of naming conventions to help you organize the software stored in the Configuration Server Database (CSDB).

When publishing applications, subscribers may have different requirements such as:

- Different operating systems.
- Different amounts of free space on their hard drives.
- Different processors, memory, and so on.
- Different data or applications, depending upon their job function or other factors.

You might need to create several packages for a single application because of these varying requirements. To keep your data organized in the CSDB, it is recommended that you create a naming convention to be used within your organization.

This section provides some recommendations that you can use as a starting point to create your own standards.

Categorizing Information

In general, consider using unique high-level identifiers with an underscore (_) to categorize information in the CSDB. Sentient Administrator CSDB Editor groups instances based on the identifier that precedes the underscore.

Note: If you decide to use a high-level identifier *without* an underscore (_), you can use the Sentient Administrator CSDB Editor's filtering capabilities to display only the instances with that identifier.

For example, if you had a Windows XP version and a Windows Server 2003 version of an application to calculate loan amortizations, you might name the packages AMORTIZE_XP and AMORTIZE_2003.

Naming Conventions for the POLICY Domain

It is recommended that you use a variation of the following standards. When naming instances in a workgroup, use information that groups your subscribers appropriately. For example, if your company is organized by division and location, you might use conventions such as the following:

Naming conventions for the USER Class

Format	Description	Example
USERID	Identifies the subscriber.	SJones
DIV_LOC_DESC	Defines ownership or assignment.	CTS_CLE_EVERYONE
DIV	Identifies the division.	CTS (Corporate Technology Services)
LOC	Identifies the location.	CLE (Cleveland)
DESC	Provides additional description of the group.	EVERYONE (all users)

Naming Conventions for the SOFTWARE Domain

In a company organized by division and location, you might organize your data using the following standards.

Naming Conventions for the PACKAGE Class

Format	Description	Example
DIV_LOC_APPNAME_VER_OS	Defines the application.	CTS_CLE_WINZIP_80_XP
DIV	Identifies the division.	CTS (Corporate

Format	Description	Example
		Technology Services)
LOC	Identifies the location.	CLE (Cleveland)
APPNAME	Identifies the application.	WINZIP
VER	Identifies the version of the application.	80
OS	Identifies the operating system that the application runs on.	XP

Naming Conventions for Delivery and Auditing Classes

Format	Description	Example
REG_DIV_LOC_APPNAME_ VER_OS	Defines the application.	NAM_CTS_CLE_WINZIP
REG	Identifies the region.	NAM (North America)
DIV	Identifies the division.	CTS (Corporate Technology Services)
LOC	Identifies the location.	CLE (Cleveland)
APPNAME	Identifies the application.	WINZIP

Determining the conventions that make sense for your organization may take some time. However, creating a convention up front and communicating it to all of your Administrators will keep you organized in the future.

Modifying Component Instances (Non-Production only)

Note: This feature is not supported in a production environment where packages are actively deployed to agents. Using this feature in such an environment might result in massive distributions of unnecessary package updates, since Sentient would detect a package change and automatically distribute the change to all authorized agents.

This section explains how to dynamically edit the distributed contents of a file from within Sentient Administrator CSDB Editor.

The following table describes the advanced edit component commands:

Advanced Edit Component Commands

Command	Typical use
Edit this	Change a serial number deployed with a file. Edit the contents of a readme

Command	Typical use
Component	file, document, spread sheet, logo image, or PDF.
Replace Component Data	Replace the entire contents of an existing file in a package with contents of another file outside of CSDB.
Add a Component	Add a new file instance to a package, using another packaged file as a template.

Accessing the Add, Edit, and Replace Component Commands

For non-production environments only, select **Show Add Edit and Replace options for File type components** to enable the Add, Edit, and Replace commands on the shortcut menu for applicable instances. The setting **Show Add Edit and Replace Options for File type components** is available on the **Advanced** tab of the **Options** dialog box.

When you select this option the **Instance** shortcut menu opens.

Editing a Component

The **Edit this Component** option permits you to select an editor of your choice with which to dynamically modify the distributed contents of a file instance. After you save the changes to the file, the modified file is sent (re-promoted) to the Configuration Server and the instance in the CSDB is updated.

For example, you can edit the distributable contents of any component instance that has been packaged and is represented in the Sentient Administrator CSDB Editor. You must save the edited file with the same name as the original file. The modified file immediately replaces the existing file in the package. The Sentient Administrator CSDB Editor takes care of the publishing for you, saving you from having to republish the package.

The original component's contents are not retained in the Sentient Administrator CSDB Editor.

For information on creating and promoting packages to the CSDB, see "Sentient Administrator Publisher" in the *Sentient Enterprise Console Online Help*.

To edit a component:

Note: The component editor does not know a component's target operating system. Before you edit an instance, it is recommended that you note or export the values of the following attributes, as well as any other attributes, whose values should not change due to your edits:

ZRSCMEMM (PDS Member Name)

ZRSCCRC (Resource CRC)

ZRSCPADM (PDS AdminID)

ZPUBVER (Publisher Version)

If the values of these attributes change as a result of the editing process, it is recommended that you restore the original values.

For example, if a component is targeted for a Windows operating system, the ZRSCMEMM,

ZRSCCRC, and ZRSCPADM attributes do not apply and should be null. After editing a component, if these attributes show values other than null, you should clear the inserted values. Likewise, the values for other attributes, such as ZPUBVER, should remain unchanged due to an edit. If changed, restore ZPUBVER to its original value.

1. In the Sentient Administrator CSDB Editor, go to **View > Options > Advanced** tab, and select **Show Add Edit and Replace options for File type components**.
2. Locate the file instance that belongs to the software package whose contents you want to change in the CSDB.
3. Right-click the instance in the tree view. A shortcut menu opens.
4. Select **Edit this Component** from the shortcut menu. The **Editor Selector** for **AMORTIZE.TXT** dialog box opens.
Use this dialog box to select the resident editor you want to use to make the changes to this file. For example, double-click **Notepad** to open the **AMORTIZE.TXT** file with **Notepad.exe**. The **Editor Selector** dialog box opens.
For details on using this dialog box, see ["Using the Editor Selector Dialog Box" below](#).
5. Double-click an editor from the **Editors** list box. The selected editor is launched and opens the previously selected file component instance.
6. Make the required changes to the file's contents and save it with the *same name*.

Note: If you save the file with a different name, the Sentient component editor cannot make the changes to the file you selected from the Sentient Administrator CSDB Editor.

7. Close the editor.
8. If necessary, click **OK** to close the **Editor Selector** dialog box and save the changes to the instance.
9. The **File Re-Promote** dialog box opens, and requests a confirmation that you want to commit the changes to the selected instance in the database.
 - Click **Yes** to save the changes, perform a mini-publish and re-promote the changed file to the database.
 - Click **No** to cancel the edits and the edit session. No changes are made to the database instance.

After you commit the changes, the instance attributes for ZRSCDATE ZRSCTIME, and ZRSCSIZE reflect the edited file size and date.

Note: It is recommended that you compare all attributes' values against their original ones, and restore any values that changed unnecessarily. For example, you might need to clear values from OS-related attributes whose values were originally null. For details, see the note text at the beginning of this procedure.

Using the Editor Selector Dialog Box

Use the **Editor Selector** dialog box to select a resident editor with which to edit the named file.

To select an editor, choose an entry from the **Editors** list, or click **Association** to use the Windows default editor associated with file's extension.

To add, delete, or rename the editors displayed in the **Editors** list, use the top three buttons on the right.

The following table summarizes the **Editor Selector** dialog box button actions.

Editor Selector Dialog Box Buttons

Button	Action
Browse/Add	Opens a File Browse dialog box that you can use to select a program to add to the list of editors. Adds an entry from the Browse dialog box to the Editors list below the highlighted entry.
Delete	Deletes a selected (highlighted) program from the Editors list.
Info/Rename	For the selected editor program, opens a dialog box that displays the editor's program name and path, and allows you to assign a friendly name to display on the Editors list.
OK	Opens the selected file, named in the title bar, with the highlighted editor. Alternatively, you can double-click on the Editors list-box entry.
Association	Opens the selected file, named in the title bar, with the default editor associated with that file type through Windows. If no association is available, you will be returned to the Editor Selector dialog box where you can specify an editor or cancel.
Cancel	Cancels the Edit a Component command without making any changes to the selected component or package.

Replacing Component Data

Use the **Replace Component Data** command to replace the entire contents of a packaged file with the contents of a file that exists outside the database. For example, if there is a new **Readme.TXT**, and you want to completely replace its contents, prepare a revised **Readme.TXT** file outside of the Sentient Administrator CSDB Editor. Then, use the **Replace Component Data** command from within the Sentient Administrator CSDB Editor to replace the contents of the existing **Readme.TXT** with the new one. When you use the **Replace Component Data** command, none of the properties of the packaged file changes, just its distributed contents.

Note: It does not support this feature in production environments, and assumes no responsibility if this feature is used where packages are actively being deployed to agents in a production environment. Using this feature in such an environment would likely result in massive distributions of unnecessary package updates, since Sentient would detect a package change and automatically distribute the change to all authorized agents.

To replace [all] component data:

1. Prepare a file outside of Sentient Administrator CSDB Editor whose contents represent the replacement data for a packaged component file in the CSDB. The prepared file can have

any name.

Note: The entire contents of the prepared file will replace the entire data contents of the file selected from the Sentient Administrator CSDB Editor using the **Replace Component Data** command. Verify the contents of the files prior to continuing with this procedure.

2. In the Sentient Administrator CSDB Editor, navigate to **View > Options > Advanced** tab, and select **Show Add Edit and Replace options for File type components**.
3. Locate the file instance in the software package whose data you want to replace in the CSDB.
4. Right-click the instance in the tree view. A shortcut menu opens.
5. Select **Replace Component Data** from the shortcut menu. A standard **Windows Open** dialog box opens.
6. Navigate to the prepared file whose contents will replace the data for the file you selected from the Sentient Administrator CSDB Editor.
7. Click **Open** to select the file with the replacement data. A confirmation message opens.
8. Click **Yes** to replace the data; click **No** to abort the data replacement.
If you click **Yes**, the Sentient Administrator CSDB Editor substitutes the entire contents of the data to be distributed with the selected file's data, and republishes the package.
The instance attributes for ZRSCDATE, ZRSC TIME, and ZRSC SIZE reflect the new file size and date, as previously illustrated.
If you click **No**, the replacement command is canceled.

Adding a Component

The **Add a Component** command is available for file and behavior components only. Files always belong to an owning package, so the most likely place to perform an Add is from the connection under a package. New files can be added by copying the properties of an existing published file; files cannot be created from scratch, that is, from files that have not been promoted from the Sentient Administrator Publisher.

New files are added by selecting an instance to use as a template. The Sentient Administrator CSDB Editor will derive the path for the new instance from the template's LOCATION variable. After a template is selected, right-click it to display the shortcut menu and select **Add Component**. An **Open File** dialog box opens where you can select the file you want to add to your package. A final confirmation box will enable you to stop the process if you need to. Click **Yes** in the box to promote the file that was selected in the **Open File** dialog box. The package now contains the new file.

To add a component to a package:

1. In the Sentient Administrator CSDB Editor, navigate to **View > Options > Advanced** tab, and select **Show Add Edit and Replace options for File type components**.
2. Locate a file instance in the software package whose LOCATION can serve as a template for the component to be added.
3. Right-click the template file's instance in the tree view. A shortcut menu opens.

4. Select **Add a Component** from the shortcut menu. A standard **Windows Open** dialog box opens.
5. Navigate to the prepared file you want to add to the package.
6. Click **Open** to select the file to be added with the same LOCATION as the template file. A confirmation message opens.
7. Click **OK** or **Cancel**.
If you click **OK**, the CSDB Editor promotes the added file and republishes the package. The instance attributes for the added file's LOCATION reflect the template file's location.
If you click **No**, the **Add a Component** command is canceled.

Publishing using Native Packaging

Sentient Batch Publisher enables you to use Sentient Native Packaging to publish applications in Linux and Macintosh environments.

Supported Native Package Types

The following table lists the native package type supported by the Sentient Native Packaging and its expected format.

Native Package and Supported

Native Package	Supported Format
RedHat Linux RPM Package	* .rpm file
Debian/Ubuntu Linux DEB Package	* .deb file

Sentient Native Packaging Command-Line Interface

Sentient Native Packaging is run from the command line. The base input parameter for Sentient Native Packaging is the source depot containing the Linux and Macintosh software. The native packages must be in a disk depot format (the native software packages are resident on disk in a format that can be utilized immediately by the native operating system's software management tools). Sentient Native Packaging is capable of publishing one or more packages in a single publishing session.

In addition, you can specify the selection of the software you want to publish, and in the event Sentient-CSDB user verification is enabled, an optional user ID and password can be designated. Here is an example of command-line usage for Sentient Native Packaging:

```
20070201 18:18:40 Info: Message catalog for en_us loaded.
Usage: rnp -d depot_path -m manager_ip:manager_port
      [-v] [-debug type] [-tmp directory]
      [-user user_id] [-pass password] [-admin]
      [-depth level] [-dist dist_depot_path]
      [-domain domain] [-l logfile] [-help]
      [-i] [-coreq] [-I] [-M] [-S]
      [-a | -A type | -p package[,r=revision][,a=architecture]
      [,v=vendor]
```

```

        -p package2[,r=revision][,a=architecture]
[,v=vendor]...]
    [-P] [-r] [-f prefix]
    [-s] [-t service_type] [-c flag] [-relyondb]

```

The following table contains the description of the command-line arguments for Sentient Native Packaging.

Command-line parameters

Parameter	Description
<code>-a</code>	Specifies to publish all native software available in the depot. This parameter is optional. You cannot use this parameter with <code>-p</code> .
<code>-A type/</code>	Select and publish all packages of specific <i>type</i>. <i>type</i> can also be one of the following: <code>help</code> for a list of valid types for the running platform. <code>all</code> to select all package types. This option would then behave like the <code>-a</code> option. <code>none</code> to select none of the package types. This would then behave like having neither the <code>-a</code> or <code>-A</code> options specified. Multiple package types can be specified and separated by commas. This parameter is optional.
<code>-c flag</code>	This option enables or disables compression on all packages to be published. <i>flag</i> can be one of the following: <code>yes</code> Enable compression for all packages. <code>no</code> Disable compression for all packages. Default behavior is dependent on each package type being published. This parameter is optional.
<code>-d depot path</code>	Specifies the path to the depot directory containing native software packages. Software contained in this depot will serve as an input to Native Packaging. This parameter is required.
<code>-debug type</code>	Specify the level of debugging desired. <i>type</i> can be one of the following: <code>init</code> for initialization data

Parameter	Description
	<code>func</code> for detailed function debugging <code>trace</code> for function tracing <code>cmd</code> for native command executions
	<code>pub</code> for publishing information <code>rapi</code> for Sentient Batch Publisher details <code>all</code> for all the above <code>none</code> to disable debugging Multiple types can be specified and separated by commas. The default behavior is that debugging is disabled. This parameter is optional.
<code>-depth</code>	For Linux RPM and DEB packages. Determines the level of dependency processing for the target package. 0 – Process all dependencies. 1 – (Default) process 1 level of dependency. You can specify any number of level dependencies you require. If no level is defined (<code>-depth</code> is not part of the <code>rnv</code> command), then one level of dependencies is processed (assuming <code>-i</code> option used). This option allows for Sentient Batch Publisher backwards compatibility.
<code>-dist distribution_depot</code>	For Linux RPM and DEB packages. Specifies the path to a distribution depot directory. Published packages will contain <code>DIST=distribution_depot</code> in the <code>CONTENTS</code> field of their package class instance.
<code>-domain domain</code>	Specify which Sentient Configuration Server domain the packages are to be published to. The default domain used is <code>PRIMARY.SOFTWARE</code>. This parameter is optional.
<code>-f prefix</code>	Instructs Sentient Native Packaging to prefix the <code>PACKAGE</code> Class and service class instance names used for the new published package with this prefix. This parameter is optional.
<code>-help</code>	Display help on the command-line usage and the <code>rnv.cfg</code> configuration file format.
<code>-i</code>	Instructs Sentient Native Packaging to include prerequisite software package (supported for RPM packages) with the package you have selected if prerequisite software is present in the source depot.

Parameter	Description
	Dependency information is published regardless of this parameter. This parameter is optional.
<code>-I</code>	Interactive mode. Allows user to select more required packages (dependency). Ignored if neither <code>-i</code> nor <code>-coreq</code> are present or no additional package is required. Note: Available for RPM packages.
<code>-l logfile</code>	Instructs Sentient Native Packaging to store the log details in the <i>logfile</i> specified. If this option is omitted, the default log file created is <code>publish.log</code> . This parameter is optional.
<code>-m ip:port</code>	Specifies the host name or IP address and port of the Sentient Configuration Server to which you intend to publish software. This parameter is required.
<code>-M</code>	Multiple. If <code>-i</code> or <code>-coreq</code> is present (so additional packages are required), and there are several versions of an additional package, then all of them will be displayed in the additional packages menu. Otherwise, only one version of each additional package will be displayed (default). It is ignored if <code>-I</code> is not present.
<code>-p package</code> <code>[,r=revision]</code> <code>[,a=arch]</code> <code>[,v=vendor]</code>	Specifies a software package to publish to the Sentient Configuration Server. Specify the following: an RPM or DEB package on Linux (software selection with optional revision, architecture and vendor. Specifying the software selection alone will work, but if there are multiple products with the same identifier, they will all be published). This parameter is optional. You can specify multiple <code>-p package</code> parameters for multiple package selections. Note: If a package is not specified on the command line, you will be presented with a list of all available packages within the specified depot.
<code>-pass password</code>	Sentient administrator password. This parameter is optional.
<code>-relyondb</code>	For Linux RPM and DEB packages. Use this option to rely on package database information instead of a native command return code when installing packages. A native database query command is used to verify the package was installed (for RPM, <code>rpm -q</code>) Promoting a package with this option sets the Sentient Configuration Server Database attribute RELYONDB to Y. Default is blank (N).

Parameter	Description
<code>-s</code>	Instructs Sentient Native Packaging to skip the creation of services for the packages to be published.
<code>-S</code>	Strict mode. If any requirements for a package are not met (for example, if <code>-i</code> or <code>-coreq</code> option are present and not all additionally required packages are in the depot), the package will not be promoted. It is ignored if <code>-I</code> option is present.
<code>-t svc_type</code>	Use this option to specify the type of service to create. Available values: M for Mandatory O for Optional Default Service type created is M. This parameter is ignored when the <code>-s</code> option is specified.
<code>-tmp dir</code>	Specify an alternative temporary directory to use when creating packages. The default value is <code>/tmp</code> . This parameter is useful when <code>/tmp</code> on the machine where publishing is performed has limited available disk space. This parameter is optional. ADMIN
<code>-user user ID</code>	Sentient administrator user ID. The default is <code>.</code> . This parameter is optional.
<code>-v</code>	Displays the version and build number of the Sentient Native Packager <code>rnp</code> command. This parameter is optional.

Note: When no packages are specified with the `-p` option or by selecting all packages with the `-a` or `-A` options, the Sentient Native Packaging command will present a text based menu of native packages found in the depot directory specified. You can then select individual or all packages from the menu to be published.

Sentient Native Packaging Options File

If you usually use the same source depot, or publish to the same Sentient Configuration Server, you can create a file, `rnp.cfg`, in the same directory where you have the Sentient Native Packaging components installed. Use of this configuration file allows you to preset default option values in the following format:

parameter=value

Example

```
depot=<depot path>
manager_ip=<Sentient configuration server IP or hostname>
manager_port=<port number that the Sentient configuration server uses>
```

Note: By default, `rnp.cfg` is not supplied.

Supported rnp.cfg settings and default values

Setting	Expected Values	Default Value
depot	Fully qualified path to the depot directory	None
manager_ip	IP address or hostname of the Sentient Configuration Server	None
manager_port	Port number of the Sentient Configuration Server	manager_port=3464
user	user=userid Administrator ID used for authentication with the Sentient Configuration Server.	User=ADMIN
create_service	create_service=[yes/no] A value of <code>yes</code> will create a ZSERVICE instance for each of the promoted packages. A value of <code>no</code> will not automatically create a ZSERVICE instance for each of the promoted packages	create_service=yes
service_type	service_type=[M/O] A value of <code>M</code> will cause the promoted ZSERVICE instance to be set as a mandatory service. A value of <code>O</code> will cause the promoted ZSERVICE instance to be set as an optional service.	service_type=M
include_responses	include_responses=[yes/no] A setting of <code>yes</code> will include SVR4 response files when they are found in the Solaris depot. Value of <code>no</code> will not include response files for Solaris SVR4 packages.	include_responses=no
include_dependencies	include_dependencies=[yes/no] A value of <code>yes</code> will attempt to publish RPM dependent packages if they are in the specified depot. A value of <code>no</code> will not attempt to publish RPM dependent packages.	include_dependencies=no

Setting	Expected Values	Default Value
include_adminfile	include_adminfile=[yes/no]	include_adminfile=no
select_types	select_types=[type1,type2,...] Publish all packages of specific types found in the depot directory. Run rnp with the -A help option to get a complete list of supported types.	select_types=none
debug	debug=[type1,type2,...] List specific types of debugging to enable. valid types are: init, func, trace, cmd, pub, rapi, all or none.	debug=none
temp_dir	temp_dir=[dir] Specify an alternate temporary directory to use for creating the packages to publish.	temp_dir=/tmp
domain	domain=FILE.DOMAIN Specify the target FILE.DOMAIN in the Sentient-CSDB where to publish the packages.	domain=PRIMARY.SOFTWARE
compress	compress=[yes/no] Enable or disable compression for all packages to be published. The default behavior is that compression depends on the package type being published.	Package Dependent
password	password=pass Administrator password, used for authentication with the Sentient Configuration Server.	Blank
interactive	interactive=[yes/no] Publish using interactive mode. Interactive mode allows you to choose whether or not to include required packages.	interactive=no

Setting	Expected Values	Default Value
strict	strict=[yes/no] Publish using strict mode. Strict mode will not publish packages missing required components.	strict=no
multiple	multiple=[yes/no] Display multiple versions of additional required packages,	multiple=no
rely_on_db	rely_on_db=[yes/no]	rely_on_db=no
requisite_depth	Set to 0 to include all dependencies or set to a number of levels.	requisite_depth=1
distribution_depot	Distribution depot path.	Blank

Publishing with Sentient Native Packaging

Refer to "[Sentient Native Packaging Command-Line Interface](#)" on page 89 for an explanation of the Sentient Native Packager command-line parameters.

To publish a specific Xchat RPM package residing in the specified depot on Red Hat Linux:

1. Change your current working directory to the Sentient Batch Publisher directory.
2. On the command line, type:

```
./rnp -d /home/rpadmin -p xchat-1.4.0.2.i386.rpm
```

Or simply:

```
./rnp -d /home/rpadmin -p xchat
```

Note: If a package is not supplied on the command line via the `-p` parameter, you will be presented with a list of all available packages within the specified depot.

Publishing with Interactive Mode

When you specify the parameter `-I` on the command line, the Sentient Native Packager interactive mode is invoked. This allows you to select which of the *available* required software you would like to include with your current package. You will also see which required prerequisite software is not available in the current depot.

The interactive mode option is ignored if neither the `-I` nor `-coreq` or `-i` parameters are specified on the command line (indicating prerequisite software is required for the current package). Here is an example of Interactive Mode:

Processing additional software required for QA_MASTER_1-1.0.0-0.i386.rpm

Following additionally required software is found in software depot and selected to be included in to promote package:

1. prereqs: - QA_RPM2-1.2.0-0.i386.rpm - included
2. prereqs: - QA_RPM3-1.0.0-0.i386.rpm - included
3. prereqs: - QA_RPM4-1.0.0-0.i386.rpm - included

Please toggle the selection:

Select (a to include all; d to exclude all; c to continue; s to skip current package; q to

quit entire session; a number to toggle its selection):

You can exclude any of the required software by entering the corresponding number. A message at the end of each line (included or not included) lets you know whether or not the required software will be included with the current package.

- Enter the number of the required software or type another option available in the interactive mode menu and press Enter to continue the native packaging process.

Interactive Mode Selections

Selection	Description
a	Selects all available required software to include with the current package. Available required software is included by default. (Set all available required software to included)
c	Continue the native packaging process. If you have made changes to the list of packages to be processed, the continue option will update this list. Use the continue option again to move on to the next processing step. This behavior accommodates multi-level dependency processing.
d	Deselects all included software. (Set all available required software to not included)
q	Quit the Sentient Native Packaging process.
s	Skip the current package.

Wrapped Native Packages

The following section lists all RCA-CSDB class instances and their attributes that are created when you publish native Linux software with Sentient Native Packaging.

Sentient Native Packaging utilizes a **method harness** to invoke agent methods, therefore when a package is published to the Sentient-CSDB, populated method attributes such as ZCREATE, ZDELETE, ZUPDATE, ZVERIFY, and ZREPAIR will contain the text "hide nvdkit method."

The supplied agent methods are designed to invoke the native software management utilities, therefore, the methods are specific for the agent platforms.

When publishing native Linux packages using the Sentient Native Publisher, the software packages are published to the RCA-CSDB (in compressed format). The following table lists the modified attributes:

RPM and DEB Class Instance Attributes

Attribute	Description
ZRSCNAME	Specifies a string that is used by native methods to identify software contained in the published depot. This is the RPM or DEB Package Name.
ZRSCCFIL	Specifies the path to the file that is included in this instance. This file contains the native packaged software.
ZCREATE	Uses method "Harness" call. The Sentient agent method <code>rpm.tcl</code> or <code>deb.tcl</code> (for Debian and Ubuntu) script contains a native command call to install the software package: <code>hide nvdkit method</code>
ZDELETE	Uses method "Harness" call. The Sentient agent method <code>rpm.tcl</code> or <code>deb.tcl</code> (for Debian and Ubuntu) script contains a native command call to remove the software package: <code>hide nvdkit method</code>
ZUPDATE	Uses method "Harness" call. The Sentient agent method <code>rpm.tcl</code> or <code>deb.tcl</code> (for Debian and Ubuntu) script contains a native command call to update the software package: <code>hide nvdkit method</code>
ZVERIFY	Uses method "Harness" call. The Sentient agent method <code>rpm.tcl</code> or <code>deb.tcl</code> (for Debian and Ubuntu) script contains a native command call to verify the installed software package: <code>hide nvdkit method</code>
ZREPAIR	Uses method "Harness" call. The Sentient agent method <code>rpm.tcl</code> or <code>deb.tcl</code> (for Debian and Ubuntu) script contains a native command call to repair the installed software package (reinstall): <code>hide nvdkit method</code>
PKGVER	Package Version. Informational attribute only.
PKGREL	Package Release. Informational attribute only. (Not applicable for Debian and Ubuntu)

Attribute	Description
PKGARCH	Package Architecture. Informational attribute only.
PKGSUMM	Package Summary. Informational attribute only.
REQPKGS	Required Packages. Informational attribute only.
REQCMDS	Required Commands. Informational attribute only.
REQLIBS	Required shared libraries. Informational attribute only.
CONTENTS	Required packages included in <code>tar</code> file.
PKGEPOCH	RPM or DEB Package EPOCH.
INSTOPTS	Package installation options. (For example, <code>--oldpackage</code> , <code>--replacepkgs</code>). NOCHECK keyword can be added to installation options. See Operational Notes for more information.
VRFOPTS	Package verify options. (For example, <code>--nomode</code> , <code>--nogroup</code>)

An instance of PACKAGE class is created that contains the instance of the RPM or DEB class. The following table describes how Sentient Native Packaging maps native package information into PACKAGE class attributes.

PACKAGE Class Attributes

Attribute	Description
Instance Name	For RPM or DEB, the <code>RPM_</code> or <code>DEB_</code> prefix is added to the RPM or DEB Package Name respectively, and date and sequence number is appended (<code>RPM_<PKG>_yyyymmddn</code> or <code>DEB_<PKG>_yyyymmddn</code>). Note: When instance names generated are longer than 32 characters, the package/patch names parts of the instance names shall be truncated.
RELEASE	The RPM or DEB version native attributes are mapped into RELEASE.
NAME	The RPM Packages, <code>RPM_</code> prefix is added to the RPM package name and suffixed with the package version , release and architecture (<code>RPM_<PKG>,ver=<VERSION>,rel=<RELEASE>,arch=<ARCH></code>). The DEB Packages, <code>DEB_</code> prefix is added to the DEB package name and suffixed with the package version and architecture (<code>DEB_<PKG>,ver=<VERSION>,arch=<ARCH></code>).
DESCRIPT	The RPM or DEB Packages, the package summary is mapped into DESCRIPT.
ZSTOP000	Contains an expression that contains target operating system information.

Attribute	Description
FILE	Holds reference to respective instance of RPM or DEB class.

Sentient Native Packaging also creates an instance of ZSERVICE Class holding previously created instance of PACKAGE Class. The following table lists the modified attributes.

ZSERVICE Class attributes

Attribute	Description
Instance Name	For RPM or DEB, the RPM_ prefix or DEB_ prefix is added to the RPM or DEB Package Name respectively, and a date and sequence number is appended (RPM_<PKG>_yyyymmddn or DEB_<PKG>_yyyymmddn). Note: When instance names generated are longer than 32 characters, the package/patch names parts of the instance names shall be truncated.
VERSION	For RPM or DEB Packages, the respective RPM or DEB Package Version is mapped into VERSION.
NAME	For RPM Packages, the RPM_ prefix is added to the RPM package name and suffixed with the package version, release and architecture (RPM_<PKG>, ver=<VERSION>, rel=<RELEASE>, arch=<ARCH>). For DEB Packages, the DEB_ prefix is added to the DEB package name and suffixed with the package version and architecture (DEB_<PKG>, ver=<VERSION>, arch=<ARCH>)
ZSVCNAME	For RPM or DEB, the respective RPM or DEB Package Name is mapped into ZSVCNAME.
VENDOR	Specifies vendor of the native Linux package.
ZSVCMO	Service is set to mandatory by default. Valid values of this attribute are: • M for mandatory • O for optional
ALWAYS	Holds reference to the respective instance of PACKAGE Class.

Note: If a package requires a system reboot after an agent connect, make sure the `hreboot radskman` parameter is set to `Y`. For more information, see ["RADSKMAN Examples" on page 485](#).

Automatic Inclusion of Required Packages

If you specify the `-i` command-line option, Sentient Native Packaging will include prerequisite packages into the depot with the (main) package you are publishing to Sentient. The prerequisite

package needs to exist in the depot Sentient Native Packaging is using as a source. For RPM packages, all prerequisite packages as well as any additional prerequisite packages they might require are included (this can be determined by using the `-depth` option).

When using the `-i` or `-coreq` options, the promotion of native software packages will not fail because of a missing prerequisite or corequisite package (unless the `-s` option is specified). Installation will fail only if prerequisite or corequisite packages are missing from both the promoted native software package *and* from the target machine.

Alternatively, if a prerequisite or corequisite package is already installed on the target machine, including these in a native software package for promotion will result only in using more network bandwidth and disk space than necessary.

Publishing packages with prerequisites included may take an extended amount of time. About every thirty seconds, a progress message is displayed:

```
Info:      Compiling extended info about all packages in the depot.  
Please wait...
```

Publishing and Deploying of Native Packages

The following describes the operations involved during the publishing and deployment of native packages. This gives you a better understanding of the current processes and capabilities provided to manage these packages.

Publishing

- All packages are selected from the software depot specified by using the `-d` option. It is recommended that you build a depot with packages of the same architecture only (for example, separate depots of RPMs for installation of i386 or x86_64 machines).
- “`-dist a_distribution_depot`” option (RPM packages bundles only) where `a_distribution_depot` is the location of packages used during deployment. If this option is present, then the `-d` option can be omitted and `a_distribution_depot` location will be used instead. Packages promoted with the `-dist` option will contain “DIST=distribution_depot_path” in the CONTENTS field of their package class instances.
- Dependency checking is performed on the target (selected) package as well as on all its dependent packages (currently, multi-level dependency checking is implemented for RPM). Use `-depth N` option to control the depth of dependency processing. If the `-depth` option is not defined, only one level of dependencies (of the target package only) are processed. To include all dependencies, use `-depth 0`.
- On Linux, a dependency’s release level can be specified as conditional (`>= version 2, release 1`). If multiple dependencies are found to satisfy this condition, by default the newest package is selected for inclusion. If a specific version is desired, one can use the `-M` option in interactive mode to list all possible matches, and select the one desired.
- Use the `-s` (strict) option to ensure that all identified dependencies are included in the deployment. If required dependencies are not found in the software depot, an error message will be displayed and publishing will be terminated.

- Using Interactive mode enables you to:
 - See all packages in the software depot available for selection
 - Review all dependencies found for a selected package.
 - Select / de-select dependencies. This allows administrators who have knowledge of their target machines to tailor the deployment to fit their environments and needs. Some dependencies can be large, and rather than waste bandwidth and client processing, if not needed, it can be removed from the deployment.

Deployment

- If a package was promoted with the `--dist` option, it will be installed from the distribution depot specified in the CONTENTS field.
- If the target package is already installed on the machine and is newer than the one to be deployed, no further processing is done, and the deployment is viewed as successful. However, since it was not deployed, it will not be removed when the service is deleted.

Note: If back leveling of the package is required, this behavior can be overridden by specifying the appropriate native command-line option in the attribute INSTOPTS for RPM packages. This requires the use of the Sentient-CSDB Editor.

- If (during installation) the target package already exists and is the same release level, it is first verified. If verification fails, it will be re-installed. Subsequent verify or delete processing would occur as usual. This behavior can be changed by adding NOCHECK keyword into INSTOPTS attribute of the package (this requires the use of the Sentient-CSDB Editor). If present, the package will be re-installed even if it does not fail verification. It is valid for RPM packages only. Appropriate options for the installation are still required.
- During install/update, the release levels of already installed dependencies are individually checked, and if newer on machine, they are not installed as this may cause conflicts for other packages. This behavior can be changed by adding the NOCHECK keyword to the INSTOPTS attribute of the package (this requires the use of the Sentient-CSDB Editor). For example, if INSTOPTS for an RPM package is set to "NOCHECK --oldpackage" or "NOCHECK --force", then required packages for the package can be back leveled; while without NOCHECK, only the main package can be back leveled.
- During verify, only the target package is verified and not its dependencies.
- After installation, the native package database is queried to make sure the target package was properly installed and registered in the database.
- During removal, the package is checked to make sure it exists (as it may have been upgraded or superseded). If it does not exist, no attempt to delete it is made, and the process is viewed as successful. Only the target package is deleted. Dependent packages are not deleted, as they may be required for other packages.
- If the verify attribute (ZRSCVRFY) of the package instance is set to N, the source depot (file actually deployed from server) is deleted after a successful installation. If a subsequent verification of the installed target package fails, this file is again downloaded and used to repair the damaged package.

Event Reporting

Use the RNPEVENT object to report events to the Sentient Configuration Server. Similar to the APPEVENT object, RNPEVENT uses the same variable set and is created if the Sentient administrator has enabled the reporting flags for a particular event in the EVENTS variable of the ZSERVICE Class. The RNPEVENT variables are listed in the table below.

RNPEVENT variables

Variable	Description	Sample Value
EVENT	Text description of the current event.	create
STATUS	Error messages.	Successful
CMDRC	Return code from native command.	0
CMDMSG	Message from native command.	Main package <Regina> on desktop <2.0> is newer than in CM-CS <1.0>. Skipping further processing.
POSTRC	Return code from RNP post-processing check.	0
POSTMSG	Message from RNP post-processing check	Post installation is successful
ZOBJDOMN	The domain name for the application.	SOFTWARE
ZOBJCLAS	The class name for the application.	ZSERVICE
ZOBJNAME	The instance name for the application.	RPM_GAIM_200504123
ZOBJID	The objects ID for the instance.	D123ACD45F67
ZUSERID	Sentient user that installed the application.	RPMUSER_LINUX
DELDATE	ISO8601 date time when the delete event occurred.	2010-05-10T16:45:00Z
VERDATE	ISO8601 date time when the verify event occurred.	2010-06-10T16:47:00Z
INSTDATE	ISO8601 date time when the install event	2010-07-10T16:44:00Z

Variable	Description	Sample Value
	occurred.	
FIXDATE	ISO8601 date time when the repair event occurred.	2010-08-10T16:43:00Z
UPGDATE	ISO8601 date time when the update event occurred.	2010-09-10T16:42:00Z
JOBID	Session identifier	MachineConnect
CJOBID	Session identifier	11122:3

Viewing Event Details

Use the Sentient Console to view the details of your Native Package Events. View the details of the Sentient Managed Service, then select the Sentient Native Package Events report. For details on using the Reporting Server, see *Sentient Enterprise Console Online Help*.

Chapter 3

Inventory Management

The Inventory Manager is an agent feature used to discover configuration information on remote computers. It enables centralized reporting and administration based upon the discovery results.

Systems administrators use the Sentient Enterprise Administrator Configuration Server Database Editor (CSDB Editor) to specify what inventory management data is to be collected. An Inventory connect (DNAME=AUDIT) is then run on the target computer to retrieve the desired information and send it up to the Sentient server for later reporting.

For more information on reporting, see the *Sentient Enterprise Console Online Help*.

The AUDIT Domain

The AUDIT Domain is located in the PRIMARY File of the Sentient Enterprise Configuration Server Database (CSDB) and contains the classes required to:

- Configure the tasks needed to collect the inventory information.
- Manage the agent computers' assets.

Note: The following figures and instructions use the CSDB Editor, which is available for 32-bit Windows platforms.

AUDIT Domain Defined

The AUDIT Domain is structured very much like the SOFTWARE Domain. The following table describes the classes present in the AUDIT Domain.

Audit Domain

Class	Description
Audit Application (ZSERVICE)	Sample services distributed with the Inventory Manager. The AUDIT.ZSERVICE instance is connected to a policy instance. A policy instance can be an instance of the Users, Departments, or Workgroups class. It can also be a customer-defined class within the POLICY Domain. Each of the sample ZSERVICE Classes is connected to the PACKAGE instances.
Audit Packages (PACKAGE)	Defines what information to collect and then what actions to take. These packages would contain various audit components. A good example is an audit of running services on a desktop. The AUDIT.ZSERVICE instance must contain a connection to an AUDIT.PACKAGE instance.
Behavior Services	Defines instances that enable the execution of auditing on the agent. Normally, there is no need to add or modify instances in this class.

Class	Description
(BEHAVIOR)	
Client Methods (CMETHOD)	Used to configure method points for Tcl inventory scans. The base instance of the SCANNER Class is connected to the CMETHOD.INV_FULL instance. This instance can be used for all inventory scans defined in the SCANNER Class.
Desktop (DESKTOP)	This class is reserved for future use.
File (FILE)	Defines file scans, such as auditing system executables/DLLs.
File Scanner (FILESCAN)	For UNIX devices only, persistent component class used to configure an inventory scan. Adding File Scanner components to an audit package creates instances of the FILESCAN class.
File Scanner Filters (FILTER)	For UNIX devices only, persistent component class used to configure an inventory scan. Adding File Scanner Filters components to an audit package creates instances of the FILTER class.
Inventory Options (RIMOPTS)	Contains the attributes that offer options to control an inventory management task. For additional information on Windows devices, see the section "RIMOPTS Class" on the next page .
Inventory Scanners (SCANNER)	This persistent component class is used to configure an inventory scan. Create instances of the SCANNER class by adding Inventory Scanners components to an audit package.
Path (PATH)	This class stores the drive and directory required to install a resource. Packages can be relocated by updating instances of this class.
Registry (REGISTRY)	Uses WMI to obtain a Registry scan of a Windows machine. Create instances of the REGISTRY class to run scans of the Windows Registry and obtain a Registry Scan report. For more information, see the REGISTRY Class .
Scheduling (TIMER)	Contains the instances that enable the Sentient administrator to set a timer on end users' computers. One or multiple auditing services can be processed whenever the timer expires.
UNIX Permissions (UNIXPERM)	Contains UNIX® file permission information.
Virtual Mgr Location (MGRVLOC)	Used to specify the initial path for files being transferred to the Configuration Server during a FILE audit.
WBEM (WBEM)	Contains instances that define Inventory Manager scans of WMI classes. These can include any class in the WMI database such as Win32_ Services.

FILTER Instances

Instance	Description
NAME	Friendly Name
ACTION	Action Flags: I – Initial (Used for file auditing only [not currently supported]) N – New C – Changed D – Deleted S – Send (upload to Configuration Server) D – Delete (not currently supported) C – Custom (not currently supported)
DIR	Directory to scan.
DEPTH	Number of subdirectory levels to scan Values: -1 root directory and all of its subdirectories 0 root directory only 1 root directory and its files >1 root directory and its files down to the specified depth
INCLUDE	Include globe pattern.
EXCLUDE	Exclude globe pattern.
COMPRESS	Compress [Y/N]
ZRSCVLOC	Name of an instance in the PRIMARY.AUDIT.MGRVLOC class that defines the location to place the uploaded scanned files. Default is RCA_UPLOAD.

RIMOPTS Class

The RIMOPTS Class is also known as the Inventory Options Class. This class contains the attributes that control an inventory management task. The following table describes these attributes.

RIMOPTS Class

Attribute	Usage
COLLECT	Audit Collection Type by selecting Diff or Full Select Diff to report the difference between the previous information collected for the service and the information collected during the current

Attribute	Usage
	agent audit. This is the default setting. Note: The first or initial scan of the DIFF setting will be a FULL scan as defined below. All subsequent scans will then be differenced unless the administrator changes the setting to FULL. Select Full to report the information collected for the service during the current agent connect process without differencing against the previous collection for that service.
RUNEXEC	Indicates what actions the Inventory Manager will take upon connection: - Select I to invoke collection of information when the service is installed - Select U to invoke collection of information when the service is updated. - Select V to invoke collection of information when the service is verified. The default settings are I and U.
ZSVCTYPE	Contains code that is used internally by the Inventory Manager agent. In all cases, this value should remain I.
NAME	Contains the friendly name of the instance. It is the name displayed for the instance in the tree view of the CSDB Editor.

To apply an option expressed in the RIMOPTS instance to the inventory management task, the RIMOPTS instance must contain a connection to an audit service.

Prior to beginning any tasks using the Inventory Manager, you must enable the drag-and-drop feature for the newly created RIMOPTS Class instances. For more information about editing instances, see the *Sentient Enterprise Console Online Help*.

To enable drag-and-drop connections for RIMOPTS Class instances:

1. Open the CSDB Editor and go to **PRIMARY > ADMIN > Name Lists (32) (ZLIST32) > CONNECT_ > CONNECT_ZSERVICE_**
2. Double-click **CONNECT_ZSERVICE_TO_RULES**.
3. The Editing Instance dialog box opens.
4. Set the value of the **ZNAME n** attribute to **RIMOPTS**.

The drag-and-drop feature is now available for all attributes in RIMOPTS.

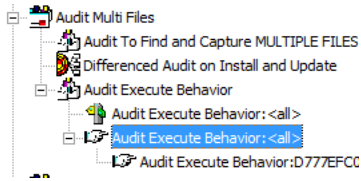
Audit Packages or PACKAGE Class

Once you are comfortable auditing using the sample packages provided by Sentient, you will probably want to take the next step in designing your own audit packages.

By expanding the Audit Packages (PACKAGE) class, you will see the audit package instances.

A complete audit service consists of several connected instances in the AUDIT Domain. The audit package instance is a container that "owns" the instances connected to it. For example, open the AUDIT.ZSERVICE class and double-click the **Individual File Audit** instance.

Individual File Audit Instance



In the example, the Individual File Audit ZSERVICE instance "owns" the Audit to Find and Capture a Local File instance. The fact that a package instance owns a component class instance means that all of the instances are managed as a package unit. If the package instance is deleted, all of its owned class instances are automatically deleted as well.

Caution: Sound database management practices dictate that the component class instances owned by a package are not connected to any other package instance.

The audit service instance must also contain a connection to an instance of the RIMOPTS Class. Connecting an instance of the RIMOPTS Class to an audit service instance causes the expressed behavior to be performed. Specified behaviors are listed in the following table.

Inventory Options (RIMOPTS) Class

Instance	Description
Default	Contains the base instance attributes for the RIMOPTS Class. - Collect attribute is set to <code>Diff</code>. - Runexec attribute is set to <code>IU</code>. - ZSVCTYPE attribute is set to <code>I</code>.
Differenced Audit on Install and Update	When connected to an audit service will difference the audited information on installation and when the audited target is updated. - Collect attribute is set to <code>Diff</code>. - Runexec attribute is set to <code>IU</code>. - ZSVCTYPE attribute is set to <code>I</code>.
Differenced Audit on Install, Verify, and Update	When connected to an audit service, will difference the audited information in initial installation, on subsequent connects, and when updated. - Collect attribute is set to <code>Diff</code>. - Runexec attribute is set to <code>IVU</code>. - ZSVCTYPE attribute is set to <code>I</code>.
Full Audit on Install and Update	When connected to an audit service, will difference the audited information on installation and update. - Collect attribute is set to <code>Full</code>. - Runexec attribute is set to <code>IU</code>. - ZSVCTYPE attribute is set to <code>I</code>.

Instance	Description
Full Audit on Install, Verify and Update	When connected to an audit service, will - Collect attribute is set to <code>Full</code>. - Runexec attribute is set to <code>IVU</code>. - ZSVCTYPE attribute is set to <code>I</code>.

Finally, a connection to an auditing behavior is needed.

The audit behavior owned by the Individual File Audit ZSERVICE is connected to the Behavior Services (BEHAVIOR) class within the AUDIT Domain.

The BEHAVIOR class in the AUDIT Domain remains unchanged from the BEHAVIOR class within the SOFTWARE Domain. For description of the attributes found within this class, see "Behavior Services (BEHAVIOR) Class" on page 1195 .

REGISTRY Class (Applicable for Windows Systems)

Table below summarizes the attributes and values for the Registry Class instances. Attributes in bold are new to this class (not in the WBEM class).

Registry Class Instance Attributes

Attribute	Description	Default Value	Valid Values
ACTION	Report Flags (I, N, C, D, S, D, C)	YYYYXXN	Y, X, or N for each flag.
NAMESPACE	Name Space	root\default	root\default – Do not change.
CLASS	WBEM Class	StdRegProv	StdRegProv – Do not change.
RPTCLASS	Report Class Name	Registry	A valid table name. If blank "StdRegProv" will be used.
PROPERTY	Registry hive	HKEY_LOCAL_MACHINE	Any Windows registry hive: HKEY_CLASSES_ROOT HKEY_CURRENT_USER HKEY_LOCAL_MACHINE HKEY_USERS HKEY_CURRENT_CONFIG HKEY_DYN_DATA

Attribute	Description	Default Value	Valid Values
CNDITION	Registry subkey	SOFTWARE\Microsoft\Internet Explorer	Any Windows registry subkey.
FORMAT	Output format	REGISTRY	REGISTRY – Do not change.
DEPTH	Starting at the registry subkey named in the CNDITION attribute, depth specifies the number of descendent key levels to include in the scan.	0	0, -1, or <i>n</i> Set to 0 to only scan current subkey. Set to -1 to scan all subkey levels. Set from 1 through <i>n</i> to scan the current subkey and the specified number of subkey levels deep.
OUTPUT	Output Object Name	WBEMAUDT	WBEMAUDT
TYPE	Scan Type (WBEM)	WBEM	WBEM – Do not change.
NAME	Friendly Name	Default	Friendly name for this instance displayed in the CSDB Editor.

Software and Hardware Auditing

This chapter assists you with installing and implementing the Inventory Manager. Choose the appropriate strategies suited for your enterprise needs.

CIM Schema and Inventory Collection

As a guide for collecting hardware and software inventory, Persistent Systems uses the Common Information Model (CIM) schema version 2.6. This allows inventory to be collected based on industry standards, as defined by the Distributed Management Task Force (DMTF).

The CIM schema allows real-world objects to be mapped to objects defined in the different schema classes and attributes. After data is discovered using these standards, the output is collected by Sentient and is available for reporting purposes.

For a description of the CIM schema classes used, see the following table CIM Classes.

CIM classes

CIM Class	Description
CIM_SCSIController	Subclass of the CIM_Controller used to represent SCSI controllers.
CIM_ResidesOnExtent	Subclass of CIM_Dependency. This is an association between the logical volume and the file system on the logical volume.
CIM_Processor	Used to represent computer processor information.
CIM_ParallelController	Subclass of CIM_Controller used to represent parallel controllers.
CIM_NFS	Used to represent general information about NFS mounted file systems.
CIM_MediaPresent	Used to represent relationship with the MediaAccessDevice. Represents logical volume or volume group and one of the disks it resides on.
CIM_LogicalDiskBasedOnVolume	Subclass of LogicalDiskBasedOnExtent used to represent the relationship between logical volume and its volume group.
CIM_LogicalDisk	Used to represent general information about the logical volume.
CIM_IDEController	Subclass of CIM_Controller used to represent IDE controllers, including ATA and ATAPI controllers.
CIM_EthernetAdapter	Used to represent capabilities of the Ethernet card.
CIM_DiskDrive	Subclass of CIM_MediaAccessDevice, includes all hard disk drives, non-removable and removable. Models the reader/writer properties of disk drives.
CIM_Directory	Used for exported directory.
CIM_DVDDrive	Subclass of CIM_MediaAccessDevice includes all of the types of DVD reader and writer drives.
CIM_CDRomDrive	Subclass of CIM_MediaAccessDevice includes CDRom reader and writer drives.
CIM_Service	Used to represent general information about NFS client/server service.
CIM_SCSIInterface	Subclass of CIM_ControlledBy. Represents unique data from the relationship between the controller and the device.
CIM_UnixLocalFileSystem	Used to represent UNIX specific information about the local file system.

CIM Class	Description
CIM_UnixComputerFileSystem	Used to represent general information about the computer.
CIM_StorageVolume	Used to represent the hand-off point between providers or the result of a redundancy.
CIM_UnixOperatingSystem	Used to represent general information about the UNIX operating system. General information about the volume groups.
CIM_SoftwareElement	Used to represent the SVR4 packages or filesets.
CIM_Export	Used to represent an association between a LocalFileSystem and its directories indicating that the specified directories are available for mount. When exporting an entire FileSystem, the directory should reference the topmost directory of the FileSystem.

For more information about the CIM schema 2.6, visit the DMTF Web site:

<http://www.dmtf.org/>

Auditing Types

When configuring your audits, the administrator should understand exactly what types of things can be audited and what the expected results from an audit will comprise.

The Inventory Manager allows for three types of audits:

- File auditing
- WBEM auditing
- Hardware auditing

File Auditing

The AUDIT.FILE class instances in an audit package control the auditing function for files on the agent computer. The RIMFSCAN and the RIMDIFF methods on the agent computer perform the actual file auditing operations by specifying what files to look for. There can be one or more AUDIT.FILE instances in an audit package. Each AUDIT.FILE instance can specify a scan for one or more files.

The following table summarizes the attributes in an AUDIT.FILE class instance and their effects on the RIMFSCAN method.

AUDIT.FILE Class Instances

Attribute	Description Examples
SCANFOR	Indicate a fully qualified path and file name to search for. Wildcards are permitted.

Attribute	Description Examples
	Example for Windows: <SystemDrive:>\WinNt**.dll
ACTION	The RIMDIFF method performs actions on the files discovered on the user's computer during the agent connect. • Y configures RIMDIFF to perform the action. • N configures RIMDIFF to not perform the action. The first four flags determine when to report that the files were found: Report on: Initial, New, Changed, Deleted • Initial means that the file was found during the first scan of the agent computer. • New means that the file was found during the current scan. The file was not present during the previous scan. • Changed means that the file was present during the previous scan and is different from the file found during the current scan. • Deleted means that the file was found during the previous scan. The file is not present for the current scan. The last three flags control the actions to take on the files detected during the current scan. Action to take on discovery: Send, Delete, Custom • Send means to send the files to the Configuration Server and store them in the location indicated by the ZRSCVLOC attribute (see ZRSCVLOC in this table). • Delete means to delete the files from the user's computer. • Custom means to execute the method indicated in the CUSTOM attribute. YYYYNYN – Report whenever encountered and delete the files. NNYYNNN – Report when changed or deleted and take no action. NYYNYYN – Report when the files are new or changed. Then send and delete the files.
OUTPUT	Output object name.
TYPE	Scan different file locations. Available scans are Behavior Services, Desktop, File, Path, Registry, and WBEM. File.
GROUP	Optional way to identify a set of scan results. This maybe useful for querying and reporting on the audited files from the database where audit results can be stored. Games, MPEGs.
ZVERINFO	Collect extended information.

Attribute	Description Examples
	- Set the value to 1 to collect more information for a file. - Set the value to 0 to not collect more information. In order for this data to be collected, the associated attribute must exist in the AUDIT.FILE class template. You can limit the scan to only those files that have some particular values in their extended information. You do so by supplying a value (either 1 or 0) for any of the associated attributes in an AUDIT.FILE instance. This causes the scan to be filtered. Only those files whose extended information data element contains the value you specify in its associated attribute will be scanned. Extended file information consists of one or more of the following data elements. The associated attribute name for the data element is in parentheses: (VENDOR) - The seller of the file/product (PRODUCT) - The name of the item for which the file is a part. (PRODVERS) - The version of the product which the file is a part. (ORGNAME) - The name of the organization. (INTERNAL) - The internal data element encoded in the file. (VERSION) - The version of the file. (LANGUAGE) - The language of the file.
ZRSCSTYP	Server file type. This can be either Binary or Text. The administrator does not set this.
ZRSCMFIL	Manager directory location.
ZRSCVLOC	The location on the Configuration Server where the files are stored because of the Send Action (see ACTION in this table). This variable needs to be configured when sending a file back to the Configuration Server. The variable should contain the name of the MGRVLOC instance that will be used to resolve the location to store the uploaded file. <SystemDrive:>\Data\&(ZOBJPID) \&(name) , for Windows systems.
ZRSCMMEM	PDS member name. This field is optional.
PRODUCT	The product name. See "ZVERINFO" on the previous page for more detail.
PRODVERS	The product version. See "ZVERINFO" on the previous page for more detail.
ORGNAME	The organization name. See "ZVERINFO" on the previous page for more detail.

Attribute	Description Examples
INTERNAL	The internal data element encoded in the file. See "ZVERINFO" on page 114 for more detail.
VERSION	The version of the file. See "ZVERINFO" on page 114 for more detail.
LANGUAGE	The language of the file. See "ZVERINFO" on page 114 for more detail.
VENDOR	The product vendor. See "ZVERINFO" on page 114 for more detail.
ZRSCCRC	Resource CRC.
ZCRCINFO	Collect file CRC. [Y/N] Default is N . - Set the value to Y to collect CRC information for a file. - Set the value to N to not collect CRC information. - If blank, defaults to N. Caution: Collecting file CRC information can dramatically extend the time it takes to collect information on the target machine.
ZRSCOBJN	Persistent object name.
ZRSCPADM	Administrator ID.
ZRSCSRC	Resource Source, i.e. Publisher.
ZINIT	Not applicable at this time.
NAME	Not applicable at this time.
LOCATION	Not applicable at this time.
ZMD5INFO	Collect file MD5 information. [Y/N] MD5 information is a 32-character value that can be used to uniquely identify a file based on its content. Default is N . - Set the value to Y to collect MD5 information for a file. - Set the value to N to not collect MD5 information. - If blank, defaults to N. Caution: Collecting MD5 information can dramatically extend the time it takes to collect information on the target machine.

Use the Admin Agent Explorer to view the FILEPREV object results.

The FILEPREV object contains one heap for each file discovered during the scan for the audit service. It contains the attributes from the AUDIT.FILE class instance that controlled the scan, as described above. It also contains the following attributes:

FILEPREV Object

Attribute	Description
ACTION (only for Unix systems)	Action flags. First four flags determine when to report. Y – ignored Y – New file Y – File changed since last scan Y – Ignored Last three flags control action to be taken. Y – send the file to RCS Y – ignored Y – ignored
ACCESSDT (for Unix and Windows systems)	The date of the most recent access of this file.
ACCESSTM (for Unix and Windows systems)	The time of the most recent access of this file.
ATTRIBUTE (only for Windows systems)	A string listing the attributes of the file: R = Read only A = Archive S = System H = Hidden C = Compressed
COMPRESS (only for Unix systems)	Compression setting.
DATA_CRC (only for Unix systems)	Data CRC
DATE (for Unix and Windows systems)	The date of the most recent modification to this file.

Attribute	Description
DIR (for Unix Systems) DRIVE (for Windows systems)	System drive location of the file.
DIRPATH (for Unix and Windows systems)	The directory path of the file.
EXCLUDE (for Unix systems)	Parameter to exclude.
FULLPATH (for Unix and Windows systems)	Fully qualified path and file name of the file.
GID (only for Unix systems)	Unix group ID of file owner.
GIDNAME (only for Unix systems)	Unix group name of file owner.
INCLUDE (only for Unix systems)	Parameter to include.
NAME (only for Unix systems)	File name.
PATHCRC (for Unix and Windows systems)	A unique number that indicates the CRC path used for differencing.
PERMISS (only for Unix systems)	4-digit octal value for file permissions.

Attribute	Description
RESOLVE (only for Windows systems)	The value of M indicates that the Configuration Server resolves each heap of the FILEAUDT object individually. This value cannot be modified.
SIZE (for Unix and Windows systems)	File size in bytes.
STATUS (only for Windows systems)	Indicates the status of the file on the agent computer. Possible values are: • Exists means that this is the first time scanning for this file and it was found. • New means that this file was added to the file system of the agent computer since the last scan was performed. • Update means that this file exists in the new and previous scans. There have been changes to the date, time, size, and/or version. • Deleted means that this file was present in the previous scan but is missing in the new scan. • Not found means that no files were found that matched this request.
TIME (for Unix and Windows systems)	The time of the most recent modification to this file.
TYPE (only for Unix systems)	File type. Can be directory, LINK, or binary.
UID (only for Unix systems)	UNIX ID of file owner.
UIDNAME (only for Unix systems)	Username of the file owner.
XTENSION	The file extension. This is useful for sorting and querying back-end database tables that store the data

Attribute	Description
	found in this object.
ZOBJDATE (only for Unix systems)	Date
ZOBJPCLAS (only for Unix systems)	Class
ZOBJCID (only for Unix systems)	Object Child ID
ZOBJPNAM (only for Unix systems)	Unique Name
ZOBJTIME (only for Unix systems)	Time
ZRSCVLOC (only for Unix systems)	Location

The FILEAUDT object contains one heap for each file discovered during the scan for the audit service. It contains the attributes from the AUDIT.FILE class instance that controlled the scan, as described above. It also contains the following attributes:

FILEAUDT Object

Attribute	Description
ACCESSDT	The date of the most recent access of this file.
ACCESSTM	The time of the most recent access of this file.
ATTRIBUTE	A string listing the attributes of the file: R = Read only A = Archive S = System H = Hidden C = Compressed
DATE	The date of the most recent modification to this file.
DIRPATH	The directory path of the file.
DRIVE	The system drive location of the file.
FULLPATH	Fully qualified path and file name of the file.

Attribute	Description
PATHCRC	A unique number that indicates the CRC path used for differencing.
RESOLVE	The value of M indicates that the Configuration Server resolves each heap of the FILEAUDT object individually. This value cannot be modified.
SIZE	File size in bytes.
STATUS	Indicates the status of the file on the agent computer. Possible values are: • Exists means that this is the first time scanning for this file and it was found. • New means that this file was added to the file system of the agent computer since the last scan was performed. • Update means that this file exists in the new and previous scans. There have been changes to the date, time, size, and/or version. • Deleted means that this file was present in the previous scan but is missing in the new scan. • Not found means that no files were found that matched this request.
TIME	The time of the most recent modification to this file.
XTENSION	The file extension. This is useful for sorting and querying back-end database tables that store the data found in this object.

Audit.FILESCAN

Note: UNIX file auditing using `filesScan.tkd` is supported for legacy purposes only. New installations of the Inventory Manager should use RIMFSCAN and RIMDIFF, described in the section AUDIT.FILE, above.

Note: Resolution of the `filesScan.tkd` service is not supported over SSL.

The AUDIT.FILESCAN class instances in an audit package control the auditing function for files on the agent computer. The *filesScan.tkd* methods on the agent computer perform the actual file auditing operations by specifying what files to look for. There can be one or more AUDIT.FILESCAN instances in an audit package. Each AUDIT.FILESCAN instance can specify a scan for one or more files.

See ["Audit Scan Results" on page 137](#) for more information on the `filesScan.tkd` methods.

The following table summarizes the attributes in an AUDIT.FILESCAN class instance and their effects on the `filesScan.tkd` method.

AUDIT.FILESCAN Class Instances

Attribute	Description
NAME	Friendly name.
DIFF	Specifies if differencing is to be done or not. If DIFF = Y, then the information from the scanned files will be compared with the information from the previous file scan.
OUTPUT	Specifies the prefix to be used for the object names created. If OUTPUT=FILE, then FILEAUDIT, FILEPREV objects will be created on the agent computer.

WBEM Auditing

Use the RIMWBEM method to query the WBEM namespaces to retrieve information about how a system's hardware and software is used. The RIMWBEM method constructs a query from the information contained in an instance of the AUDIT.WBEM Class. WBEM has a query engine that processes the query statement and returns the query results to RIMWBEM. There is one heap in the query result object for every discovered instance.

Caution: Inventory Manager leverages Microsoft's Windows Management Instrumentation (WMI) to collect hardware and software inventory data by using WMI queries. Some WMI queries can traverse the network contacting other servers in the enterprise to collect the requested information. This may result in large volumes of data being returned, and could have a significantly negative effect on network performance. An example of this would be querying all users on the network using the **W32_UserAccount WMI** class. Extreme caution must be taken to understand the scope of these queries to ensure unexpected results do not occur. While Inventory Manager provides an interface to WMI and its providers, it cannot control how these queries are satisfied. It is the customer's responsibility to safeguard against using WMI queries that span the network, if this behavior is not as expected.

An AUDIT.WBEM class instance defines a query into the WBEM namespace.

The following table describes the attributes of the AUDIT.WBEM instance.

AUDIT.WBEM Instance

Attribute Name	Description
ACTION	RIMDIFF method performs actions on the WBEM namespaces (s) instances discovered on the user's computer during the agent connect. • Y configures RIMDIFF to perform the reporting action. • N configures RIMDIFF to not perform the reporting action. The first four flags determine when to report that the WBEM namespace instance was found: Report on: Initial, New, Changed, Deleted, Scan, Delete, Custom • Initial means that the file was found during the first scan of the agent computer.

Attribute Name	Description
	• New means that the file was found during the current scan. The file was not present during the previous scan. • Changed means that the file was present during the previous scan and is different from the file found during the current scan. • Deleted means that the file was found during the previous scan. The file is not present for the current scan. • Scan means that the file was found during the current scan. • Delete means that the file was found during the previous scan. The file is not present for the current scan. • Custom means that the file was found during a custom scan. The last three flags are not applicable to WBEM audits.
NAMESPACE	Name of the WBEM namespace to query or HARDWARE.
CLASS	Name of the WBEM Class to query or HARDWARE.
PROPERTY	Specify one or more property names to be queried and reported. Use commas to separate more than one property name. If this attribute is blank, all properties in the class will be queried and reported.
CNDITION	An optional condition to narrow results of an audit.
OUTPUT	This is the name of the object to send to the Configuration Server.
TYPE	Indicates that WBEM scan is to be employed for this audit package.
NAME	Friendly name for this instance. This name will appear in the CSDB Editor tree view to identify this instance.

Note: When the keyword **HARDWARE** is used in the **NAMESPACE** and/or **CLASS** attributes of **AUDIT.WBEM**, hardware information is collected. This information is essentially the same as the **ZCONFIG** object.

The agent stores the results of a WBEM scan in a WBEM object. This object can be found in the service node of the client object tree. The results are also sent to the Configuration Server.

In addition to the attributes described above, the WBEM object also contains the following:

WBEM object attributes in the agent

Attribute	Description
ZOBJCID	Object child ID.
ZOBJCLAS	Targeted class for the audit such as ZRSOURCE or ZSERVICE.

Attribute	Description
ZOBJCRC	CRC of all persistent and transient objects under the current node.
ZOBJDATE	Last date under the current node.
ZOBJDOMN	Domain name of the object.
ZOBJID	Object ID of the instance used to obtain information from the Resource file.
ZOBJNAME	Instance name of the object.
ZOBJPCLS	Parent class name.
ZOBJPID	Parent class ID.
ZOBJRCRC	Resource CRC maintained by the Configuration Server.
ZOBJRSIZ	Resource size maintained by the Configuration Server.
ZOBJTIME	Latest time under the current node.
ZRSCSRC	Name of the program promoted the resource.

Manual Scanning Using RIMWBEM

Note: Start CIM Server before using the RIMWBEM.

RIMWBEM can be run from the command line to manually scan for a particular WBEM Class using the following syntax:

```
./rimwbem class=CIM_ComputerSystem
```

The example above will scan for CIM_ComputerSystem. Replace this CIM provider name with any WBEM Class information for which you want to manually scan. After the scan, check `$IDMSYS/log/rimwbem.log` and `$IDMSYS/lib/WBEMCURR.EDM` for the results.

To verify the results of the scan, run a custom query using a CIM navigator program (for example, CimNavigator).

NVDCIM.TKD

Note: UNIX WBEM auditing using `nvdcm.tkd` is supported for legacy purposes only. New installations of the Inventory Manager should use RIMWBEM, described in the section WBEM Auditing, above.

The `nvdcm.tkd` method is used to query the WBEM namespaces to retrieve information about a system's hardware and software. The method constructs a query from the information contained in an instance of the AUDIT.WBEM Class. WBEM has a query engine that processes

the query statement and returns the query results to `nvdcim.tkd`. There is one heap in the query result object for every discovered instance.

An `AUDIT.WBEM` class instance defines a query into the WBEM namespace.

The following table describes the attributes of the `AUDIT.WBEM` instance.

AUDIT.WBEM Instance

Attribute	Description
ACTION	The <code>filescan.tkd</code> method performs actions on the WBEM namespaces (s) instances discovered on the user's computer during the agent connect. • Y configures <code>filescan.tkd</code> to perform the reporting action. • N configures <code>filescan.tkd</code> to not perform the reporting action. The first four flags determine when to report that the WBEM namespace instance was found: Report on: <code>Initial</code>, <code>New</code>, <code>Changed</code>, <code>Deleted</code> • <code>Initial</code> means that the file was found during the first scan of the agent computer. • <code>New</code> means that the file was found during the current scan. The file was not present during the previous scan. • <code>Changed</code> means that the file was present during the previous scan and is different from the file found during the current scan. • <code>Deleted</code> means that the file was found during the previous scan. The file is not present for the current scan. The last three flags are not applicable to WBEM audits.
NAMESPACE	Name of the WBEM namespace to query or <code>HARDWARE</code> .
CLASS	Name of the WBEM Class to query or <code>HARDWARE</code> .
PROPERTY	Specify one or more property names to be queried and reported. Use commas to separate more than one property name. If this attribute is blank, all properties in the class will be queried and reported.
CNDITION	An optional condition to narrow results of an audit.
OUTPUT	This is the name of the object to send to the Configuration Server.
TYPE	Indicates that WBEM scan is to be employed for this audit package.
NAME	Friendly name for this instance. This name will appear in the CSDB Editor tree view to identify this instance.

Note: When the keyword **HARDWARE** is used in the **NAMESPACE** and/or **CLASS** attributes of **AUDIT.WBEM**, hardware information is collected. This information is essentially the same as the **ZCONFIG** object.

The agent stores the results of a WBEM scan in a WBEM object. This object can be found in the service node of the client object tree. The results are also sent to the Configuration Server.

The WBEM object contains more attributes described in the following table.

WBEM object attributes

Attribute	Description
ZOBJCID	Object child ID.
ZOBJCLAS	The targeted class for the audit such as ZRSOURCE or ZSERVICE.
ZOBJCNUM	Number of children under current instance.
ZOBJCRC	The CRC of all persistent and transient objects under the current node.
ZOBJDATE	The last date under the current node.
ZOBJDOMN	The domain name of the object.
ZOBJID	The object ID of the instance used to obtain information from the Resource file.
ZOBJNAME	The instance name of the object.
ZOBJPCLS	The parent class name.
ZOBJPID	The parent class ID.
ZOBJRCRC	The resource CRC maintained by the Configuration Server.
ZOBJRSIZ	The resource size maintained by the Configuration Server.
ZOBJTIME	The latest time under the current node.
ZRSCSRC	The name of the program promoted the resource.
ZUNUSED1	For future use.

WBEM Object Processing

When the Inventory Manager agent sends a WBEMAUDT object to the Configuration Server, processing is defined as follows:

1. When a WBEMAUDT object is found, the Configuration Server ZTASKEND calls `QMSG.EXE`.
2. `QMSG.EXE` places the WBEMAUDT objects into the Configuration Server's `\data\wbem` directory, or message queue.

3. The Messaging Server includes a WBEM Data Delivery Agent (WBEM.DDA) that monitors this `\data\wbem` message queue and processes the WBEM objects.
4. The WBEM.DDA is usually configured to post the WBEM objects directly to an ODBC-compliant Inventory Manager database, or, it may be configured to first forward the WBEM objects to another Messaging Server located closer to the database. In the later case, the receiving Messaging Server posts the WBEM data to the Inventory ODBC-compliant database.
5. Once posted to the Inventory database, the new WBEM information is immediately available for query and reporting purposes through the Reporting Server.

Disabling Remnant Configuration Server instances for WBEM Object Processing

Inventory Manager no longer supports processing WBEM objects using these instances in the Configuration Server database:

- `SYSTEM.PROCESS.WBEMAUDT`
- `SYSTEM.ZMETHOD.POST_WBEM`

If these remnant instances exist or were imported into your Configuration Server database, you must disable any configurations within them in order to ensure successful WBEM object processing.

Edit `SYSTEM.PROCESS.WBEMAUDT` and remove any connection to the `SYSTEM.ZMETHOD.POST_WBEM` instance.

Hardware Auditing

Each time an agent connects to the Configuration Server, information about the subscriber's hardware configuration is stored in the ZCONFIG object. The ZCONFIG object is calculated and stored in the application service directory of the agent's object directory tree.

A separate ZCONFIG object is calculated and stored for each service installed or updated during the agent connect process.

To force the transfer of the hardware information, the ZCONFIG variable must be set to Y in the `POLICY.USER` class (see figure below). To change this, use the CSDB Editor, which is available for 32-bit Windows platforms.

The ZCONFIG object stores information about the agent computer's hardware.

The ZCONFIG object stores hardware information discovered by the agent's standard hardware auditing method. Certain types of hardware can occur multiple times. The ZCONFIG object automatically expands to allow more information to be stored.

The following table describes the variables that are stored in a sample ZCONFIG object.

Attributes in a Sample ZCONFIG (for Unix)

Attribute	Description	Example
DESCRIPT	Internal use only	Processing Client Request for &ZCUROBJ

Attribute	Description	Example
IPADDR01	IP address of network adapter 1	1.1.1.99
LADAPT01	LAN Adapter 1	02608C2CBDCE
LANNUM	LAN Number	1643292
OSREV	Operating System revision number	4
OSVER	Operating System Version	3
ZHDWCPU	CPU Type	000019131C00
ZHDWD00	Drive Name for Drive 00	/dev/hd4
ZHDWD00F	Current free space on drive 00	7028736
ZHDWD00M	Drive 00 mount	/
ZHDWD00T	Total space for drive 00	25165824
ZHDWD01	Drive name for drive 01	/dev/hd2
ZHDWD01F	Current free space on drive 01	15859712
ZHDWD01M	Drive 01 mount	/usr
ZHDWD01T	Total space for drive 01	1577058304
ZHDWD02	Drive name for drive 02	/dev/hd9var
ZHDWD02F	Current free space on drive 02	2973696
ZHDWD02M	Drive 02 mount	/var
ZHDWD02T	Total space for drive 02	16777216
ZHDWD03	Drive name for drive 03	/dev/hd3
ZHDWD03F	Current free space on drive 03	28729344
ZHDWD03M	Drive 03 mount	/tmp
ZHDWD03T	Total space on drive 03	41943040
ZHDWDNUM	Number of drive letters assigned	9
ZHDWIPAD	IP address	&(IPADDR01)
ZHDWLANA	LAN Adapter	&(LADAPT01)
ZHDWMEM	Total physical memory (RAM)	65536
ZHDWOS	Operating system and version	Red Hat
ZHDWXHID	Host ID	0x1010163

Attribute	Description	Example
ZHDWXHN	Host name	linuxdemo
ZOBJRRC	Resolution return code	000
ZOBJRSTY	Resolution type	C
ZSRCCLAS	Service class	ZCONFIG
ZSRCCRC	Service CRC	8B37472C
ZSRCDATE	Service date	20001211
ZSRCDOMN	Service domain	SYSTEMX
ZSRCNAME	Service name	HARDWARE_SCAN
ZSRCPID	Service parent ID	0000000000
ZSRCTIME	Service time	11:52:59
ZUSERID	User ID	jdoe

Sample Attributes in ZCONFIG Object (for Windows)

Attribute	Description
GATEWAY	Router for your subnet.
HALCOMP	Company of HAL.DLL
HALDATE	Date and time of HAL.DLL
HALFNAME	Original name of HAL.DLL
HALFVER	Internal version of HAL.DLL
HALINAME	Name of HAL.DLL
HALLANG	Language of HAL.DLL
HALPNAME	Product name of HAL.DLL
HALPVER	Product version of HAL.DLL
HALSIZE	Size of HAL.DLL
IPADDR##	IP address of network adapter (there can be multiple addresses)
LADAPT##	Network card (there can be multiple network cards)
REBOOTD	Last reboot date
REBOOTT	Last reboot time

Attribute	Description
SUBNET##	Subnet mask.
ZGATEWAY	Looks at GATEWAY attribute
ZHDWBIOS	BIOS type.
ZHDWCDDR	Client Automation agent's CD-ROM drive letter
ZHDWCOMP	Computer name
ZHDWCPU	Current CPU type
ZHDWFPU	Current FPU type
ZHDWIPAD	The IP address of the computer
ZHDWKYBD	Keyboard type
ZHDWLANA	LAN adapter
ZHDWLANG	Language setting
ZHDWMEM	Total physical memory (RAM)
ZHDWMEMF	Total free memory (RAM)
ZHDWMOUS	Mouse type
ZHDWNET#	Network card information (can be multiple cards)
ZHDWNNET	Number of network cards
ZHDWOS	Computer's operating system and version
ZHDWOSCL	Operation system classification (Workstation or Server)
ZHDWOSDB	Operating system's build number
ZHDWOSOG	Organization
ZHDWOSOW	Owner
ZHDWOSSR	Windows 9x Sub-Version Number (i.e., A, B, C)
ZHDWPA##	Printer information
ZHDWPPAR	Number of parallel ports
ZHDWPPRN	Number of printers available
ZHDWPSE	Number of serial ports
ZHDWVIDEO	Video type
ZHDWVMSI	MSI Version

Attribute	Description
ZHDWVRES	Video resolution
ZHDWXPAG	Page size
ZHWCPU01	CPU type
ZHWFP01	FPU type
ZMODEM	Modem present? Y or N
ZOBJDATE	The date of the Client Automation agent connect for this service
ZOBJNAME	HARDWARE_SCAN (hard coded)
ZOBJTIME	The time of the agent connect
ZSUBNET	The subnet mask
ZUSERID	The name of the user who connected

Whenever an agent connects to the Configuration Server, certain subscriber hardware information is automatically forwarded to the Inventory Manager ODBC database as part of the Messaging Server processing of CORE objects. Use the Reporting Server to view hardware information.

Configuring Auditing

This section is provided to assist you with implementing and using the Inventory Manager. Choose the appropriate strategies suited for your enterprise needs.

Sample Auditing

To illustrate the concepts of inventory information collection, the Inventory Manager installation contains a set of representative audit service examples. These samples are located in the PRIMARY.AUDIT.Audit Application (ZSERVICE) Class. To view these, use the CSDB Editor, which is available for Windows platforms.

These sample services represent common scenarios for inventory collection and management. The best way to develop your own audit services is to study these samples.

The sample audit services are described in the following table.

Sample of Auditing Services

Service	Connected to Audit Package (PACKAGE)	Description
_BASE_INSTANCE_		This service instance is the base instance for the Audit Application (ZSERVICE) Class.

Service	Connected to Audit Package (PACKAGE)	Description
Audit Multi Files	Audit to find and Capture Multiple Files	This service scans for a file name or pattern and reports that information back to the administrator.
CE PDA XML Inventory	CE PDA XML Inventory	This service scans for and reports back information on installed Windows CE PDA devices. Will only report back if a device is found.
Delete Discovered Application Component	Audit to Find and Remove Local File	This service looks for a specific file on the user's computer. If it is found, it will be deleted.
Discover Product Licenses	Discover Product Licenses	This service collects license information for installed Microsoft products on managed devices.
Individual File Audit	Audit to Find and Capture Local File	This service performs an NVDM scan of the user's computer for a specified file of an instance of the AUDIT.FILE classes.
NVDM Discovery of Applications	NVDM Discovery of Applications	Used to discover software applications that are installed on an agent machine.
Palm PDA XML Inventory	Palm PDA XML Inventory	This service scans for and reports back information on installed Palm PDA devices. Will only report back if a device is found.
RIM Reporting	RIM Reporting	This service performs a scan of a systems Win32 devices such as: BIOS, Computer System, environment, keyboard, logical disk, logical memory configuration, network adapter, operating system, pointing device, printer, processor product, serial port, service, software element, and video controller. Note: This is a very large scan and may take several minutes to complete.
Unix File Scan Audit	UNIX File Scan Audit	This service performs a scan of the user's computer for a specified file of an instance of the AUDIT.FILE classes on UNIX platforms.
Unix	Unix	This service scans for and reports on a user's hardware on UNIX computers.

Service	Connected to Audit Package (PACKAGE)	Description
Hardware Inventory	Hardware Inventory	
Unix Software Inventory	Unix Software Audit	This service performs an audit to find UNIX-based software.
WBEM MSI Based Applications	WBEM Scan for Windows Installer Applications	This service performs a WBEM scan of the user's computer for components registered in the WMI database that have been installed by Microsoft Windows Installer.
WBEM Running Services	WBEM Scan for Running Services	This service scans the user's computer for system services that are running at the time of the scan.
WBEM Scan for Hardware	WBEM Scan for System Software	This service scans for and reports on a user's hardware.
WBEM Scan with Condition Statement	WBEM Scan with Condition Statement	This service performs scans based on a conditional statement set in the CONDITION attribute.
WBEM Stopped Services	WBEM Scan for STOPPED Services	This service scans the user's computer for system services that are stopped at the time of the scan.
WBEM System Drivers	WBEM Scan for Windows System Drivers	This service scans the user's computer for Win 32 system drivers.
WBEM Windows Services	WBEM Scan for Windows Services	This service scans for and reports on Windows Services.
Windows System DLL	Audit System DLL	This service scans for system DLLs and reports on them.

WBEMCURR object is not created by default on a Linux 32-bit system when the following Audit services are installed.

- UNIX_WBEM_COMPUTERSYSTEM
- UNIX_WBEM_PROCESS
- UNIX_OPERATINGSYSTEMS

To make Agent Audit services that use WBEM services to work successfully on a Linux 32-bit system, create a softlink `libssl.so.0.0.1` and `libcrypto.so.0.0.1` with `libssl.so` and `libcrypto.so` respectively. Make sure that the CIM Server is started before running the Audit services.

Configuring a Sample Audit

All of the examples presented can be configured for individuals, departments, work-groups, and so forth.

Sample Audit Service for Linux Systems

For documentation purposes, the sample audit service Unix Software Inventory is configured. This type of audit scans for all UNIX software that is installed and managed on the agent computer. The ACTION attribute indicates that the discovery of the file will be reported and sent to the Configuration Server for storage.

Sample Audit Service for Windows Systems

For documentation purposes, the sample audit service Audit Multi Files is configured. The file type to be audited is indicated in the SCANFOR attribute within the instance. This instance directs the Inventory Manager agent to scan for any `*:\program files\*\*.exe` files on the agent computer. The ACTION attribute indicates that the discovery of the file will be reported and sent to the Configuration Server for storage.

To configure a sample Audit package for Unix as well as Windows systems:

1. If you have not already done so, start the CSDB Editor.
2. Navigate to and expand the PRIMARY.AUDIT Domain.
3. Double-click **Application (ZSERVICE)** to expand the class.
4. Scroll to and expand the POLICY Domain.
For our example, all users who are members of the Workgroup class would select this audit package from their Service Lists.
5. Expand the POLICY.WORKGROUPS class.
6. Select the **Unix Software Inventory** package from the ZSERVICE Class, drag it to the POLICY.WORKGROUPS class, and drop it on the **Default** instance. The Select Connection Attribute window opens.

Note: For Windows select the **Audit Multi Files** package from the ZSERVICE class and drag it to the POLICY.WORKGROUPS class and drop it on the **Default** instance.

The Select Connections Attribute window opens.

7. Click **Copy** to add this package.
The Confirm Connection dialog box opens.
8. Click **Yes** to confirm the connection.

Sample Audit Service for Unix Systems

The Unix Software Inventory package is added to WORKGRP Class.

The collection of inventory information occurs on the agent computer when a user connects to the Configuration Server through the Application Manager agent when scheduled or notified to connect.

Note: For Unix systems, some scans may take several minutes to complete. This is a normal behavior of the audit scanning process.

Sample Audit Service for Windows Systems

Note: For Windows the Audit Multi Files package is added to WORKGRP Class.

The collection of inventory information occurs on the Inventory Manager agent computer when a user connects to the Configuration Server as follows:

- Through an Application Self-Service Manager agent connect, when the user launches that program.
or
- Through the Application Manager agent when the user double-clicks the **Connect** icon on his desktop, or is scheduled or notified to connect.

The following figure shows the available Audit Multi Files package that an Application Self-Service Manager user would see when connecting to the Configuration Server:

Application Self-Service Manager shows Audit Multi Files

Name	Status	
Audit Multi Files	Version 1.0	Size 7.00 KB Compressed Size 2.00 KB
Available		 

When the subscriber selects and installs the Audit Multi Files package from the Application Self-Service Manager, there are really two connections. The first connection downloads the Audit service. The second connection sends the audit results back to the Configuration Server. The audit-related scans are done between the two connections.

Note: Some scans may take several minutes to complete. This is a normal behavior of the audit scanning process.

Audit Scanning Processes

Use the Agent Explorer to locate the ZSERVICE for the Audit Multi Files package in the LIB directory.

AUDIT_MULTI_FILES in the LIB Directory

The screenshot shows a Windows Explorer window with the address bar displaying the path: \\10.66.193.109\\c\$\\Program Files (x86)\\PSL\\RCA\\AgentLib\\SYSTEM\\RADIA\\audit\\ZS. The left pane shows the directory tree, and the right pane shows a list of files in the audit directory.

Directory Tree (Left Pane):

- CM
 - C:\Program Files (x86)\PSL\RCA\Age
 - Data
 - Command Line
- \\10.66.193.109\\c\$\\Program Files (x86)
 - RADIA
 - audit
 - ZSERVICE
 - AUDIT_MULTI** (selected)
 - 00000000
 - os
 - OUTBOX
 - PATCH
 - SOFTWARE

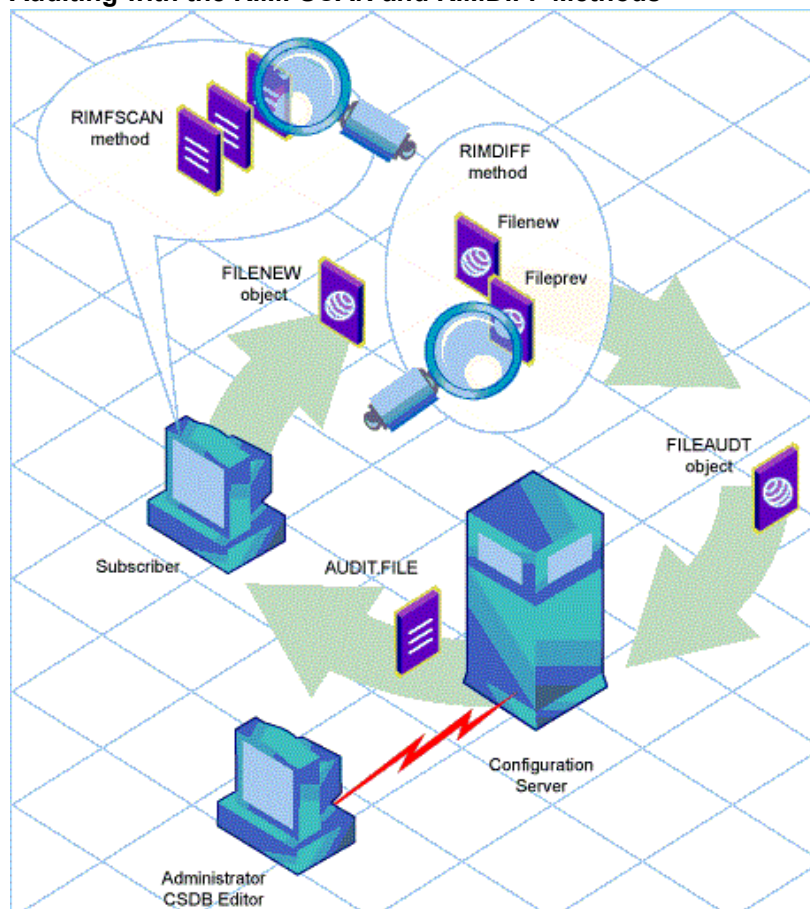
File List (Right Pane):

Name	Instance	Size	Modified
APPINFO	1	6KB	11/10/2013 11:40:24 PM
CONNECT	1	5KB	11/10/2013 11:40:24 PM
DMSYNC	1	5KB	11/10/2013 11:40:24 PM
FILEAUDT	73	259KB	11/10/2013 11:40:27 PM
FILEPREV	73	223KB	11/10/2013 11:40:29 PM
LASTEVNT	1	8KB	11/10/2013 11:40:25 PM
PATH	1	8KB	11/10/2013 11:40:24 PM
PCLSIGNO	14	18KB	11/10/2013 11:40:24 PM
STRINFO	1	4KB	11/10/2013 11:40:24 PM
TRANSFER	1	4KB	11/10/2013 11:40:24 PM
ZCONFIG	1	8KB	11/10/2013 11:40:24 PM
ZLOCAL	1	5KB	11/10/2013 11:40:24 PM
ZMASTER	1	8KB	11/10/2013 11:40:29 PM
ZTRANSRS	1	5KB	11/10/2013 11:40:24 PM

Within the ZSERVICE, note the two objects, **FILEAUDT** and **FILEPREV**. These objects are created and stored in the ZSERVICE of the **LIB** directory whenever an audit package is installed. The FILEAUDT object contains one heap for each file discovered during the auditing scan. It also contains the attributes from the AUDIT.FILE instance that controlled the scan.

The `AUDIT.FILE` class instances in an audit package control the auditing for files on the agent computer. The `RIMFSCAN` and the `RIMDIFF` methods on the agent computer perform the actual file auditing operations by specifying what files to look for.

Auditing with the RIMFSCAN and RIMDIFF Methods



- The RIMFSCAN method scans the Client Automation agent's file system based on the values in the AUDIT.FILE class instance in the audit package. It constructs an object named FILENEW. The FILENEW object contains one heap per file discovered during the current scan.
- The RIMDIFF method compares scan results from the current scan (the scan done during the current agent connect stored in the FILENEW object) with scan results from a previous scan (the scan done during a previous agent connect process stored in the FILEPREV object). It will construct the FILEAUDT object that is then sent to the Configuration Server. The RIMDIFF method then deletes the FILEPREV object and will rename the FILENEW object to FILEPREV.

For our particular example, there were 486 instances for both the FILEAUDT and the FILEPREV object located on the Client Automation agent's computer.

Audit Scan Results

Use the Admin Agent Explorer to locate the ZSERVICE instance for the Unix Software Inventory package (for the Audit Multi Files package in Windows Software Inventory package) in the LIB directory.

To locate the ZSERVICE object using the Admin Agent Explorer:

1. Start the Admin Agent Explorer (. /radobjed).
2. Navigate to the correct path of the Unix Software Inventory ZSERVICE instance. A sample location for the ZSERVICE object would be:

```
/opt/PSL/RCA/Agent/lib/SYSTEM/NVDM/SOFTWARE/ZSERVICE/UNIX_SOFTWARE_
INVENTORY
```

Within the ZSERVICE, note the object WBEMPREV. This object is created and stored in the ZSERVICE of the LIB directory whenever a WBEM package is installed. The WBEMPREV object contains one heap for each file discovered during the scan. It also contains the variables from the AUDIT.WBEM instance that controlled the scan.

The AUDIT.WBEM Class instances in an audit package control the auditing for files on the agent computer.

- The agent scans the client's computer file system based upon the values contained in the AUDIT.WBEM Class instance in the audit package. It constructs an object called WBEMCURR.
- The WBEMCURR object contains one heap per instance of each WBEM Class discovered during the current scan.
- The agent compares the scan results from the current scan (the scan done during the current agent connect stored in the WBEMCURR object) with the scan results from a previous scan (the scan done during a previous agent connect process stored in the WBEMPREV object). It will construct the WBEMAUDT object that is then sent to the Configuration Server.
- The agent then deletes the WBEMAUDT object and will rename the WBEMCURR object to WBEMPREV.

For our particular example, there were 318 instances for the WBEMPREV object located on the subscriber's computer.

Configuring Timers for Audit Collection

This chapter helps you install and implement the Inventory Manager. Choose the appropriate strategies suited for your enterprise needs.

ALWAYS The Scheduling (TIMER) Class

The Scheduling (TIMER) Class enables the administrator to set a timer on the agent computer and will cause one or more audit services to be processed whenever the timer expires. The administrator can use this method to process mandatory audit services automatically according to a predetermined schedule.

Note: As distributed by Persistent Systems, the SOFTWARE Domain also contains a Scheduling (TIMER) Class. Timers can be specified in instances of either Scheduling (TIMER) Class and can be connected to an Application (ZSERVICE) Class instance in either the SOFTWARE or AUDIT Domains interchangeably.

Housed within the AUDIT.Scheduling (TIMER) Class are three sample Timer packages:

- **Daily**
which will deploy a ZSERVICE everyday at the time specified.
- **Weekday**
which will deploy a ZSERVICE on Mondays, Wednesdays, and Fridays at a specified time.
- **Weekly**
which will deploy a ZSERVICE every seven days at a specified time.
- **Discover Installed Software Timer** executes a ZSERVICE weekly between 8:30 am and 10:30 pm. Use this particular timer in conjunction with the ZSERVICE Discovery of Applications that audits the ADD/REMOVE PROGRAM part of the OS.

These sample packages can be copied, changing the time parameters to suit your needs. For information on copying an instance, see the *Sentient Enterprise Console Online Help*. Or, you can create a new timer instance by following the procedure "[To create a new timer in the AUDIT Domain:](#)" on page 142.

Timers can be set to expire periodically (hourly, daily, weekly, monthly, or at defined intervals), on a specific date, or at a specific time. Each agent is installed with the Scheduler service. This service contains an executable timer component that executes any program on the end-user desktop when a timer expires.

Typically, the Scheduler service lies dormant in the background, and wakes up once per minute to see if a timer has expired. When a timer expires, the command line associated with the expired timer is executed. Normally, this command line invokes a connection to the Configuration Server to deploy or maintain a service.

The following table explains the Scheduling (TIMER) Class attributes:

Scheduling (TIMER) Class

Attribute	Description
ZOBJPRI	Priority [00-99] Sets the priority for deployment of the ZTIMEQ object. The ZTIMEQ object is deployed relative to the other elements being deployed during the agent connect. The elements with a priority number less than the value of ZOBJPRI are deployed before the ZTIMEQ object. A value of 90 is inherited from the base instance and should not be changed.
ZSTOP	Stop Expression Expressions evaluating to "true" in ZSTOP attributes cause resolution of the instance to be skipped. If left blank, the instance is not skipped, and resolution continues. This is useful for assigning a version to a specific set of users. Use the CSDB Editor to set this attribute.
ZSCHMODE	Timer Owner [MANAGER/DEFAULT] This attribute specifies the timer owner. It is recommended that you leave the timer owner configuration value as DEFAULT.
ZSCHDEF	Time Parameter Indicates when the timer expires. The syntax varies depending on the frequency of expiration that can be DAILY, HOURLY, INTERVAL, NUMDAY, WEEKDAY, WEEKLY.
ZSCHTYPE	Type [IMMEDIATE/DEFERRED] <i>Used only when ZSCHFREQ = PERIODIC.</i> Set ZSCHTYPE to DEFERRED to indicate that the first time an event is attempted to be launched, it will be deferred until the <i>next</i> scheduled time, no matter when the timer instance is evaluated. This was designed to handle the case of a daily 4 AM (non-peak) scheduled event that is sent to the agent computer during the day. If it was not deferred, it would launch during the day instead of "waiting" until the next morning. Example 1: Suppose you create and deploy a timer with the ZSCHDEF = DAILY (&ZSYSDATE,4:00:00) If ZSCHTYPE = IMMEDIATE and it is: Before 4:00:00, the command in the instance will be executed the same day at 4:00:00 After 4:00:00, the command in the instance will be executed immediately If ZSCHTYPE = DEFERRED and it is: Before 4:00:00, the command in the instance will be executed the <i>next</i> day at 4:00:00 After 4:00:00, the command in the instance will be executed the <i>next</i> day at 4:00:00 Example 2: Suppose you create and deploy a timer with the ZSCHDEF = WEEKDAY (FRIDAY,4:00:00) If ZSCHTYPE = IMMEDIATE and it is:

Attribute	Description
	Not Friday or Friday and before 4:00:00, the command in the instance will be executed on Friday at 4:00:00 Friday and after 4:00:00, the command in the instance will be executed immediately If ZSCHTYPE = DEFERRED and it is: Not Friday or Friday and before 4:00:00, the command in the instance will be executed a week later on Friday at 4:00:00 Friday and after 4:00:00, the command in the instance will be executed a week later on Friday at 4:00:00
ZSCHFREQ	Frequency [PERIODIC/ONCE/RANDOM] Use this attribute to specify how often the timer should expire. ONCE = expire one time PERIODIC = expire repeatedly RANDOM = expire in random intervals
ZRSCCMDL	Command Line to Execute This attribute indicates the command line that is executed on the subscriber's computer when the timer expires.
ZSVCOID	Service ID This attribute specifies the object ID of the Application instance that this Scheduling instance is connected to. This value is inherited from the base instance and should not be modified.
ALWAYS	Connect to Class connections determine the path of resolution of an agent's desired state. An _ALWAYS_ class connection is always resolved. Stores the connections to other instances.
NAME	The friendly name for this instance.
APPSVC	The Application Name.
REQUEST	The Application Request.
DOMAIN	The server's domain name.
IPADDR	The server's IP address/name.
SOCKET	The server's socket number.
MGRNAME	The server's name.
ZCREATE	Client Create Method Sentient-supplied method or fully qualified, user-defined method. This value is inherited from the base instance and should not be changed.
ZVERIFY	Client Verify Method The Scheduler VERIFY method that runs on the agent computer. This value is inherited from the base instance and should not be changed.

Attribute	Description
ZUPDATE	The Scheduler UPDATE method that runs on the agent computer. This value is inherited from the base instance and should not be changed.
ZDELETE	Client Delete Method Sentient-supplied method or fully qualified, user-defined method. This value is inherited from the base instance and should not be changed.
ZNOPING	Stop RCS Detection [Y/N/W/S/P] Controls the automatic sensing of a network connection between the agent computer, the Configuration Server. If set to S or P it will ping SAP entries according to their priorities, in addition to the Configuration Server. An expired time will continually evaluate whether communications with the Configuration Server can be established. When communications are established, the command line associated with the time is executed. After executing the command line, the Scheduler service resumes normal evaluation of whether the timer has expired again. This attribute is especially useful for mobile users. Note: In order to use this attribute, you must add it to the TIMER Class template.
NETAVAIL	Check Network Availability [Y/N/W] Checks for network availability before executing the command
OBJGUID	Instance GUID Object Global Unique Identification – <i>for future use</i> .
PINGCNT	Number of Ping Attempts This attribute sets the number of ping attempts to be made by the Configuration Server.
PINGDLAY	Time between Pings (in msec) This attribute sets the amount of time between pings in msec.
REQUEST	Application Request
RUNSYNC	Synchronous Timer Execution [Y/N] This attribute sets the value of Yes or No for the synchronous timer execution. The default value is Yes.
ZCHNNAME	Channel [DOMAIN] Name
ZPRVNAME	Provider (Manager) Name

Creating a Timer Instance

This section covers how to create and configure a timer and connect it to the service that you want to deploy. Prior to creating and configuring a timer, consider the following:

- What time of day should the timer expire?
- How often do you want the timer to expire?
- Does the timer need to expire more than once?
- What should happen when the timer expires?

To create a timer, use the CSDB Editor to create a Scheduling (TIMER) instance in the AUDIT Domain.

Note: As distributed by Persistent Systems, the SOFTWARE Domain also contains a Scheduling (TIMER) Class. Timers can be specified in instances of either Scheduling (TIMER) Class and can be connected to an Application (ZSERVICE) Class instance in either the SOFTWARE or AUDIT Domains interchangeably.

For the purposes of documentation, the timer created will be created from within the AUDIT Domain.

For more information concerning the Scheduling (TIMER) Class, see the *Sentient Enterprise Console Online Help*.

Note: The following section uses the CSDB Editor, which is available for 32-bit Windows platforms.

To create a new timer in the AUDIT Domain:

1. Go to **Start > Programs > Sentient Enterprise Administrator > Sentient Enterprise CSDB Editor**

The CSDB Editor Security Information dialog box opens.

Note: The default user ID is Admin and password is secret. This might have changed during installation. Check with your Sentient security administrator to obtain your own User ID and Password, if necessary.

2. If necessary, type a User ID and Password, and then click **OK**. The CSDB Editor window opens.
3. Double-click **PRIMARY**.
4. Expand the **AUDIT Domain**.
5. Right-click **Scheduling (TIMER)**.
A shortcut menu opens.
6. Select **New Instance**. The Create Instance dialog box opens.
7. Type a name for the new timer instance.
8. Click **OK**. The new timer instance appears in the Scheduling (TIMER) Class.

Specifying Timer Settings

Whether you copied an existing timer or you created a new Timer instance, you will need to review and/or customize your timer settings.

Note: For more information concerning the Scheduling (TIMER) Class, see the *Sentient Enterprise Console Online Help*.

Specifying ZSCHDEF

Use ZSCHDEF to indicate when the timer should expire. The syntax varies depending upon the expiration frequency. When configuring ZSCHDEF, the variable is set in the following form:

```
freq(date,time[,limit_time][count])
```

- The value of *freq* can be:
DAILY, WEEKLY, WEEKDAY, HOURLY, INTERVAL, NUMDAYS
 - If the value of *freq* is DAILY, WEEKLY, HOURLY, INTERVAL, or NUMDAYS, the date is then specified in the following form:
YYYY/MM/DD
 - If the value of *freq* is WEEKDAY, the date is then specified as the name of a day of the week in all uppercase letters. This would be one of the following:
MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY, SUNDAY
- The value for *time* or *limit_time* is optional. It is specified in the following form:
HH:MM:SS
- The value for *count* is optional. It is specified as an integer.
- The timer expiration can also be configured on the value of ZSCHFREQ. Use the following table to help you determine the appropriate syntax.

Syntax of ZSCHDEF Variables

Type	Syntax	Timer Expires
DAILY	DAILY (&ZSYSDATE,24:00:00)	Daily at midnight by the system's date.
WEEKLY	WEEKLY (&ZSYSDATE,01:00:00)	Every 7 days at 1:00 AM.
WEEKDAY	WEEKDAY (MONDAY,01:00:00)	Every Name of Weekday* starting on MONDAY at 1:00 AM. The weekday must be specified in uppercase.
HOURLY	HOURLY (&ZSYSDATE,08:41:00)	Hourly starting at 8:41 AM on the systems date.
INTERVAL	INTERVAL (&ZSYSDATE,08:41:00,,30)	Every 30 minutes starting at 8:41 AM based on system's date.

Type	Syntax	Timer Expires
NUMDAYS	NUMDAYS (20000803,08:00:00,,14)	Every 14 days starting on August 3, 2000 at 8:00 AM.

* *Name of Weekday* is the name of a specific weekday in uppercase letters, e.g. MONDAY.

Specifying ZSCHTYPE

The ZSCHTYPE controls how the timer handles the scheduled event when the client receives the initial TIMER definition for a service. There are two valid controls:

- **IMMEDIATE**
will execute the command specified in the ZRSCCMDL immediately if the date and time indicated in ZSCHDEF has passed when the ZTIMEQ object is initially created.
- **DEFERRED**
will defer the execution if the date and time defined in the ZSCHDEF has passed and will wait until the next occurrence to execute. This is the recommended setting.

If the time and date indicated in ZSCHDEF has not passed when the ZTIMEQ object is deployed, this setting has no effect.

Specifying ZSCHFREQ

Use ZSCHFREQ to specify whether the timer should expire once (ONCE) or repeatedly (PERIODIC) according to the frequency specified in ZSCHDEF.

Specifying ZRSCCMDL

Use ZRSCCMDL to execute a command on the subscriber's computer when the timer expires.

Use the following command line to run the audit service when the scheduled time occurs:

```
radskman,cat=y,uid=&(ZMASTER.ZUSERID),startdir=&(ZMASTER  
.LOCALUID),mname=&(ZMASTER.ZMGRNAME,dname=&(ZMASTER.ZDOMNAME,  
sname=&(ZSERVICE.ZOBJNAME)
```

Note: Execution causes Client Automation to launch the AUDIT service behavior, (EXECUTE.REXX) attached to the AUDIT service.

Note: The parameters indicated in the radskman command may differ depending upon customer specific implementations.

Specifying ZNOPING, PINGDLAY, and PINGCNT

The ZNOPING attribute controls automatic sensing of a network connection between the agent computer and the Configuration Server. Use this attribute when there is a possibility that the client will not be able to connect with the Configuration Server, such as when the client is a mobile user.

- If the ZNOPING attribute is not in the ZTIMEQ object, or if ZNOPING is not equal to **N**, the Scheduler service does not ping the Configuration Server.
- If ZNOPING = **N**, the Scheduler service will ping the Configuration Server.
 - If the Configuration Server is pinged successfully, the command in ZRSCCMDL is executed. The PENDING attribute in the client's ZTIMEQ object is then set to **N**. This will indicate that the Scheduler service does not need to ping the Configuration Server again.
 - Set ZNOPING to **W** if you are specifying an end limit in the ZSCHDEF attribute. The Scheduler pings the Configuration Server before executing the command. If the Configuration Server is unavailable, the ZPENDING flag is set to **W**. If the ZSCHDEF has a limit time, then when that time passes, the ZPENDING flag is set to **N**, and the Scheduler will not attempt to execute the command until its next scheduled time.
 - If the Configuration Server is not pinged successfully, the timer is not processed any further. The PENDING attribute value remains set to **Y**. The next time the Scheduler service expires, it should ping the Configuration Server again.

If ZNOPING is set to **N**, also use the PINGDLAY and PINGCNT attributes to further specify the timing and number of pings between the agent computer and the Configuration Server.

- If ZNOPING is set to **N**, PINGDLAY specifies the time in milliseconds between pings. The default is **2000**.
- If ZNOPING is set to **N**, PINGCNT specifies number of ping attempts. The default is **3** attempts.
- If ZNOPING = **S**, the Scheduler service will ping the Configuration Server, and additionally, it will ping SAP entries according to their priorities.
 - If the Configuration Server or any SAP entry is pinged successfully, the command in ZRSCCMDL is executed. The PENDING attribute in the client's ZTIMEQ object is then set to **N**. This will indicate that the Scheduler service does not need to ping the Configuration Server again.
 - Set ZNOPING to **P** if you are specifying an end limit in the ZSCHDEF attribute. The Scheduler pings the Configuration Server and SAP entries as per their priorities, before executing the command. If the Configuration Server and all SAP entries are unavailable, the ZPENDING flag is set to **W**. If the ZSCHDEF has a limit time, then when that time passes, the ZPENDING flag is set to **N**, and the Scheduler will not attempt to execute the command until its next scheduled time.

Connecting the Timer to a Service

Once you have created your timer, you must connect it to a service. Each subscriber that receives the ZSERVICE to which the timer is connected, will receive the timer information in the ZTIMEQ object the next time the agent connects to the Configuration Server.

For the Unix systems

Use the CSDB Editor to connect the ITA Audit Timer to the ITA Audit ZSERVICE created earlier in this document.

Then connect the AUDIT.ZSERVICE .ITA Audit to a user or group of users within the POLICY Domain.

For the Windows systems

Use the CSDB Editor to connect the **Log Audit Timer** to the **Log Finder ZSERVICE** created earlier in this document.

Then connect the `AUDIT.ZSERVICE.Log Finder` to a user or group of users in the POLICY Domain.

Audit Execution Configuration

By default, when an Audit service is installed on an end user's computer, it executes immediately and reports to the Configuration Server. This can be time consuming, especially if the audit service type is WBEM or File Scan (or an MSI request for Windows systems). The audit service definition may also be installed at a time when an audit scan is not desirable. For example, when an end user visits the Application Self-service Manager and mandatory applications are processed as defined in the embed tag `enterprisemanagement=auto`.

The easiest way to approach this issue is to manipulate how and when the audit actually executes. This can be accomplished by:

- Customizing the Inventory Options (RIMOPTS) attribute.
- and
- Updating the embed tags in the html file for the Application Self-service Manager.

The following describes the steps necessary to customize RIMOPTS and update the embed tag to prevent audit execution during mandatory application processing.

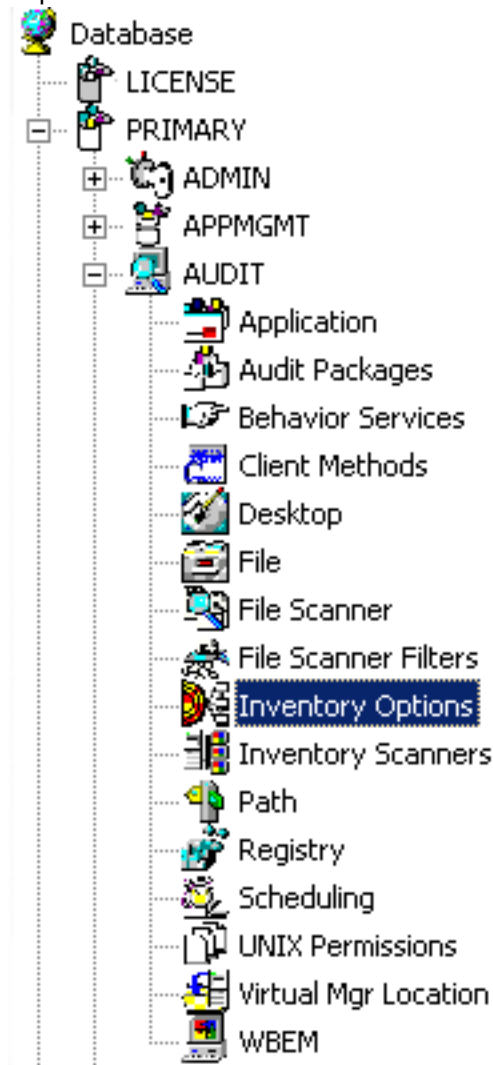
To customize the RIMOPTS instance:

1. Go to **Start > Programs > Sentient Enterprise Administrator > Sentient Enterprise Administrator CSDB Editor**
The Security Information dialog box opens.

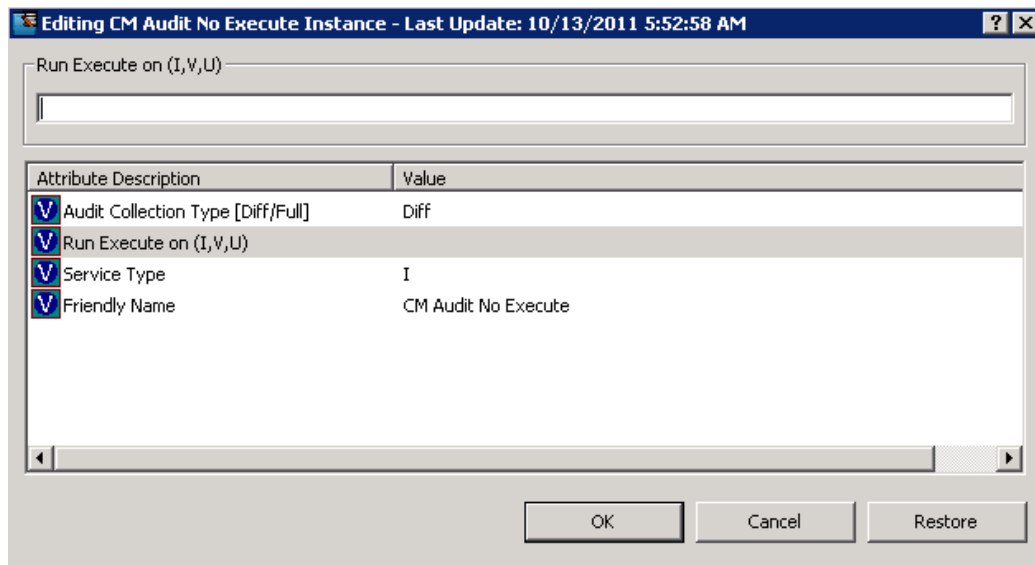
Note: The default user ID is Admin and password is secret. This might have changed during installation. Check with your Sentient security administrator to obtain your own User ID and Password, if necessary.

2. If necessary, type a User ID and Password, and then click **OK**. The System Explorer window opens.

3. Expand the **PRIMARY File** and the **AUDIT Domain**.



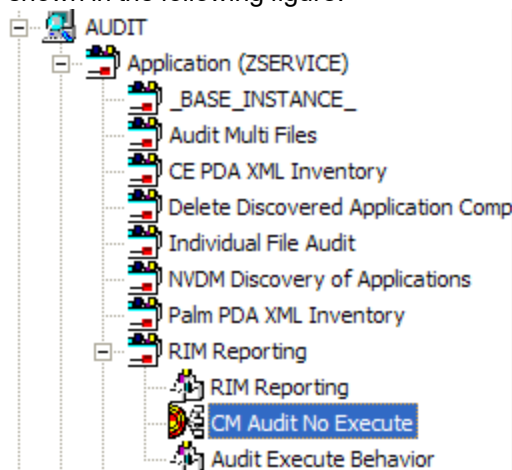
4. Create a new instance in the Inventory Options (RIMOPTS) Class called **CM_AUDIT_NO_EXECUTE**, and click **OK**.
5. Expand the Inventory Options (RIMOPTS) Class and double-click the **CM Audit No Execute** instance.
6. Double-click the **RUNEXEC** attribute in the list view to edit it. Remove any attribute information. This will ensure that the audit service will not run during the installation, verification, or update function.



Next, determine which AUDIT service you will be adding the new RIMOPTS service to. For example, select the RIM_REPORTING service.

7. Right-click the **RIM_REPORTING** Service in the AUDIT Class.
8. Select **Edit Instance**.
9. Locate the **_ALWAYS_ Contains** attribute with the value of `AUDIT.RIMOPTS.DIFF_INSTALL_UPDATE` and change it to a value of `AUDIT.RIMOPTS.CM_AUDIT_NO_EXECUTE`.
10. Next, to define the audit service as Mandatory, locate the **ZSVCMO** field and set it to M. This will cause the initial TIMER definition associated with the audit service to be created on the agent.

The **CM Audit No Execute** instance is now connected to the RIM Reporting service, as shown in the following figure:



Customizing the RIMOPTS Instance

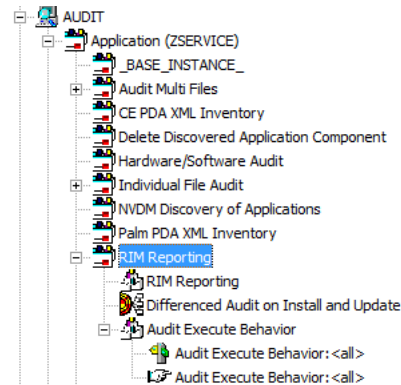
1. From the Start menu, select **Programs > Sentient Enterprise Administrator > Sentient Enterprise Administrator CSDB Editor**. The CSDB Editor Security Information dialog box

opens.

Note: The User ID, as shipped from Persistent Systems, is `Admin` and password is `secret`. This may have been changed during installation. Check with your security administrator to obtain your own User ID and password, if necessary.

2. If necessary, type a User ID and password, and then click **OK**. The CSDB Editor window opens.
3. Expand the **PRIMARY File** and the **AUDIT Domain**.
4. Create a new instance in the **Inventory Options (RIMOPTS)** class named `CM_AUDIT_NO_EXECUTE`, and click **OK**. The Create Instance dialog box opens.
Next, you will need to edit the Audit No Execute instance.
5. Expand the **Inventory Options (RIMOPTS)** class and double-click the **Audit No Execute** instance.
6. Double-click the **RUNEXEC** attribute in the list view to edit it. Remove any attribute information. This will ensure that the audit service will not run during the installation, verification, or update function.
Next, determine which AUDIT service you will be adding the new RIMOPTS service to. For example, select the `RIM_REPORTING` service.
7. Right-click **RIM_REPORTING** Service in the AUDIT class.
8. Select **Edit Instance**.
9. Locate the `_ALWAYS_ Contains` attribute with the value of `AUDIT.RIMOPTS.DIFF_INSTALL_UPDATE` and change it to a value of `AUDIT.RIMOPTS.AUDIT_NO_EXECUTE`.
10. To define the audit service as Mandatory, locate the `ZSVCMO` field and set it to `M`. This will cause the initial `TIMER` definition associated with the audit service to be created on the Client Automation agent.

The Audit No Execute instance is now connected to the RIM Reporting service.



This completes the steps necessary to customize RIMOPTS and update the embed tag to prevent audit execution during mandatory application processing.

Chapter 4

Patch Management

Patch management enables you to discover, analyze, and deploy patches in your environment. Patch management enables you to acquire patches from vendor web sites, determine the vulnerabilities on the managed devices, deploy applicable patches to managed devices, verify applied patches on managed devices to ensure compliance and security, and generate reports that provide information on which acquired and applicable patches have or have not yet been applied to a managed device.

Note: Patch management enables you to manage security and non-security patches for Microsoft. You can also manage security patches RedHat, SUSE, and HP Softpaq. Patch management does not support Update Rollups as part of non-security patches.

Patch Management Overview

The Sentient Enterprise Patch Management provides value for business continuity and security initiatives.

Key capabilities for patch management activities include:

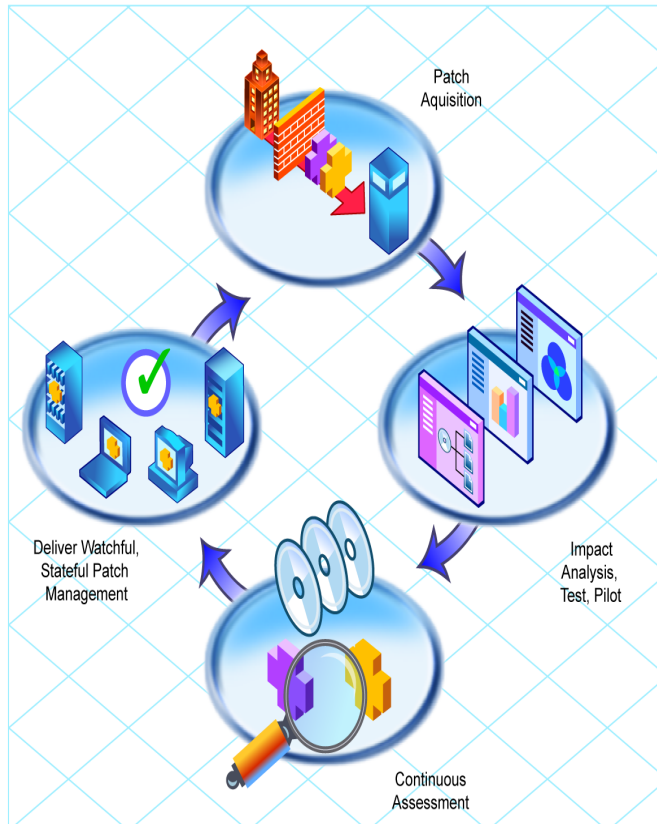
- **Acquisition:** Configurable tools to enable automatic collection of security and non-security patches from Microsoft, as well as security bulletins (advisories) for Red Hat and SuSE, based on content derived from supported vendor supplied web-based repositories.

Note: There is no support for the acquisition of security bulletins (advisories) for HP-UX and Sun Solaris (Sparc). Sentient does not support managing HP-UX or Sun Solaris Agent devices.

- **Pilot Testing:** Ability to allow IT administrators to select target pilot groups based on usage or critical need. Sentient Enterprise is the only solution with these unique pilot testing capabilities that help ensure the stability of business critical systems.
- **Compliance and Vulnerability Assessment:** Automatic and continuous discovery of devices on the network, software products that are installed on each device, the patches that are already applied by each software product, and identification of applicable software products. Through this complete discovery and assessment process, the IT administrator can understand the full scope of security vulnerability and system compliance at all times.
- **Deployment:** Policy-based deployment capabilities that interface directly with a variety of existing policy sources such as Active Directory, LDAP, or SQL databases to enable automatic, rapid, and precise targeting of patches for deployment to servers, desktops, and laptops. Sentient Enterprise patented differencing, bandwidth optimization, multicast, and checkpoint-restart capabilities and multi-tiered infrastructure ensure that patches are deployed with minimal impact on network resources, and allow patches to be managed across an enterprise of any size.

- **Compliance and Assurance:** Unique desired-state management that automatically and continuously ensures that patches remain applied in their proper state as prescribed by policy. Devices and users are monitored and checked against policy and, if found to be out of compliance, are automatically adjusted to appropriate patch levels.

Patch Management Life Cycle



Patch Acquisition Process

Patch Management is used to acquire patches and to synchronize the patch information in the CSDB on the Configuration Server with the Patch database on the SQL or Oracle Server. If you have already performed an acquisition, only instances that are different are updated.

During the acquisition, the following things occur:

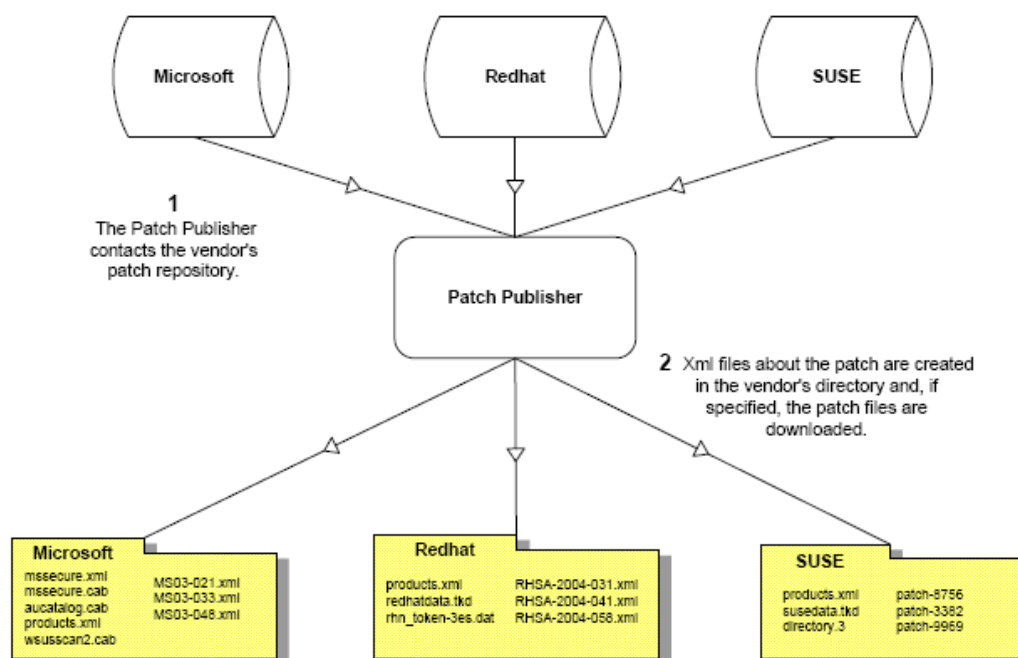
- The vendor's web site is contacted to prepare for the acquisition.
- Either the information about the Bulletins, Security Advisories, and Service Packs and the actual patch files or only the information about the patches is downloaded. The information downloaded contains, but is not limited to, detailed data about each patch, such as supersedence, reboot requirements, and probe information.
- An xml file is created for each bulletin acquired and is put in the vendor's folder in the Patch Management's directory. These files are called patch descriptor files.

- The Configuration Server Database's PATCHMGR Domain is populated with this information.
- Services are created in the PATCHMGR Domain for each of the bulletins acquired.
- The PATCHMGR Domain is synchronized with the ODBC database you created.

Overview

Patch Management provides a tool that connects to the selected vendor's web site, downloads the information regarding patches including the files, and publishes this information to the Configuration Server DB. The acquisition process fetches patches from the vendor and publishes this information to the Configuration Server DB.

Vendor's patch repository is contacted



About Patch Descriptor (XML) Files

When patches are acquired an xml, or patch descriptor file, with information about the patch is created and placed in the vendor's directory. The vendor directories are located by default in:

```
<InstallDir>\data\PatchManager\patch
```

For example, patch descriptor files for Microsoft bulletins would be in:

```
<InstallDir>\data\PatchManager\patch\Microsoft
```

while those for Red Hat are located in:

```
<InstallDir>\data\PatchManager\patch\Redhat
```

The bulletin number is the file name with an .xml extension. If the bulletin is identified by MS03-051, then the patch descriptor will be named MS03-051.xml. If you also acquired the actual files

associated with the bulletin, a folder is created with the name of the bulletin that contains the patch files.

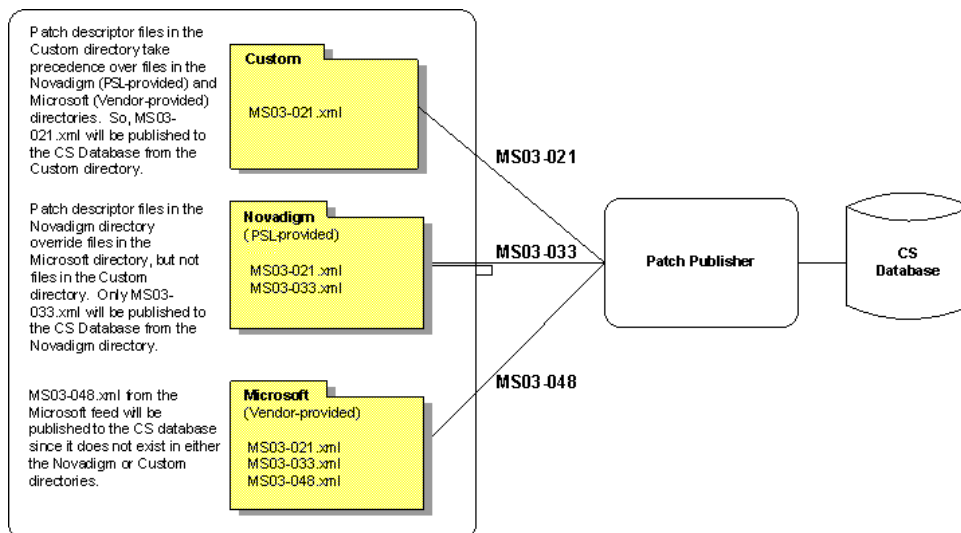
Some of the information acquired from the vendor may need to be altered before the patch can be managed. Therefore, there are two other subdirectories in the `\data\PatchManager\patch` subfolders: `novadigm` and `custom`. You are provided with some additional patch descriptor files that are located in the `novadigm` subdirectory. Patch descriptor files located in the `novadigm` subdirectory override patch descriptor files in the relevant vendor's directory. You can also create or modify your own patch descriptors and place them in the `custom` subdirectory. These custom files will override files in the `novadigm`, `microsoft`, `redhat` and `suse` directories. Use a text editor to make the changes, name the file *exactly* as it is named in the vendor's directory, and place these xml files in the Custom subdirectory.

The figure below illustrates an example of this hierarchy using Microsoft bulletins.

Note: Two *sample* descriptor files have been provided for Windows Operating System service packs, `MS03-WIN2k_4.xml` and `MS03-WINXP_1.xml`. To deploy other Microsoft Operating System service packs, you must create your own patch descriptor files and save them in the Custom subdirectory. You are responsible for deploying the service pack in a test environment before automating the deployment.

The figure below illustrates the patch descriptor override for Microsoft security bulletins. Note that the same hierarchy applies to all vendors: Microsoft, SuSE and RedHat.

Patch descriptor files



Microsoft Update Catalog Requirements: Minimum OS and Service Pack Levels

See Microsoft's website for specific information concerning the minimum Operating System and Service Pack requirements for Microsoft Update Catalog and Windows Update technologies leveraged by Patch Management. As of this writing, the supported OS Versions and languages can be viewed from the Microsoft Update Home page at this link:

<http://update.microsoft.com/microsoftupdate/v6/default.aspx>

Click **Get help and support** and access the Frequently Asked Questions.

Customers can continue to patch older operating systems without enforcing the minimum service pack levels required by Microsoft Update Catalog.

Patch Management Vendor Settings for Microsoft Data Feeds

To support the currently available Microsoft update repositories and methods, Patch Management offers the following Microsoft Data Feed Prioritization options on the Vendor Settings Page:

- **Microsoft Update Catalog Only**
- **Microsoft Update Catalog, Legacy Catalog**

Microsoft Office and Microsoft Update Catalog

Office patches deployed via the Microsoft Update Catalog will not detect if Office Applications are currently being managed by an Sentient Enterprise management application (for example, Application Manager or Application Self-service Manager), or an Administrative Control Point. In either case, if a bulletin affecting an Office application is entitled to a device, Patch Management will manage the Office patch and install it locally onto those devices that are vulnerable. For more information on patching devices with Microsoft Office, see "[Detecting and Managing Microsoft Office Security Bulletins](#) " on page 161

About Microsoft Automatic Updates

Automatic Updates is a feature of Microsoft Windows that enables users to initiate a scan of their system for needed patches. Microsoft Automatic Updates also allows for the download and installation of the patches. This Microsoft feature is accessed from **Start > Control Panel> System and Security > Windows Updates (Turn automatic updating on or off)**.

Automatic Updates currently supports the following configuration options other than Automatic:

1. Download updates but let me choose whether to install them
2. Check for updates but let me choose whether to download and install them
3. Never check for updates (not recommended)

Both Microsoft Automatic Updates and Patch Management use an underlying Windows component, Windows Update Agent (WUA), to scan a device and install updates.

Caution: To avoid a situation where WUA may be in use by another patch management product, you are strongly advised to Turn off Automatic Updates. It is recommended to prevent collisions in patch management products until such a time that Microsoft supplies a software update to Windows Update Agent.

The potential consequences of using Automatic Update options with Patch Management are discussed below.

- If you set Important Updates to **Never check for updates**, it is possible that you will not be informed of all updates available because Patch Management does not support that product,

but Automatic Updates does.

- If you set Important Updates to **Check for updates but let me choose whether to download and install them**, it is imperative that users do not initiate the Automatic Updates download process while the Patch Management Agent is scanning or installing updates. If the Automatic Updates process is initiated manually, it could result in either process failing to download and install updates on the managed device. This behavior is not specific to Patch Management. It is also exhibited when other patch management products attempt to use WUA, and WUA is already in use.

Consult the following Microsoft KB Articles for more information:

Microsoft KB Article 910748; at the time of this writing, the url is <http://support.microsoft.com/?kbid=824684>.

If you have virus scanners installed and enabled in your enterprise, please see Microsoft KB Article 922358. This documents a need to exclude the folder `%Windir%\SoftwareDistribution` from virus scans. While this Microsoft document references specific Microsoft patch management technologies, the same Windows Update Agent limitation can occur in an enterprise using Patch Management, since it leverages Windows Update Agent technologies. Please review this Microsoft KB Article:

- Microsoft KB Article 922358; at the time of this writing, the url is: <http://support.microsoft.com/kb/922358>.

Caution: WUA uses the Microsoft Windows Service called Automatic Updates Service; this windows service must be set to either Automatic or Manual on target devices. The Automatic Updates Service can be in a stopped state since WUA will start it as needed.

For more information about the configuration of Automatic Updates. See the Microsoft article *How to configure and use Automatic Updates in Windows* which is at url <http://support.microsoft.com/kb/306525> at the time of writing.

About Red Hat Patch Acquisition

To acquire security patches for Red Hat:

- Establish a Red Hat Network account using the Red Hat web site. At the time of this writing, the location is <http://redhat.com>.
- You will need a Red Hat Network account with one system entitlement for each of the Red Hat Server OS Filter options (version + release + hardware architecture combination) for which you want to acquire and manage patches. These should correspond to the OS Filter options you selected in the Patch Management Configuration.

Note: For example, to perform patch acquisitions for Red Hat Enterprise Server (ES) Version 4 on x86 systems only, you will need a Red Hat Network account with one Red Hat Network system entitlement. To perform patch acquisitions for Red Hat ES Version 4 on x86-64 systems, you will need an additional Red Hat Network system entitlement. To perform acquisitions for Red Hat Version 5 Servers, on both x86 and on x86-64 systems, you will need an additional two Red Hat Network system entitlements. To perform

acquisitions for Red Hat Version 6 Servers, on both x86 and on x86-64 systems, you will need an additional two Red Hat Network system entitlements.

Prerequisite, or dependent, Red Hat packages required to install Red Hat Security Advisories can be acquired from two places. They can either be downloaded from the Red Hat Network during acquisition or they can be found locally if copied from the Red Hat Linux installation media. During an acquisition, Patch Management will first look for the .rpm packages in the appropriate directory. For example:

- For Red Hat Enterprise Linux 4ES on x86, place the baseline operating system rpm files supplied on Red Hat installation media in `data/patch/redhat/packages/4es`.
- For Red Hat Enterprise Linux 4ES on x86-64, place the baseline operating system rpm files supplied on Red Hat installation media in `data/patch/redhat/packages/4es-x86_64`.
- When naming the `data/patch/redhat/packages/` subdirectories, see the list of **OS Filter Architecture** values with on Red Hat Feed Settings section in the Configuration chapter in the *Sentient Enterprise Console Online Help*. Use the applicable folder name based on the value following REDHAT:: as the subdirectory name.

If a patch's prerequisite software is not found locally, then the package will be downloaded from the Red Hat Network. To decrease the time needed for acquisition, it is recommended copying the dependency packages to the appropriate packages directory from your Linux installation media. The Red Hat RPM packages can be found on the installation media under the `RedHat/RPMS` directory.

- Use the `rhncerttool` to create a Red Hat Network (RHN) `systemid` file. This file will be used to pass RHN credentials during acquisition. See the procedure below for details.

Creating a Red Hat systemid file

To create a Red Hat systemid file:

1. Perform a root login to a Linux Server running the Red Hat OS for which you would like to automatically acquire security patches.
2. Execute the command `rhncerttool` on the command line when logged into the system as root.
3. When prompted by the `rhncerttool` to use an existing or new account, select existing and supply the Red Hat Network username and password you created on the Red Hat web site.
4. Enter a unique profile name for this computer such as the IP address or hostname, and exit the `rhncerttool` without applying any patches to the system where you ran `rhncerttool`. A file called `systemid` is created.
5. Copy the file `/etc/sysconfig/rhn/systemid` produced by the `rhncerttool` to the `\PatchManager\etc` directory on your Patch Management Server.
6. Rename the file from `systemid` to one of the following `redhat-*.sid` filename conventions. They vary according to the hardware architecture:

- For x86 systems, rename `systemid` to `redhat-version+release.sid`, where `version+release` represents one of the three combinations of Red Hat Version 4 followed directly by the Release (`as`, `es`, or `ws`), or, for Red Hat Version 5, `version+release` is either `5server` or `5client`, or, for Red Hat Version 6, `version+release` is either `6server` or `6client`.
For example, if the computer was running Red Hat Enterprise Server V 4, then rename the `systemid` file to `redhat-4es.sid`.
- For x86_64 systems, rename `systemid` to `redhat-version+release-x86_64.sid`. This is the same naming convention as above, except it adds the architecture type of `-x86_64` to the filename, prior to the `.sid` extension. For example, if an x86_64 computer was running Red Hat Enterprise Server V 4, then rename the `systemid` file to `redhat-4es-x86_64.sid`.

Note: Access to the Red Hat network might be disabled if the network determines that patches have been acquired too frequently. An error will show in the `patch-acquire.log` including the text Abuse of Service detected for server linux. To resolve this issue, delete the registered system from the Red Hat network web interface at <https://rhn.redhat.com>. Recreate the Red Hat credentials file (`systemid`) using the procedure above.

Now, you can run Red Hat Enterprise Server patch acquisition. Be sure that the proper Configuration Server and ODBC parameters are configured.

Creating Custom Patch Descriptor Files

The patch descriptor files that are created using the **acquire** command use the information from the vendor data feeds. These files may be missing information or contain incorrect information regarding the patch. A **probe** defines what is needed to be in compliance with the security issue that the patch fixes. You can create a custom patch descriptor files using supported XML tags. The custom descriptor file must be placed in the custom directory and be named identically to the file it will be overriding in the `microsoft`, `redhat`, `suse`, or `novadigm` directories. The following is an example of creating a custom descriptor file for a Microsoft bulletin.

To create a custom descriptor file:

1. Copy the Microsoft version of the XML file located in `<InstallDir>\data\PatchManager\patch\microsoft` directory generated during an acquisition into the `<InstallDir>\data\PatchManager\patch\custom` directory.
2. Use a text or xml editor to view the patch descriptor file. Validate the data with the releases itemized in the URL located at the top of the xml. Change Source to Custom.

```
<Bulletin PopularitySeverityID="0"
URL="http://www.microsoft.com/technet/security/bulletin"
FAQURL="http://www.microsoft.com/technet/security/bulletin"
MitigationSeverityID="0" Supported="Yes" ImpactSeverityID="0"
SchemaVersion="1.0" PreReqSeverityID="0" DateRevised="20021119"
Source="NOVADIGM" Stays NOVADIGM, like the folder name Name="MS02-065"
Title="Buffer Overrun in Microsoft Data Access Components Could
Lead to Code Execution (Q329414)" DatePosted="20021119" >
```

Note: When generating a custom xml, it is recommended including all Product releases. This allows a managed device running any available releases of the product to be discovered.

3. Make any changes required to adjust the data, and save the custom patch descriptor file. Change the `Source` tag to `Custom`. This value is reflected in the BULLETIN instance's SOURCE attribute.
Use the Patch Management in the Sentient Console to publish the custom patch descriptor file. Be sure to set the Replace option to Yes if you wish to entirely replace the bulletin previously published to the Configuration Server.
4. You may view the `patch-acquire.log` to see where the publishing process obtained the xml from:

```
20100722 10:13:09 Info: Syncing bulletin
<InstallDir>/Data/PatchManager/patch/custom/MS10-001.xml
20100722 10:13:09 Info: Publishing bulletin MS10-001, 1 of 1
20100722 10:13:09 Info: Loading XML file
<InstallDir>/Data/PatchManager/patch/custom/MS10-001.xml
20100722 10:13:10 Info: Loading BULLETIN.MS10-001 from RCS
```

Settings the Manage Installed Bulletins (mib) Option

Patch Management supports the Manage Installed Bulletins (-mib) option. By default, when Patch Management runs a discovery on target devices, it starts managing all applicable bulletins it finds installed on the target device. This means upon successive connects, Patch Management ensures previously installed bulletins are still installed.

The -mib option is available for customers who want Patch Management to skip the processing of applicable bulletins already installed on target machines, and only process the bulletins not already installed on the machines. The -mib option can take the following values:

-mib none

Manage Patch Management-installed bulletins only, and do not check the service library or binary resources for alternatively installed bulletins. This is the default behavior since there is no impact on the client agent in terms of vulnerability or re-patching, and it offers greater performance.

-mib hppm (or n)

Manage Sentient Patch Management-installed bulletins, only; do not manage bulletins installed by an external source.

-mib all (or y)

Manage all installed bulletins, whether installed by Patch Management or an external source. This option is resource intensive.

When the Patch Management is configured with the -mib option set to hppm or none, there is a substantially-reduced processing load on both the Configuration Server and the Patch Management agents.

To set the Manage Installed Bulletins (mib) Option:

Use the Agent Options page from the Sentient Console to set the Manage Installed Bulletins (-mib) option. The Agent Options page is accessed from the Configuration tab, Patch Management Group. For more information on Agent options, see *Sentient Enterprise Console Online Help*.

Patch Assessment and Analysis

This topic describes the process to discover which patches are installed on the managed devices in your environment and analyze these patches.

Product Discovery and Analysis

Before you can manage vulnerabilities, the Patch Management Agent must discover which products are on the device. Patch Management objects are cached locally on the managed device to optimize bandwidth. Objects are downloaded only if they are different. In addition, the Patch Management agent needs to detect which patches are installed for each discovered product. To do this, assign the Patch Management services for DISCOVER_PATCH and FINALIZE_PATCH to the managed devices.

Note: Running the Patch Management agent connect requires that the dname parameter be set to PATCH. This will keep separate the resolution of services for the Patch Management agent from the resolution of services for the Application Manager agent.

To perform patch discovery:

1. Connect your managed device (e.g. POLICY.USER.&(ZUSERID)) directly to the PRIMARY.PATCHMGR.ZSERVICE.DISCOVER_PATCH service. This service is prioritized to run as the first service on Patch Management agents. During a Patch Management Agent connect, this service deploys methods to the patch manager agents, and performs product discovery and vulnerability assessment.
2. Connect your managed device (e.g. POLICY.USER.&(ZUSERID)) directly to the PRIMARY.PATCHMGR.ZSERVICE.FINALIZE_PATCH. During a Patch Management Agent connect, applicable patches are downloaded and queued for management by a Patch Management Service called FINALIZE_PATCH. This service is prioritized to run as the last service on Patch Management agents. This service is required to report real time patch compliance information. Add the FINALIZE_PATCH service to the policy for all managed devices, in addition to any patch.

Caution: Failure to use this service will result in extended patch management activities and failures to report real time patch compliance information.

3. Create a radskman command line to make a regular agent connect. At a minimum, the command line should resemble the following:
`radskman ip=<ConfigurationServerIPaddress>,port=<ConfigurationServerport>,dname=patch,catexp=runmode:automatic.` For additional information on creating a radskman command line, see "[RADSKMAN Command Line Parameters](#)" on page 477.

Detecting and Managing Microsoft Office Security Bulletins

Patch Management can manage the acquisition and deployment of Microsoft Office updates. However, because Microsoft Office applications use Windows Installer technology, patching and self-healing are inherently provided. Therefore, it is important to consider how you currently install and update Microsoft Office in your environment before you enable Patch Management to deploy patches for Microsoft Office.

If you are currently distributing Microsoft Office using an external **ACP** (also known as an **Administrative Install Point, AIP**) or a Client Automation management application (Application Manager or Application Self-service Manager), it is recommended that you continue to use these solutions for updating Microsoft Office applications.

If you would like to begin using Patch Management to update Microsoft Office applications, you must discontinue using an ACP or a Client Automation management application for distributing updates to your Microsoft Office applications. You can continue to use an ACP or a Client Automation management application to deploy Microsoft Office applications; however, updates must be managed solely by Patch Management.

Caution: After Patch Management is used to distribute Microsoft Office application updates, ACP-managed and Client Automation-managed Microsoft Office applications will no longer be able to receive updates through those technologies. That is, ACP managed applications rely on a registered client-side synchronization mechanism by which updates are distributed from the ACP to the device, and Client Automation-managed applications use desired-state technology to distribute updates to Microsoft Office applications. Therefore, before enabling Patch Management for the purpose of updating Microsoft Office applications, be sure that you no longer intend to use ACP or a Client Automation application to distribute Microsoft Office updates.

This topic outlines the choices, best practices, and implementation details related to managing Microsoft Office updates with Patch Management. The topics include:

- ["Best Practices for Managing Microsoft Office Security Bulletins" below](#)
- ["About Patch Management and Microsoft Update Catalog" on page 164](#)
- ["Enabling Microsoft Office Updates in Patch Management \(Versions 3.0.2 and later\)" on page 165](#)

Best Practices for Managing Microsoft Office Security Bulletins

The following information applies to both migrated and new installations. It identifies when and how to enable Patch Management as your solution for patching Microsoft Office.

Windows Installer 3.1 Requirement

When running Patch Management, Microsoft Windows Installer version 3.1 or later is required on all target devices. Windows Installer 3.1 is needed to detect updates for Microsoft Office applications.

Options for Updating Microsoft Office Products

The method initially used for deploying Microsoft Office products determines available options for patching the agent software. Microsoft Office products use Windows Installer technology, which supports installation from compressed media typically found on a CD-ROM or an AIP. For details on Microsoft best practices, see the Microsoft article, [Distributing Office 2003 Product Updates](#).

If you deployed Microsoft Office to agents without using an Sentient Enterprise application, these Microsoft recommendations apply.

- If the Microsoft Office product was initially installed using compressed media from a CD-ROM or network file server, Microsoft recommends updating these agents by distributing the binary patch to the agent device, and allowing Windows Installer to perform local patching of the application.
- If the Microsoft Office product was installed from an AIP, Microsoft recommends that administrators obtain the appropriate administrative updates, and continue to update the centrally located AIP. This will keep agents reliably synchronized.

If you deployed Microsoft Office to agents using an Sentient Enterprise (RCA) application, these recommendations apply.

- If the Microsoft Office product was deployed using Application Manager or Application Self-service Manager, determine if the application was published in accordance with the Basic or Advanced management guidelines. If the Basic approach was used, the media was in compressed (CD-ROM) format and there are no potential software conflicts in moving to the Patch Management solution; it is recommended introducing Patch Management into this model.
- If the Microsoft Office product was deployed using Application Manager or Application Self-service Manager using Advanced management guidelines, then the media was in AIP format; it is not recommended introducing Patch Management into this model. Administrators should continue to use the Admin Publisher to streamline the AIP update process, and distribute updates using Application Self-service Manager.

Note: Before ignoring this recommendation and enabling Patch Management for Office products that were deployed using Advanced management guidelines (media in AIP format), read all of the CAUTION and WARNING statements throughout this topic to understand the potential software conflicts.

When to use Patch Management to Deploy Microsoft Office Updates

Use Patch Management to publish and deploy patches for Microsoft Office applications only when you no longer want to use another solution, such as Application Manager, Application Self-service Manager, or an external AIP. You must choose only one solution to publish and deploy patches.

Use Patch Management to deploy Microsoft Office product updates only when you are certain that the Microsoft Office product was installed from:

- Compressed media (CD-ROM).
- An AIP, but you have decided to no longer use the AIP synchronization process for updating Microsoft Office products.
- Application Manager or Application Self-service Manager, but you have decided to no longer publish or deploy patches for Microsoft Office using Application Manager or Application Self-service Manager.

Caution: Administrators who manage agent devices running Microsoft Office products currently patched through the AIP synchronization process must be careful not to interchange these patching methods (AIP synchronization process and Patch Management). Doing so may cause a break in the synchronization between the agent device and the AIP. For details about the synchronization process, see the [Microsoft article Updating Office XP Clients from a Patched Administrative Image](#).

Client Automation Management Features that are Disabled when using Patch Management

The management features of Application Manager and Application Self-service Manager that are derived from the method fields ZCREATE, ZVERIFY, and ZUPDATE will no longer be available for Microsoft Office applications once they are managed by Patch Management; these include the ability to install on first use and the ability to manage MSI Features and Properties.

If you want to continue to use these features, do not introduce Patch Management into this model. Instead, publish Microsoft Office patches using the Admin Publisher, and deploy and manage Microsoft Office patches using Application Manager or Application Self-service Manager.

Note: Microsoft Office can still be uninstalled using Application Manager or Application Self-service Manager even after it has been enabled for patching using Patch Management. This is because the ZDELETE method is never disabled.

Microsoft Update Catalog Supports Office XP, Office 2003, and Office 2007

When using the new Microsoft Update Catalog data feed, Patch Management provides support for patching Microsoft Office XP, Microsoft Office 2003, and Microsoft Office 2007, as well as their stand-alone products. For example, the stand-alone products for Microsoft Office 2007 that can be patched using Patch Management with the Microsoft Update Catalog are:

Access 2007	PowerPoint 2007
Excel 2007	Project 2007
Groove 2007	Publisher 2007
InfoPath 2007	SharePoint Designer 2007
OneNote 2007	Visio 2007
Outlook 2007	Word 2007

When using the new Microsoft Update Catalog data feed, the Patch Management does not provide support for patching Microsoft Office 2000, or earlier, applications. This restriction is a result of the Patch Management agent's reliance on the Microsoft Update Catalog to detect Microsoft vulnerabilities.

Microsoft Office Service Packs

Sentient Patch Management supports deployment and acquisition of Microsoft Office service packs. In some cases, Microsoft will determine that a particular Microsoft Office patch is dependent on a specific service pack. In those cases, it will be necessary to distribute the Microsoft Office service pack prior to installing the patch.

The Microsoft Data Feed Prioritization selection determines whether Patch Management has the ability to report and apply a pre-requisite service pack to a device.

- **For Microsoft Update Catalog Only:** Due to changes in this data feed, service pack dependencies cannot be obtained and reported by Patch Management. It is imperative that Administrators research the pre-requisite service packs for applicable bulletins and entitle them to devices.
- **For Microsoft Update Catalog, Legacy:** For devices running Windows 2000 only, Patch Management follows the behavior of the default data feed and will report and deploy dependent service packs to devices entitled to it in policy. For devices running platforms other than Windows 2000, Patch Management follows the behavior of Microsoft Update Catalog and Administrators must research and entitle devices to pre-requisite service packs.

About Patch Management and Microsoft Update Catalog

Enhancements introduced with Patch Management version 3.0.2 enable it to use new technology, including the Microsoft Update Catalog data feed and the Windows Update Agent.

For more information about Microsoft Update Catalog, see the Microsoft FAQs article at: <http://update.microsoft.com/microsoftupdate/v6/about.aspx?ln=en-us>.

Patch Management takes advantage of the Microsoft Update Catalog by using the Windows Update Agent to scan for vulnerabilities, install updates, and verify updates. The Windows Update Agent is responsible for installing updates for Windows operating systems as well as applications including Microsoft Office, thereby preventing Patch Management from determining whether Microsoft Office applications are managed by Application Manager, Application Self-service Manager, or an Administrative Control Point.

Note: Windows Installer version 3.1 is required to detect patches for Windows Installer enabled applications like Microsoft Office, which use Microsoft Update Catalog.

When using Patch Management Version 3.0.2 or above, updates for Microsoft Office applications will be detected and reported automatically, however the update will only be installed if the device is entitled.

- If you deploy patches for Microsoft Office using Patch Management with Microsoft Update Catalog enabled, then you should no longer publish or deploy patches for Microsoft Office using Application Manager or Application Self-service Manager, or an external ACP. You must choose between the Patch Management solution for patch management and your existing solution.
- If you choose Application Manager or Application Self-service Manager because you would like to leverage a feature derived from the ZCREATE, ZVERIFY, or ZUPDATE methods (such as their ability to manage MSI Features and Properties or install on first use), then you are advised to publish Microsoft Office patches via the Publisher and then deploy and manage them using Application Manager or Application Self-service Manager. You should not introduce Patch Management into this model.
- If you choose to continue to use an external ACP, then you should not introduce Patch Management into this model. Doing so will likely break the synchronization between the agent and the ACP.
- If you choose to enable Patch Management with the Microsoft Update Catalog data feed, perform the tasks in the topic below, Enabling Microsoft Office Updates in Patch Management (Versions 3.0.2 or above)

Enabling Microsoft Office Updates in Patch Management (Versions 3.0.2 and later)

Patch Management is installed by default with Microsoft Office (!Office*) patches being excluded from acquisition. As of Version 5.0, both Microsoft Office and its set of standalone products are excluded from acquisition by default.

Use the following steps to enable Microsoft Office acquisition and deployment to agents in a Patch Management environment that uses the Microsoft Update Catalog feed.

1. Ensure all devices have Windows Installer 3.1 installed.
2. When using Patch Management with Microsoft Update Catalog data feed to deploy Microsoft Office patches, you do not need to modify any Patch Management methods (as was

required in versions prior to 3.0.2). This is because the code that honors the –IR and –IACP parameters is never executed due to changes in the Microsoft patch data feed.

Caution: As previously discussed, do not enable Patch Management using Microsoft Update Catalog feed unless you will no longer be managing Microsoft Office updates with an existing solution: either Application Manager or Application Self-service Manager or an AIP. As soon as Patch Management applies a patch using Microsoft Update Catalog, a Client Automation-managed application will fail verification, and the AIP-synchronized agents will no longer be connected to the AIP.

3. If you previously used Application Manager or Application Self-service Manager to manage ZUPDATE methods in the existing SOFTWARE.ZSERVICE class instance for Microsoft Office in your database. This ensures the `radiamsi` calls do not take place, and any desired state processing by Application Manager or Application Self-service Manager does not undo the Patch Management-deployed updates. For more information on editing these methods, see the Engineering Note, *RCA Client Methods and Pre-method Variables* (Document ID KM99949) on the Persistent Systems Software Support web site.

Caution: Do not blank out the ZDELETE method; ZDELETE gives you the ability to use Application Manager or Application Self-service Manager to uninstall Office.

4. On the patch acquisition machine, remove !Office* from the product exclusion filter.
5. If you are running Patch Management V 5.0 or above from a fresh install, the default filter excludes acquisition of patches for Microsoft Office as well the individual Office products. On the patch acquisition machine, also remove any of the following Microsoft Office standalone products from the product exclusion filter, as desired:

```
,!Access*,!Excel*,!FrontPage 200[023],!FrontPage 9[78],  
!InfoPath*,!Office*,!OneNote*,!Outlook*,!PowerPoint*, !Project 200  
[023],!Project 98,!Publisher*,!Visio*, !Word*,!Works*
```

Caution: After removing the desired entries from the exclusion list, ensure the remaining entries are comma-separated.

6. Entitle the Microsoft Office bulletins to devices in policy.

About Patch Objects Used for Device Compliance Reporting

The following agent objects are created to identify what products and patches are installed on the managed device.

- DESTATUS - Device Status Object: Contains a single heap identifying the overall device status, how many bulletins are in each compliance status, and the last scan time. Compliance status values include OK, Warning, Reboot Pending, Error, and Not Applicable.
- RESTATUS - Release Status Object: Contains one heap for every release that is present on the device.

- BUSTATUS - Bulletin Status Object; Contains one heap for each bulletin and gives the bulletin status.
- PASTATUS - Patch Status Object; Contains one heap for each patch and provides the patch status.
- DEERROR - Device Error Object: Contains any errors that occurred during discovery or management of the device.

These five objects correspond to five tables in the Patch ODBC Database: NVD_DESTATUS, NVD_RESTATUS, NVD_BUSTATUS, NVD_PASTATUS and NVD_DEERROR.

During the Client Automation Agent connect process, these objects are sent to the Configuration Server, where their contents are not stored in the Configuration Server Database, but copied to a directory that is monitored by the Messaging Server. The default location of this directory varies by platform, and is given below:

- `<InstallDir>\ConfigurationServer\data\patch` (Windows).
- `/opt/PSL/RCA/ConfigurationServer/data/patch` (UNIX®).

The Patch Delivery Agent in the Messaging Server posts this information to the Patch ODBC Database for storage and reporting. Only the most recent object for each device is kept.

Note: Patch Agents prior to Version 5.0 reported this information using a single object, named ZOBJSTAT. The Patch Data Delivery Agent in the Messaging Server Version 5.10 and above will automatically post in-bound ZOBJSTAT objects to the current Patch Management ODBC Database tables, as noted above.

Managing Vulnerabilities

After you have found where vulnerabilities may exist in your enterprise, use Patch Management to manage these vulnerabilities on managed devices. For every bulletin, there is a Services (ZSERVICE) instance in the PATCHMGR Domain that is similar to the Application (ZSERVICE) instance in the SOFTWARE domain. In addition, the PATCHMGR.ZSERVICE instance supports bandwidth throttling. Visit the Persistent Systems Support web site for details.

Set policy entitlement at the ZSERVICE level. Connect the ZSERVICE instance that has the same name as a bulletin to the user instances in the POLICY domain or to the Null Instance.

Note: SuSE 10 and 11 bulletins will have Persistent Systems-assigned instance names that are derived from the original bulletin names. For details, see ["Instance Naming Convention for SuSE 10 and 11 Bulletins" on the next page](#).

To manage a vulnerability:

1. Open the CSDB Editor and navigate to the PRIMARY.POLICY.USER class.
2. Right-click a user instance and select **Show Connections**.
3. Select **PATCHMGR Domain** from the **Show connectable classes for domain** drop-down box.
4. Click **OK**.

5. Drag-and-drop the bulletin you want to manage the vulnerability for to the appropriate user instance. When the cursor turns to a paper clip, release the mouse button.
6. Click **Copy**.
7. Click **Yes** to confirm the connection.

The patch is added to the user's policy. The next time the user logs in the vulnerability will be managed, including installation if necessary.

Instance Naming Convention for SuSE 10 and 11 Bulletins

As the instance field length in the CSDB is limited to 32 characters, all SuSE 10 and 11 bulletins will be published using Persistent Systems-reformatted instance names that are shorter and easier to distinguish than the actual SuSE 10 and 11 bulletin names.

For SuSE 10

During acquisition the SuSE10 bulletin name will be converted into a new name and given to the instance names under PRIMARY.PATCHMNGR.BULLETIN and PRIMARY.PATCHMGR.ZSERVICE.

The new instance name will be formed in the following way:

For example, Sentient Patch Management will convert the SuSE 10 bulletin name entered for acquisition as follows:

On SuSE Linux Enterprise Server 10:

`SUSE-patch-MozillaFirefox-2683`

to:

`SLES10SP0-2683-MOZILLAFIREFOX`

On SuSE Linux Enterprise Desktop 10:

`SUSE-patch-MozillaFirefox-2683`

to:

`SLED10SP0-2683-MOZILLAFIREFOX`

On SuSE Linux Enterprise Server 10SP3:

`SUSE-patch-SLESP3-MozillaFirefox-2683`

to:

`SLES10SP3-2683-MOZILLAFIREFOX`

On SuSE Linux Enterprise Desktop 10SP3:

`SUSE-patch-SLESP3-MozillaFirefox-2683`

to:

```
SLED10SP3-2683-MOZILLAFIREFOX Radia
```

The reformatting drops the SUSE-PATCH prefix and reorders the remaining content to move the unique numbering scheme earlier in the format. For the above example, the CSDB instance under `PRIMARY.PATCHMGR.BULLETIN` and `PRIMARY.PATCHMGR.ZSERVICE` will be created using the name `SLES10SP0-2683-MOZILLAFIREFOX`.

Note that any comma or dot in the original SuSE bulletin name is always replaced by a hyphen (-) in the reformatted instance name created in CSDB.

For SuSE 11

During acquisition, the SuSE 11 bulletin names will be converted into a new name and the instance name under `PRIMARY.PATCHMGR.BULLETIN` and `PRIMARY.PATCHMGR.ZSERVICE` will be created in the newly formed name.

The new instance name will be formed in the following way:

For example, Sentient Patch Management will convert the SuSE 11 bulletin name entered for acquisition as:

```
UPDATEINFO-SLESSP0-MOZILLAFIREFOX-1234
```

to:

```
SLES11SP0-1234-MOZILLAFIREFOX
```

The reformatting drops the `UPDATEINFO-` prefix and reorders the remaining content to move the unique numbering scheme earlier in the format. For uniqueness among multiple SuSE versions, `SLESSP0` is expanded to include the version (11) between the product and service pack, as in `SLES11SP0`.

For the above example, the CSDB instance under `PRIMARY.PATCHMGR.BULLETIN` and `PRIMARY.PATCHMGR.ZSERVICE` will be created using the name `SLES11SP0-1234-MOZILLAFIREFOX`.

Note that any comma, dot, or underscore in the original SuSE bulletin name is always replaced by a hyphen (-) in the reformatted instance name created in CSDB.

Entitle the FINALIZE_PATCH Service

During a Patch Management Agent connect, applicable patches are downloaded and queued for management by a Patch Management Service called `FINALIZE_PATCH`. This service is prioritized to run as the last service on Patch Management agents. This service is required to report real time patch compliance information.

Add the `PRIMARY.PATCHMGR.ZSERVICE.FINALIZE_PATCH` service to the policy for all managed devices, in addition to any patch.

Caution: Failure to use this service results in extended patch management activities and times, as well as failures in the reporting of real time patch compliance information.

Deploying Automatic and Interactive Patches

Some patches require user intervention for deployment as designed by the patch's vendor. Patch Management defines a patch as **automatic** if it does not require user interaction for deployment. A patch is defined as **interactive** if it requires user interaction for deployment. Patch Management can detect vulnerabilities for both automatic and interactive patches. Patch Management supports deployment of both interactive and automatic patches. However, those which the vendor has created as interactive will either require user intervention to be installed or will fail to be installed.

Only bulletins that Hewlett-Packard has provided data correction for in an xml file or that a customer has customized may be marked as interactive. This information can be found in the Deployment attribute in the Bulletin and Patch nodes of a Hewlett-Packard provided xml file. Valid values are AUTOMATIC and INTERACTIVE. By default, the vendor does not supply this information. Therefore, customers are required to test the deployment of a patch to verify if it is interactive before entitling the bulletin in their environment.

When the bulletin is published to the Configuration Server Database, the RUNMODE attribute of the ZSERVICE class of the PATCHMGR Domain defines the type of patch. Use the catexp parameter of the radskman command line to limit your installation to bulletins marked as automatic only. The format would be catexp=runmode:automatic. If the catexp parameter does not exist, all bulletins will be processed. For a typical Patch Management Agent connect, you may want to use the following radskman command line:

```
radskman
```

```
ip=<RCSIP>,port=<RCSPORT>,dname=patch,catexp=runmode:automatic
```

For more information on radskman, see ["RADSKMAN Command Line Parameters" on page 477](#).

Customizing Reporting Options

In some cases, you may not want to mark a vulnerability as an error (shown as an X), or you may not want to mark a warning (shown as an exclamation point [!]) with a status of OK (check mark). Defaults are supplied in the OPTIONS class. You may want to view instances of the OPTIONS class as examples.

Note: When patches are acquired from Microsoft Update, the Source column in the report will show "Microsoft Update" instead of "Microsoft."

If you need to modify this behavior, create a custom .xml file using the following three new descriptor attributes.

- **DesiredState**
This attribute maps to the DSTATE attribute in the OPTIONS, FILECHG, and REGCHG classes. Use this attribute to set what the return code should be based on the criteria stated in the USE variable.
- **Report Threshold**
This xml attribute maps to the REPORT attribute in the OPTIONS, FILECHG, and REGCHG classes. The properties of the file or registry key will be sent to the Patch Management based on this value. If the return code is greater than or equal to the value of the REPORT attribute, the file and registry information will be sent to the Patch Management and will be available in

Patch Management reports. For example, set REPORT to 1 to send the properties if the return code is either 4 (Warning) or 8 (Error).

Note: Setting REPORT to 0 will send the information for all files that show an OK status. This may overburden the Patch Management Server.

- **Use**

This xml attribute maps to the USE attribute in the OPTIONS, FILECHG, and REGCHG classes. USE specifies what the criteria are that you are judging against. The possible criteria for the files (FILECHG) are GMTDATE, SIZE, VERSION, CHECKSUM, and CRC32. For registry the option is VALUE.

Caution: Be aware that if you customize how a file or registry change is reported, then vulnerabilities may still exist, but will not be reflected in your reports. Prior to changing the reporting status of a detected vulnerability, be sure you have taken measures to eliminate the particular exposure or vulnerability in your environment. Keep track of any customizations that you create.

Values for these attributes in the FILECHG and REGCHG instances will override the value in a connection OPTIONS instance. If these variables are blank in the FILECHG and REGCHG instances, then the value from the connected OPTIONS class will be used. If the patch descriptor xml file does not contain these attributes, then the values from the connected OPTIONS instance will be used.

To customize reporting options:

For the purpose of this exercise, assume that all changes are to the OPTIONS Class. Connect instances of the OPTIONS Class to the file or registry component that you want to customize reporting for.

1. In the USE attribute in the appropriate class (or in the patch descriptor file), specify what properties of the file or registry key you want to evaluate. For example, if you were only interested in the date of a file, set USE to GMTDATE.
2. Set DesiredState (DSTATE) by equating a state with a return code. Separate multiple conditions with commas. Use the appropriate state from the list below:
 - Use state E (exists) if your only criterion for status is the existence of the file or registry key.
 - Use state !E (does not exist) if your only criterion for status is whether the file or registry key does not exist.
 - Use state EQ (equal) if the file or registry key meets the exact criteria.
 - Use state !EQ (not equal) if the file or registry key does not meet at least one of the criteria.
 - Use state LT (less than) if the file or registry key is less than at least one of the criteria.
 - Use state GT (greater than) if the file or registry key is greater than at least one of the criteria.

Use the appropriate return code from the list below:

- Use 0 to represent a status of OK.
- Use 4 to represent a warning status.
- Use 8 to represent an error status.

Follow these rules for valid DSTATE values:

- At least one of the conditions should have a return code of 0 (OK), but you could have more than one condition return a non-zero value (4, 8).
- Testing for Equality (EQ) implies that the component should exist and need not be expressed in the DSTATE variable.

The samples below show an example of a customized option for a file option. The criteria specified in the Use tag are VERSION, GMTDATE, and SIZE. The DesiredState tag describes to:

- Return a status of OK if the file does not exist (!E=0).
- Return a Warning status if the VERSION, GMTDATE or SIZE of the file are greater than the patched file (GT=4).
- Return an Error status if the VERSION, GMTDATE or SIZE of the file is less than the patched file (LT=8).

```
<FileChg Name="snmpsfx.dll" CRC32=""  
Gmtime=""Path="%windir%\system32" Size=""  
Checksum="14922"Gmtime="19990212"  
Version="4.0.1381.164"DesiredState="!E=0,GT=4,LT=8"  
ReportThreshold="1"Use="VERSION,GMTDATE,SIZE" />
```

Note: The values in the XML file are entirely surrounded by quotes.

3. Set a REPORT threshold. The properties of the file or registry key will be sent to the Patch Management based on this value. If the return code is greater than or equal to the value of the REPORT attribute, the file and registry information will be sent to the Patch Management and will be available in Patch Management reports. For example, set REPORT to 1 to send the properties if the return code is either 4 (Warning) or 8 (Error).

The changes will take effect the next time you publish the patch descriptor file to the Configuration Server Database.

Disabling Vulnerability Detection and Deployment

You can disable the detection or deployment of a bulletin or patch. To do this, use the Admin CSDB Editor to set the ENABLED attribute to N in the Bulletin or Patch instance in the PATCHMGR Domain.

If you want to disable all patches for a particular bulletin, set the ENABLED attribute to N in the bulletin's instance. If you want to disable only a specific patch file's detection and deployment, set the ENABLED attribute in the patch file's instance.

Controlling Patch Deployment (PATCHARG)

For each patch file, Patch Management populates the parameters for installing and, where possible, for removing the patch. These parameters can be found in the Patch Command Line (OCREATE) and the Uninstall Command Line (ODELETE) attributes in the PATCHARGS class in the PATCHMGR Domain.

Note: When patches are acquired from Microsoft Update, the Source column in the report will show “Microsoft Update” instead of “Microsoft.”

You can change the command-line parameters for installing and uninstalling the patch file. To do this, use the PATCHARG class to create an instance and connect it to the appropriate patch file.

To create alternate command line parameters using PATCHARG:

1. Use the CSDB Editor to navigate to the PATCHARG Class in the PATCHMGR Domain.
2. Right-click **PATCHARG** and create a new instance. A new instance called WSPARGS has been created in our example.
3. Type the new parameters that you want to use. There are two attributes in the PATCHARG Class: OCREATE to install the patch, and ODELETE to remove the patch.
4. Type the path to the PATCHARG Instance in place of the PATCHARG Attribute for the patch file in the BULLETIN Class.

The parameters you created will be used for this patch file.

Removing a Patch

By default, if you disconnect a user from a Microsoft vulnerability service (ZSERVICE) instance, the patch that was installed is not removed. This behavior is controlled in the ZDELETE attribute of the MANAGE instance in the Client Method (CMETHOD) class, and is disabled by default.

Both Red Hat Security Advisory and SuSE Security Advisory removal is disabled deliberately in Patch Management. When a Linux vendor supplied patch is applied to a target system, the affected Linux software is updated to the current rpm package version and release that addresses the specific security vulnerability. Application of a Linux vendor supplied advisory (patch) does not maintain a backup of the original package, making automated rollback to a prior version impossible. An attempt to remove a Linux rpm package from a device would result in the removal of the patch as well as the rpm software package to which the patch applies. If a new vulnerability is found, Linux Security patch vendors release a new patch. This is the nature of Red Hat and SuSE Security Advisories as provided by these patch vendors.

For Microsoft patches, if you want the patch files removed when you remove a user from vulnerability management, edit the ZDELETE attribute.

Caution: Modifying the PATCHMGR.CMETHOD.MANAGE.ZDELETE method will remove *all* patches for *all* users if the user is no longer assigned the vulnerability.

See "Removing a Patch" above for additional information.

To remove a patch when a user is no longer assigned the service:

1. Use the CSDB Editor to navigate to the MANAGE Instance of the Client Method (CMETHOD) Class in the PATCHMGR Domain.
2. Double-click the ZDELETE Attribute in the tree view, and in the text box, type:
`hide nvdkit &(ZMASTER.ZSYSDRV) &(ZMASTER.ZSYSDIR) patchagt.tkd
manage.`
3. Click **OK** to change the instance. The Instance Edit Confirmation opens.
4. Click **Yes** to confirm the changes. The Patch Management Agent must make a connect in order for the managed device to receive the necessary configuration change to allow the removal of patches.

The next time you disconnect a user from a ZSERVICE Instance in the PATCHMGR Domain, the patch files will be removed.

Supported XML Tags for Patch

This chapter describes the XML tags supported in various Patch Manager classes. You can use the tags that are supported to create custom patch descriptor files.

Descriptor Files

The patch descriptor files from Persistent Systems contain information about Products, Releases, Patches, and Patch Manifests. These are shown in tables following the figure below.

If you are creating custom patch descriptor files, use the tags that are supported. The node hierarchy of a patch descriptor file is shown in the following figure:

Sample patch descriptor file

```
- <Bulletin PopularitySeverityID="0" Type="Security"
  URL="http://www.microsoft.com/technet/security/bulletin"
  FAQURL="http://www.microsoft.com/technet/security/bulletin" MitigationSeverityID="0"
  Vendor="MICROSOFT" Supported="Yes" ImpactSeverityID="0" SchemaVersion="1.0" PreReqSeverityID="0"
  DateRevised="20030120" Source="MICROSOFT" Name="MS03-001" Title="Unchecked Buffer in Locator
  Service Could Lead to Code Execution (810833)" DatePosted="20030120" Platform="winnt">
- <Products>
- <Product Name="Windows 2000 Advanced Server" FixedInRelease="Windows 2000 Service Pack 4">
- <Releases>
- <Release Name="Windows 2000 Service Pack 2">
+ <Patch VerifyCmdline=""
  PatchURL="http://download.windowsupdate.com/msdownload/update/v3-
  19990518/cabpool/Q810833_W2K_SP4_7BCAD659FA326D4979A3CE9034300EA83A30F5EC.EXE"
  Architecture="" Reboot="Y" InstallCmdline="-q /Z" Language="en"
  MSSUSName="com_microsoft.810833_W2K_SP4_5936" SupercededByBulletin=""
  SupercededByMSPatch="" OSVersion=""
  MSSecureName="Q810833_W2K_SP4_X86_EN.exe" ObjectType="winnt.patch"
  QNumber="810833" ProbeCmdline="" Superceded="N" OSType="" OSSuite=""
  Platform="winnt" UninstallCmdline="">
```

Bulletin Node

Node name: Bulletin

Parent node: None

Children: Products

XML Tags in the BULLETIN class

XML Tag	Sentient Attribute	Description
PopularitySeverityID	POPULAR	Popularity ID
URL	URL	Bulletin URL
FAQURL	FAQURL	Frequently Asked Questions (FAQ) URL
Supported	SUPPORT	Supported [Y/N]
ImpactSeverityID	IMPACT	ImpactID Source: Red Hat Network, Novell (SuSE) data feeds
MitigateSeverityID	MITIGATE	Mitigate ID
PreReqSeverityID	PREREQ	Prereq ID
DateRevised	REVISED	Bulletin Revised On Date the bulletin was revised in YYYYMMDD format. Source: Red Hat Network, Novell (SuSE) data feeds
Source	SOURCE	Source [MICROSOFT] NOVADIGM CUSTOM REDHAT SUSE] Directory from which the patch descriptor file was published.
Vendor	VENDOR	MICROSOFT/REDHAT/SUSE
Type	TYPE	Type of Bulletin Security/ServicePack/Other
Platform	PLATFORM	winnt//redhat/suse
Name	NAME	External ID Source: Red Hat Network, Novell (SuSE) data feeds
Title	TITLE	Title Bulletin title. Source: Red Hat Network, Novell (SuSE) data feeds
DatePosted	POSTED	Bulletin Posted On Date the bulletin was posted in YYYYMMDD format. Source: Red Hat Network, Novell (SuSE) data feeds
Schema Version		The patch schema version currently 1.0
	MTIME	Time the instance was modified in the CSDB.
	CTIME	Time the instance was created in the CSDB.
	ID	Internal instance ID.
PSLPosted	PSLPOSTED	Date the bulletin was initially posted by Persistent Systems.

PSLRevised	PSLREVISD	Date the bulletin was revised by Persistent Systems.
Deployment	RUNMODE	Specifies whether the patch can be installed automatically (AUTOMATIC) or needs user interaction (INTERACTIVE).

Products Node

Node name: Products

Parent node: Bulletin

Children: Product

Attributes: None

Product Node

Node name: Product

Parent node: Products

Children: Releases

XML Tags in the PRODUCT class

XML Tag	Sentient Attribute	Description
Name	NAME	Source: Red Hat Network, Novell (SuSE) data feeds
Name	NAME	Source: Red Hat Network, Novell (SuSE) data feeds

Releases Node

Node name: Releases

Parent node: Product

Children: Release

Attributes: None

Release Node

Node name: Release

Parent node: Releases

Children: Patch

XML Tags in the RELEASE class

XML Tag	Sentient Attribute	Description

Name	NAME	Source: Red Hat Network, Novell (SuSE) data feeds

Patch Node

Node name: Patch

Parent node: Release

Children: Package

XML Tags in the PATCH class

XML Tag	Sentient Attribute	Description
PatchURL	PATCHURL	A URL that points to an .EXE or .MSI file. Source: SUS, Red Hat Network, Novell (SuSE) data feeds
Reboot	REBOOT	Specified if the device should be rebooted, after the patch is installed. Source: SUS, Red Hat Network, Novell (SuSE) data feeds
Architecture	ARCH	x86 i64 Source: SUS, Red Hat Network, Novell (SuSE) data feeds
Language	LANG	en,fr,de Source: SUS
MSSUSName	SUSNAME	The SUS name for the patch
SupercededByBulletin	SUPERBU	The bulletin name that supersedes this patch. Source: Red Hat Network, Novell (SuSE) data feeds
Superceded	SUPERCED	Specifies if the patch has been superseded. Valid values are Y or N. Source: Red Hat Network, Novell (SuSE) data feeds
OSVersion	OSVER	Operating System Version
OSType	OSTYPE	The operating system type, such as server or workstation.
OSSuite	OSSUITE	The operating system suite, e.g., datacenter, blade.
Platform	PLATFORM	The platform type: winnt, redhat, suse

XML Tag	Sentient Attribute	Description
InstallCmdline	OCREATE	This is the arguments that are passed to the create procedure. Source: SUS, Red Hat Network, Novell (SuSE) data feeds
VerifyCmdline	OVERIFY	The Verify Arguments.
UninstallCmdline	ODELETE	The Uninstall Arguments.
ObjectType	OTYPE	Format: namespace=script filename Default: winnt.patch This specifies the type of the object and the name of the script file that would have the following procedures defined verify create delete assert The procedures should have the namespace as part of the name, e.g., winnt.patch::create. If the script filename is not specified then the filename is {namespace}.tcl. Source: Novadigm
ProbeCmdline	OVERIFY	The probe command line. Source: Novadigm
	ID	The unique ID created in the RCA-CSDB for this patch.
	PATCHSIG	The name of the Patch Signature instance. Source: Novadigm
	LOCATION	The name of the LOCATION instance that contains the patch data.
	BULLETIN	The bulletin name set during publishing. Source: Red Hat Network, Novell (SuSE) data feeds
	DATA	Does the RCS have the patch data [Y/N] filled in during publishing. If the RCS has the data the value would be Y else it would be N.

XML Tag	Sentient Attribute	Description
	DSTATE	Desired state for a patch, this is usually classed in from an instance. Source: Novadigm
	REPORT	Report threshold, similar to DSTATE is classed in from an instance. Source: Novadigm
	USE	The variables used in checking the desired state. Source: Novadigm
Deployment	RUNMODE	Specifies whether the patch can be installed automatically (AUTOMATIC) or needs user interaction (INTERACTIVE).

Patch Signature Node

Node name: PatchSignature

Parent node: Patch

Children: FileChg, RegChg

Attributes: None

FileChg Node

Node name: FileChg

Parent node: PatchSignature

Children: None

XML Tags in the FILECHG class

XML Tag	Sentient Attribute	Description
Name	NAME	File name.
Path	PATH	The directory name, this can contain environment variables, e.g., %windir%, and is used by the appropriate scripts for Windows and Linux.
CRC32	CRC32	The CRC of the data.
Gmttime	GMTTIME	The GMTDATE expressed as YYYYMMDD.
Gmtdate	GMTDATE	The GMTTIME expressed as HH:MM:SS.

XML Tag	Sentient Attribute	Description
Size	SIZE	The size of the file.
Checksum	CHECKSUM	The checksum of the file.
Version	VERSION	The version of the file.
	DSTATE	The desired state of the FILECHG instance, this is usually classed in from another instance in the CSDB. Source: Novadigm
	REPORT	The report threshold. If on evaluation of this file change instance the RC is greater than the threshold then a ZOBJSTAT is created for that instance. Source: Novadigm
	USE	The variables to use during comparison, e.g., Version,Checksum,Gmtdate. Source: Novadigm

RegChg Node

Node name: RegChg

Parent node: PatchSignature

Children: None

XML Tags in the REGCHG class

XML Tag	Sentient Attribute	Description
Name	NAME	Value Name.
Path	PATH	The fully qualified Registry Key Name.
Value	VALUE	The Data value stored in the registry.
Type	TYPE	Registry data type should be one of the following: sz = Simple Registry String multi_sz = Registry Multi String expand_sz = Registry string with environment variables dword = Registry dword binary = Binary data
	DSTATE	Desired state of FILECHG instance, this is usually classed in from another instance in the RCS database. Source: Novadigm
	REPORT	Report threshold. If on evaluation of this file change instance, the RC is greater than the threshold then a ZOBJSTAT is created for that instance. Source: Novadigm
Type	TYPE	Registry data type should be one of the following: sz = Simple Registry String multi_sz = Registry Multi String expand_sz = Registry string with environment variables dword = Registry dword binary = Binary

XML Tag	Sentient Attribute	Description
		data
	DSTATE	Desired state of FILECHG instance, this is usually classed in from another instance in the RCS database. Source: Novadigm

HPFileset Node

Node name: HPFileset

Parent node: PatchSignature

Children: None

XML Tags in the HPFSET class

XML Tag	Sentient Attribute	Description
Name	NAME	Fileset Name
Version	VERSION	Fileset Version

Patch.cfg Parameters

This section describes all of the possible parameters in the Patch Management Server configuration file, `patch.cfg`. Wherever possible these parameters should be edited using the Sentient Console. This list is provided as supporting information.

Patch Management Server Configuration Parameters

It is recommended that you configure the Patch Management parameters in the Sentient Console. If you cannot use the console, you can make changes directly in the `patch.cfg` file. The default location is `<InstallDir>\PatchManager\etc`. The parameters are listed in this appendix.

Caution: If you are migrating from a previous version of Patch Management, your old values in `patch.cfg` will be retained. Be aware that you will not get the new available parameters in your old `patch.cfg`, nor will you get the new default values for old parameters.

- **admin_date_fmt:** Specify the date and time format for the Sentient Console. The default is `{%Y- %m-%d %H:%M:%S}` where %Y is the year with century, %m is the month number, %d is the day of the month, %H is the hour in 24-hour format, %M is the minute, and %S is the seconds.

- **data_dir**: Specify the directory on the local computer (Patch Management Server) where you want the patches downloaded to before they are sent to your Configuration Server. Use this parameter to set where to store your patch descriptor files and patch data files in an alternate directory. If you choose to perform an acquisition using a directory that is pre-populated with data from a previous acquisition, specify a different directory in this parameter. The default is `<InstallDir>\data\PatchManager\patch`.
- **delete_data**: Specify the value as 1 to delete patch data from the `<InstallDir>Data\Patchmanager\patch\microsoft` directory, whenever any patch is retired.
- **db_type**: Specify the database type. The two possible values are `mssql` for Microsoft SQL Server (the default) and `oracle` for Oracle. If you are using Oracle, change this value to `oracle` before doing a patch acquisition or a database synchronization. This parameter is required to synchronize with an Oracle database
- **dsn**: Specify the **Data Source Name (DSN)** the Patch SQL database. This parameter is required.
- **dsn_user**: Specify the SQL user for the DSN for the Patch SQL database.
- **dsn_pass**: Specify the password for the SQL user for the DSN for the Patch SQL database.
- **ftp_proxy_pass**: If you use a proxy server for FTP traffic, specify your password.
- **ftp_proxy_url**: If you use a proxy server for FTP traffic, specify its URL in the format `ftp://ip:port`. At the time of this writing, Patch Management supports basic authentication only.
- **ftp_proxy_user**: If you use a proxy server for FTP traffic, specify your user ID.
- **history**: Specify how many days to keep the Patch Auth Store (PASTORE) instances. This class contains one instance for each patch acquisition session. It is recommended specifying this in the `patch.cfg` file and not on the command line so that the instance is saved in the `patch.cfg` file even if the Sentient Core server or your system is restarted. If history has a smaller value than `purge_errors`, then purge errors will be set to the value for history. The default of 0 means never delete any history of Patch Acquisition.
- **http_proxy_pass**: If you use a proxy server for HTTP traffic, specify your password.
- **http_proxy_url**: If you use a proxy server for HTTP traffic, specify its URL in the format `http://ip:port`. At the time of this writing, Patch Management supports basic authentication only.
- **http_proxy_user**: If you use a proxy server for HTTP traffic, specify your user ID.
- **http_timeout**: Set the total amount of time to wait for the file to be completely downloaded. If the acquisition session is unable to download the file in this time, then the acquisition will abort the current HTTP location, and will continue the acquisition with the next HTTP location. Increase the `http_timeout` if you need to allow additional time for a bulletin to download. This parameter is expressed in seconds in the `setup.tsp` page. Specify `http_timeout` in either the `patch.cfg` file or on the command line in milliseconds. This is reflected in `patch.cfg` as 3600000. If you specify `http_timeout` on the command line, it will be for this acquisition session only.
- **lang**: Patch Management supports non-double byte languages. Specify the abbreviation of the languages for which you want to acquire patches. Precede any products you want

excluded with an exclamation point (!). The default is en (English). If you wanted to include French and English, specify, `- lang fr, en`.

- **microsoft_sus_url**: Specify the URL for the Microsoft SUS feed. The default is <http://www.msus.windowsupdate.com/msus/v1/aucatalog1.cab>.
- **nvdn_url**: Specify the URL to connect to the Patch Update web site provided by Persistent Systems. This is the same as the `nvdn_url` parameter in `patch.cfg`. The default is [http://managementsoftware.hp.com/Radia/patch management/data](http://managementsoftware.hp.com/Radia/patch%20management/data).
- **purge_errors**: Specify how many days to keep the Publisher Error (PUBERROR) instances. This class contains one instance for each patch acquisition error. It is recommended specifying this in the `patch.cfg` file, not on the command line so that the instance is saved in the `patch.cfg` file even if the Sentient Core server or your system is restarted. If history has a smaller value than `purge_errors`, `purge_errors` will be set to the value for history. The default is 7.
- **rcs_pass**: If authentication has been enabled on your Configuration Server, specify the password for the `rcs_user`.
- **rcs_url**: Specify the location of your Configuration Server in URL format. This parameter is required. Use the format `radia://ipaddress:port`, where:
 - `radia` indicates the session type to be opened to the Configuration Server
 - `ipaddress` is the hostname or IP address of the computer hosting the Configuration Server
 - `port` is the port number of the Configuration Server.
- **rcs_user**: If authentication has been enabled on your Configuration Server, specify the `rcs_user`.
- **reporting_url**: Specifies the URL of your Reporting Server. The default is <http://localhost/reportingserver>.
- **retire**: Specify the bulletins to retire separated by commas. Use the `-retire` parameter to:
 - Delete specified bulletins if they exist in the Configuration Server database during the current publishing session.
 - Not publish the bulletins specified in the retire parameter to the Configuration Server database during the current publishing session. The use of the retire option supersedes the bulletins option.

This parameter works on the bulletin level, not at the product or release level.

To only retire a specific bulletin, but not acquire any new ones, use `-bulletin NONE` in addition to the retire parameter

Note the following:

- The only time the retire option should be used on the command line is to delete specific bulletins from the Configuration Server Database. However, it does not keep a cumulative list of retired bulletins if you specify the option on the command line.
- It is recommended that you set a retired bulletin list in the `patch.cfg` so a cumulative list is maintained. As needed, add to the list in `patch.cfg` instead of recreating the list of retired bulletins on the command line each time you want to retire a new one.

- If you have enabled patch-removal capabilities, and retire bulletins that are currently under management in your enterprise, the retired patches may be removed from your Patch Management Agent devices.

Example: `-retire MS00-001,MS00-029`

- **rh_depends:** Specify yes if you want to publish additional Red Hat packages that downloaded security advisories may depend on. You can override this setting for a specific acquisition in Acquisition Settings. Prerequisite, or dependent, Red Hat packages required to install Red Hat Security Advisories can be acquired from two places. They can either be downloaded from the Red Hat Network during acquisition or they can be found locally if copied from the Red Hat Linux installation media. During an acquisition, Patch Management will first look for the **.rpm** packages in the appropriate directory. For example:
 - For Red Hat Enterprise Linux 4ES on x86 devices, place the baseline operating system rpm files supplied on Red Hat installation media in `data/patch/redhat/packages/4es`.
 - For Red Hat Enterprise Linux 4ES on x86-64 devices, place the baseline operating system rpm files supplied on Red Hat installation media in `data/patch/redhat/packages/4es-x86_64`.
 - When naming the `data/patch/redhat/packages/` subdirectories, see the list of **OS Filter Architecture** values listed in Red Hat Feed Settings section in the Configuration chapter in the *Sentient Enterprise Console Online Help*. Use the applicable value following `REDHAT::` as the subdirectory name.

If a patch's prerequisite software is not found locally, then the package will be downloaded from the Red Hat Network. To decrease the time needed for acquisition, it is recommended copying the dependency packages to the appropriate packages directory from your Linux installation media. The Red Hat RPM packages can be found on the installation media under the `RedHat/RPMS` directory. The default is No.

- **rhn_url:** Specify the URL for the Red Hat Security Network. The default is <http://xmlrpc.rhn.redhat.com/XMLRPC>.
- **suse_pass:** :Specify the password for the Novell web site that is hosting SuSE 9 patches.
- **suse_urls:** :Specify the URLs for the Novell web site that is hosting SuSE patches. The defaults are: 9:
 {<https://you.novell.com/update/i386/update/SUSE-CORE/9/>}
 {<https://you.novell.com/update/i386/update/SUSE-SLES/9/>}
 9-x86_64:
 {https://you.novell.com/update/x86_64/update/SUSE-CORE/9/}
 {https://you.novell.com/update/x86_64/update/SUSE-SLES/9/}
- **suse_user:** Specify the user for the Novell web site that is hosting SuSE 9 security patches.
- **suse10_pass:** Specify the password for the Novell web site that is hosting SuSE 10 and 11 patches.
- **suse10_urls:** Specify the URLs for the Novell web site that is hosting SuSE 10 and 11 patches. The defaults are:
 10:
 10SP1:
 {[https://nu.novell.com/repo/\\$RCE/SLES10-Updates/sles-10-i586](https://nu.novell.com/repo/$RCE/SLES10-Updates/sles-10-i586)}

{[https://nu.novell.com/repo/\\$RCE/SLED10-Updates/sled-10-i586](https://nu.novell.com/repo/$RCE/SLED10-Updates/sled-10-i586)}

10SP2:

{[https://nu.novell.com/repo/\\$RCE/SLES10-SP2-Updates/sles-10-i586](https://nu.novell.com/repo/$RCE/SLES10-SP2-Updates/sles-10-i586)}

{[https://nu.novell.com/repo/\\$RCE/SLED10-SP2-Updates/sled-10-i586](https://nu.novell.com/repo/$RCE/SLED10-SP2-Updates/sled-10-i586)}

10-x86_64:

{[https://nu.novell.com/repo/\\$RCE/SLES10-Updates/sles-10-x86_64](https://nu.novell.com/repo/$RCE/SLES10-Updates/sles-10-x86_64)}

{[https://nu.novell.com/repo/\\$RCE/SLED10-Updates/sled-10-x86_64](https://nu.novell.com/repo/$RCE/SLED10-Updates/sled-10-x86_64)}

10SP1-x86_64:

{[https://nu.novell.com/repo/\\$RCE/SLES10-SP1-Updates/sles-10-x86_64](https://nu.novell.com/repo/$RCE/SLES10-SP1-Updates/sles-10-x86_64)}

{[https://nu.novell.com/repo/\\$RCE/SLED10-SP1-Updates/sled-10-x86_64](https://nu.novell.com/repo/$RCE/SLED10-SP1-Updates/sled-10-x86_64)}

10SP2-x86_64:

{[https://nu.novell.com/repo/\\$RCE/SLES10-SP2-Updates/sles-10-x86_64](https://nu.novell.com/repo/$RCE/SLES10-SP2-Updates/sles-10-x86_64)}

{[https://nu.novell.com/repo/\\$RCE/SLED10-SP2-Updates/sled-10-x86_64](https://nu.novell.com/repo/$RCE/SLED10-SP2-Updates/sled-10-x86_64)}

10SP3:

{[https://nu.novell.com/repo/\\$RCE/SLES10-SP3-Updates/sles-10-i586](https://nu.novell.com/repo/$RCE/SLES10-SP3-Updates/sles-10-i586)}

{[https://nu.novell.com/repo/\\$RCE/SLED10-SP3-Updates/sled-10-i586](https://nu.novell.com/repo/$RCE/SLED10-SP3-Updates/sled-10-i586)}

10SP3-x86_64:

{[https://nu.novell.com/repo/\\$RCE/SLES10-SP3-Updates/sles-10-x86_64](https://nu.novell.com/repo/$RCE/SLES10-SP3-Updates/sles-10-x86_64)}

{[https://nu.novell.com/repo/\\$RCE/SLED10-SP3-Updates/sled-10-x86_64](https://nu.novell.com/repo/$RCE/SLED10-SP3-Updates/sled-10-x86_64)}

11:

{[https://nu.novell.com/repo/\\$RCE/SLES11-Updates/sle-11-i586](https://nu.novell.com/repo/$RCE/SLES11-Updates/sle-11-i586)}

{[https://nu.novell.com/repo/\\$RCE/SLED11-Updates/sled-11-i586](https://nu.novell.com/repo/$RCE/SLED11-Updates/sled-11-i586)}

11SP1:

{[https://nu.novell.com/repo/\\$RCE/SLES11-SP1-Updates/sles-11-i586](https://nu.novell.com/repo/$RCE/SLES11-SP1-Updates/sles-11-i586)}

{[https://nu.novell.com/repo/\\$RCE/SLED11-SP1-Updates/sled-11-i586](https://nu.novell.com/repo/$RCE/SLED11-SP1-Updates/sled-11-i586)}

11-x86_64:

{[https://nu.novell.com/repo/\\$RCE/SLES11-Updates/sle-11-x86_64](https://nu.novell.com/repo/$RCE/SLES11-Updates/sle-11-x86_64)}

{[https://nu.novell.com/repo/\\$RCE/SLED11-Updates/sled-11-x86_64](https://nu.novell.com/repo/$RCE/SLED11-Updates/sled-11-x86_64)}

11SP1-x86_64:

{[https://nu.novell.com/repo/\\$RCE/SLES11-SP1-Updates/sles-11-x86_64](https://nu.novell.com/repo/$RCE/SLES11-SP1-Updates/sles-11-x86_64)}

{[https://nu.novell.com/repo/\\$RCE/SLED11-SP1-Updates/sled-11-x86_64](https://nu.novell.com/repo/$RCE/SLED11-SP1-Updates/sled-11-x86_64)}

- **suse10_user**: Specify the user for the Novell web site that is hosting SuSE 10 and 11 security patches.
- **sync**: Specify the targets that need to be synchronized. The default is rcs.

Patch Acquisition Parameters

To acquire patches from a command line:

1. From a command prompt on your Patch Management Server, navigate to the Patch Management directory. The default location is `<InstallDir>\data\PatchManager\`

Note: You can also use the acquisition file you created from a command line. To do this, use the config parameter.

2. Using the parameters listed in the bulleted list below, create a command line similar to the following:

```
nvdkit-rca-patch ./modules/patch.tkd acquire -bulletins MS04-*
```

where you want to acquire the patch files for only bulletins from the Microsoft web site matching a filter of MS04-.*.

Note: Parameters specified on the command line overwrite those specified in patch.cfg. Use patch.cfg for default parameters.

- **arch**: Specify the computer architecture for which you want to acquire patches separated by a comma. Valid values for the arch parameters are given in the Vendor Feed Settings in the Configuration chapter in the *Sentient Enterprise Console Online Help*.
- **bulletins**: Specify the bulletins for acquisition separated by commas. The asterisk (*) wildcard character is recognized. This is the same as the bulletins parameter in `patch.cfg`. For Red Hat Security advisories, use a hyphen (-) in place of the colon (:) that appears in the Red Hat Security advisory number as issued by Red Hat.
 - Microsoft Security bulletins use the naming convention `MSYY-###`, where `YY` is the last two digits of the year that the bulletin was issued and `###` is a sequential number of the bulletin number being released for this the year specified. Microsoft service packs are listed in the format `MSSP_operatingsystem_snumber`. To acquire sample Microsoft Operating System service packs, specify `MSSP*`. This will download sample service packs using information in the `novadigm` or `custom` folders. For example, specify `-bulletins MS00-001,MS00-029`.
 - Red Hat Security advisories are issued using the naming convention `RHSA-CCYY:###`, where `CC` indicates the century and `YY` the last two digits of the year when the advisory was issued, and `###` the Red Hat patch number. However, because the colon is a reserved character in Client Automation products, you must use a hyphen (-) in place of the colon (:) that appears in a Red Hat-issued Security advisory. Specify individual Red Hat Security advisories to Patch Management using the modified naming convention of `RHSA-CCYY-###`.
 - SuSE Security patches use the naming convention `SUSE-PATCH-####`, where `###` represents a numbering scheme provided by SuSE.

If you do not want to download any bulletins, use `-bulletins NONE`. You may want to do this when you want to only acquire agent updates.

- **config:** Use this parameter to append an alternate configuration file for acquisition to override settings in `patch.cfg`. The default is `patch.cfg`.
- **data_dir:** Specify the directory on the local computer (Patch Management Server) where you want the patches downloaded to before they are sent to your Configuration Server. Use this parameter to set where to store your patch descriptor files and patch data files in an alternate directory. The default is:
`<InstallDir>\Data\PatchManager\data\patch`.
- **force:** Use force in the following situations.
 - You previously ran an acquisition using the mode `MODEL`, and now you want to use `BOTH`.
 - You previously ran an acquisition filtering for one language (`lang`), and now, you need to acquire bulletins for another.
 - You previously ran an acquisition specifying one product, and, now, you need to acquire for another.

For example, suppose that originally you only had Windows XP computers in your enterprise, so you used `-product {Windows XP*}`. A month later, you roll out Windows Vista®. If you want to acquire the same bulletins, you will need to run the acquisition with `-product {Windows Vista*,Windows XP*}` and `-force y`.

The default is `N`. If `replace` is set to `Y`, the bulletins will be removed and reacquired, regardless of the value of `force`.

- **mode:** Specify `BOTH` to download patches and the information about the patches. Specify `MODEL` to acquire only the metadata for patches. Only the Bulletins and Numbers for the patches are downloaded, but not the actual patch files. Use this mode so that you can use the reports to expose vulnerabilities on Agent devices. `BOTH` is the default.
- **product:** Specify which products you want to include in the acquisition in the format of `vendor::product` in a comma separated list. Precede any products you want excluded with an exclamation point (!). If an include filter is not set, all products are assumed. If you provide any included filters, then the excluded filters will be a subset of the included products. Be sure to conform to the vendor's naming standards. For example, Microsoft refers to Internet Explorer using its full name, rather than a common abbreviation such as IE. For example, to include all Windows products except Windows 95, type `{Microsoft::Windows*,Microsoft::!Windows 95}`.

By default, the following Microsoft products are excluded from patch acquisition and management:

```
!Windows 95,!Windows 98*,!Windows Me,!Access*,!Excel*,!FrontPage
200[023],!FrontPage 9[78-
],!InfoPath*,!Office*,!OneNote*,!Outlook*,!PowerPoint*,!Project
200[023],!Project 98,!Publisher*,!Visio*,!Word*,!Works*
```

The following products are in the exclusion list because they are not supported by Patch Management: Microsoft Windows 95, Windows 98, Windows Me and SuSE specific products `*-yast2`, `*-yast2-*`, and `*-liby2`.

If specifying a product for exclusion on the command line, surround the complete product string filters in quotes.

- **Replace:** Set replace to Y to delete old bulletins, specified in the bulletins parameter, and then re-acquire them. This will supersede the value for force. In other words, if you set replace to Y, then any bulletin specified for that acquisition will be deleted and reacquired, whether force is set to N or Y. The default is N.
- **superseded_patches:** Set superseded_patches to Y if you want to publish the data even if a patch is marked as superseded. The default is N.
- **vendors:** Specify the vendors to acquire patches from. Example: -vendors Microsoft, RedHat, SuSE, HPUX, SOLARIS. The default is Microsoft.
- **vendor_os_filter:** Specify a filter for the vendor's operating systems in the format *vendor::operatingsystem*. Red Hat and SUSE filters for x86_64 architectures use the format: *vendor::operatingsystem-x86-64*. SUSE 10 filters indicate a relevant service pack (SP1 or SP2) directly after the operating system, as in: *vendor::operatingsystemSP1-x86-64*.
 - RedHat examples:
 REDHAT::4es,REDHAT::4ws,REDHAT::4as;
 REDHAT::4es-x86_64,REDHAT::4ws-x86_64, REDHAT::5Server,
 REDHAT::5Client,REDHAT::6Server,REDHAT::6Client REDHAT::5Server-x86_64,
 REDHAT::5Client-x86_64,REDHAT::6Server-x86_64,REDHAT::6Client-x86_64
 - SuSE examples:
 SUSE::8, SUSE::9, SUSE::10SP1-x86-64; SUSE::11; SUSE::11- x86_64
 - Do not use vendor_os_filter to specify Microsoft operating systems as they are treated as products. Use the product filter for Microsoft operating systems instead.

Database Synchronization Parameters

To synchronize the databases from a command line:

- Run the following command line from the Patch Management directory:

```
nvdkit-rca-patch ./modules/patch.tkd sync -db_type mssql -dsn patch
-dsn_user rpadmin -dsn_pass rpmdb -host localhost:3464 -class "*"
dsn is a required parameter; db_type is also a required parameter when the database type
is Oracle.
```

For example, if you only wanted to update the PRODUCT class for a SQL Server database, you would type:

```
nvdkit-rca-patch ./modules/patch.tkd sync -dsn PATCH -host
localhost:3464 - class "PRODUCT"
```

If you wanted to update the PRODUCT class for an Oracle database, you would type:

```
nvdkit-rca-patch ./modules/patch.tkd sync -db_type oracle -dsn PATCH
-host localhost:3464 -class "PRODUCT"
```

where the dsn is called PATCH and the Configuration Server is the local machine.

The parameters are described below:

- **db_type:** Specify the database type. Valid values are mssql for Microsoft SQL Server and oracle for Oracle. The default is mssql. Specify this parameter (-db_type oracle) to synchronize with an Oracle database.
- **dsn:** Specify the Data Source Name (DSN) the Patch ODBC database. This parameter is required.

- **dsn_user**: Specify the user for the dsn for the Patch ODBC database.
- **dsn_pass**: Specify the password for the user of the Patch ODBC database.
- **host**: Specify the location of your Configuration Server in URL format. This parameter is required. Use the format `radia://ipaddress:port`
 - `radia` indicates the session type to be opened to the Configuration Server.
 - `ipaddress` is the hostname or IP address of the computer hosting the Configuration Server.
 - `port` is the port number of the Configuration Server.
- **class**: Specify the classes you wish to synchronize between the Configuration Server and the Patch SQL Database. For example, if you want to synchronize only the DEVICE class, specify `class="DEVICE"`. This parameter also accepts a wildcard. The default is `""` (synchronize all classes). **commit**: Specify 1 if you want to commit changes found in the Configuration Server database to the SQL database. Specify 0 if you do not want change automatically committed. You can view the changes. By default, all changes are committed.
- **rcs_pass**: If authentication has been enabled on your Configuration Server, specify the password for the `rcs_user`.
- **rcs_user**: If authentication has been enabled on your Configuration Server, specify the `rcs_user`.

Patch Agent Update Parameters

These settings are for the maintenance of the Patch Management Agent files. For more information on this, see the *View Agent Updates* section in the *Operations chapter of the Sentient Enterprise Console Online Help*. The following settings are configured in the Patch Agent section:

- **agent_updates**: Use Publish and Distribute to publish the updates to the PATCHMGR Domain and connect them to the DISCOVER_PATCH instance. This option will distribute the updates to your Patch Management managed devices. Use Publish only to publish the update, but not connect for distribution (deployment) to Patch Management managed devices.
- **agent_os**: Specify for which operating systems to acquire the agent updates. Valid values are win32, linux and suse.
- **agent_version**: Select which Patch Management versions you would like to acquire the agent updates for. You can only publish one version to one Configuration Server.

See the following sample `patch.cfg` file. Note the use of brackets for parameters and forward slashes in directory paths. If you are specifying any of these from a command line for acquisition, be sure to use quotes around values containing spaces.

```
patch::init {
    AGENT_UPDATES PUBLISH,DISTRIBUTE
    ARCH REDHAT::*,SUSE::*,HPUX::*,SOLARIS::*,MICROSOFT::x86
    CFG_VER 7.5
    DATA_DIR { C:/Program Files/PSL/RCA/Data/PatchManager/data}
    DSN PATCH
```

```
DSN_USER sa
FORCE no
FTP_PASS {{AES256}vQP8q3G7N5j4iMhgA2QUuw==}
HTTP_RETRIES 2
LANGUAGE {}
MODE both
MODULE patch
RCS_URL radia://localhost:3464
RCS_USER RAD_MAST
REPLACE no
RETIRE {}
SECTION all
  USING_DEFAULT_PATCH_CFG Y
VENDOR_OS_FILTER {}
}
```

Chapter 5

OS Management

Sentient enables you to configure and deploy operating systems based on the target device's capabilities. You can deploy operating systems to bare metal devices (no existing operating system) or to devices currently running an existing operating system. Using the operating system management features in Sentient, you can also perform pre-operating system provisioning tasks that include applying configuration settings to hardware on your target device. You can update the BIOS firmware, configure a disk array controller, or configure the nonvolatile RAM of a target device.

Sentient enables you to migrate individual user settings and data using Personality Backup and Restore.

Sentient enables you to deploy operating systems and software to HP thin clients, such as HP t5550 Thin Client and HP t5565 Thin Client, running Windows XPE, Windows CE, and embedded Linux.

Product Media

The following DVDs are used for OS management:

- Use `iso\ImageCapture.iso` to create the reference image media.
- Use `iso\ImageDeploy.iso` to create the media used to restore an image.

Restoring Operating Systems

Sentient enables you to restore your operating system in last resort situations. Restoring the operating system provides you with a working operating system however *you will lose all data* and you may need to perform some customizations such as changing the computer name or installing the agent.

Caution: The ROM object will not be updated and therefore may not reflect the device's actual state.

Pre-requisites

The following prerequisites should be met before restoring the Operating System:

- Create the ImageDeploy media. For more information about how to create this media, see "[OS Management](#)" above.
- A working operating system stored on the network, to a cached location or on a CD/DVD.

Recovering the operating system

To recover your operating system, complete the following steps:

1. Insert the CD-ROM that you created from the ImageDeploy.iso in the \service_cd folder on the product CD-ROM.
2. Boot the target device.
3. When asked which Service OS to use, select `_SVC_LINUX_` or `SVC_PEX86_`.
4. You will see several messages and then a menu opens with the following choices:
 - a. Service OS networking (default selection if no option is chosen)
 - b. Install OS from cache partition
 - c. Install OS from CD or DVD
5. Type the number corresponding to the action you want. If you select:
 - a. Service OS Networking you must be connected to a network.
If you chose to use the Linux Service OS, and DHCP is found, you will be prompted for the Sentient server's IP address and then the appropriate OS image will be installed to your device.

or

If DHCP is not found, you will be prompted for network information such as the following before the appropriate OS image can be installed to your machine:
 - IP address for the target device
 - Default gateway
 - Subnet
 - Subnet mask
 - DNS address
 - Sentient server IP address

Deploying OS Images Using CSDB Editor

This section provides information on using the CSDB Editor to prepare your operating system images for deployment to the appropriate target devices. Sentient allows OS installations on bare metal devices, migration of existing OSs, and disaster recovery of devices.

Caution: The following are not supported on thin clients:

- Managing Hardware Configuration
- Defining Drive Layouts
- Multicast
- `getmachinename.tcl`
- Deploying OSs from CD or DVD
- Sysprep

It is important to be aware of this, because the interface for these features has been disabled. If you use these features, they will be ignored on a thin client device.

About Policy

Sentient uses the following classes in the POLICY Domain for OS Management:

- Machine manufacturers (MANUFACT)
- Machine models (MODEL)
- Machine roles (ROLE)
- Machine subnets (SUBNET)

These classes are resolved in the following order: ROLE, MANUFACT, MODEL, and SUBNET. *This order is subject to change.* See ["Advanced Topic: Assigning OSs by Using Policy"](#) below for important information about implementing policy.

Caution: When using the Machine ROLE, be aware that setting a ROLE value in the device's ROM object must be done through a special script, as this is not currently exposed in the Sentient console. See ["Advanced Topic: Configuring ROM Objects for OS Management"](#) on page 224 for more information.

Assigning OSs to Devices and Groups

Use the OS Management feature in the Sentient console to assign operating systems to individual devices or groups of devices.

Manufacturer, model, and subnet are based on attributes related to a device. Role is *not* based on a device's attributes. It refers to grouping of devices, similar to assigning policy based on departments. You can set policy based on a device's assigned role—such as server or workstation.

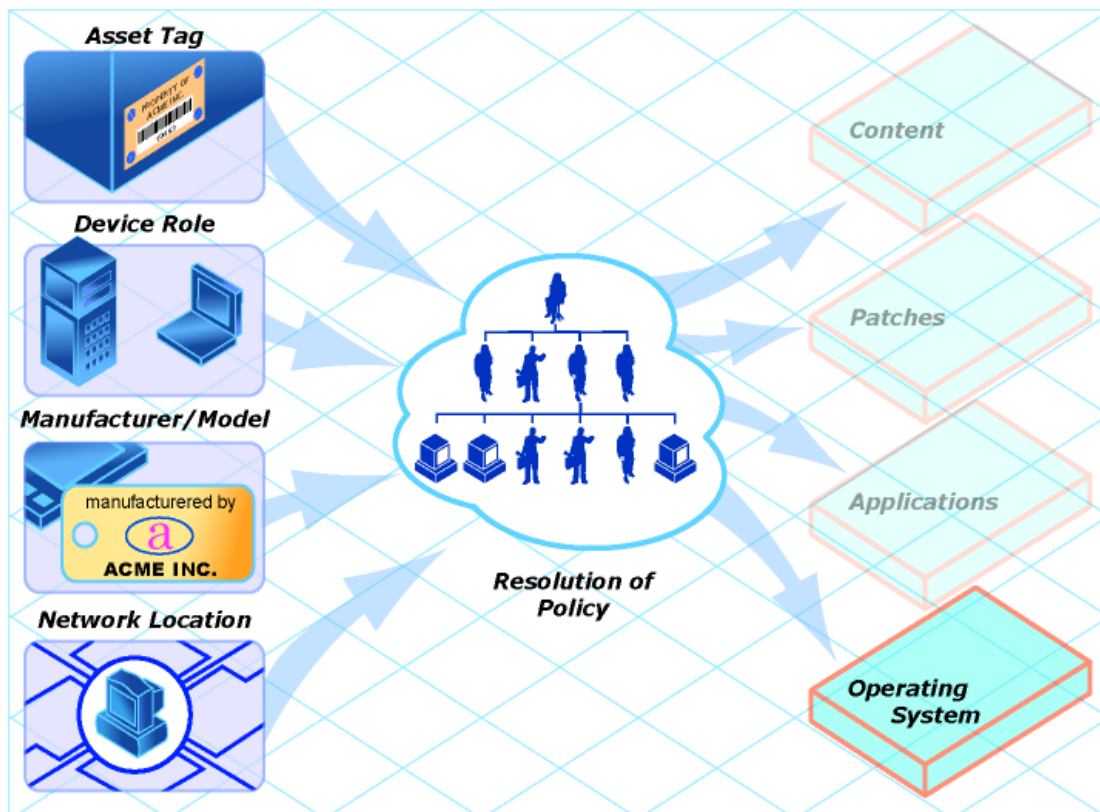
Role is the only criterion that you can use for a user to determine the OS installed on the device. Note that to allow a user to select an OS, you must set the system behaviors accordingly (see ["About System Behaviors"](#) on page 200). After a role is selected by the user, only the administrator can reset it to a different value (or to empty) so that the user may select the role again.

For information about setting the role, see ["Advanced Topic: Configuring ROM Objects for OS Management"](#) on page 224.

Advanced Topic: Assigning OSs by Using Policy

As an alternative to using the Sentient console to assign operating systems to managed devices (or groups of devices), you can use policy assignments to determine which OS is installed on a particular device. However, this is much more difficult than using the console method and must only be attempted by experienced Sentient administrators.

Resolution of Policy



To determine which criterion to use, look at your overall environment. In general, policy is assigned by subnet.

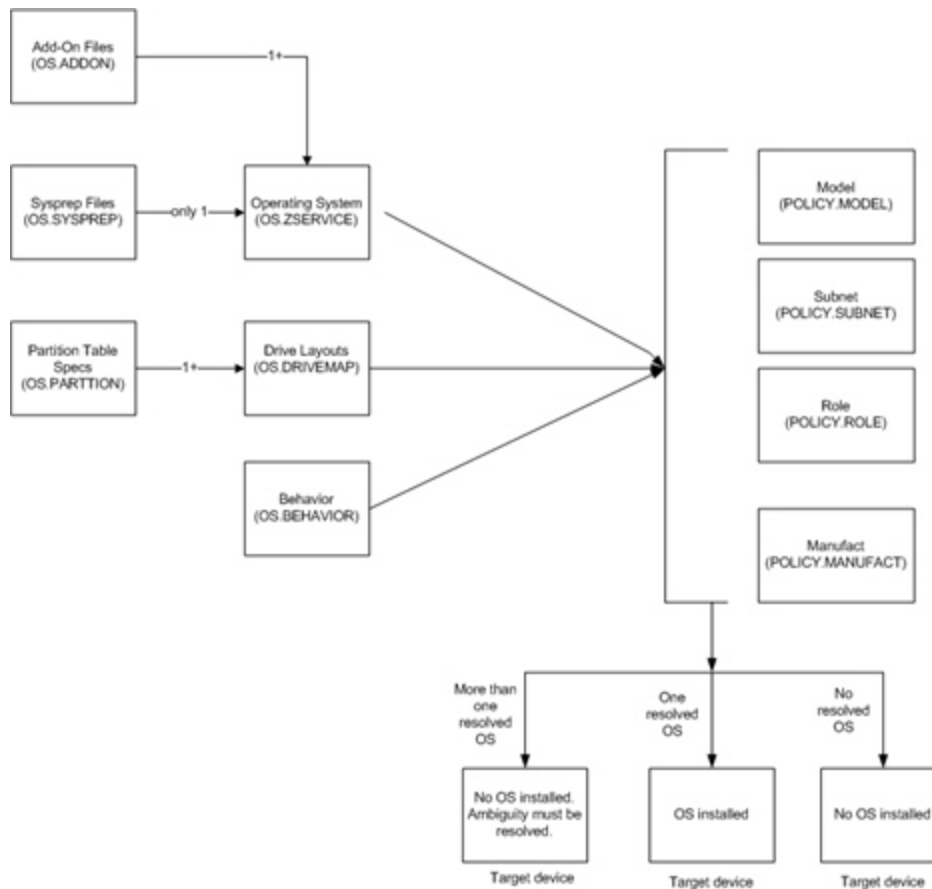
- If your environment is divided by subnets, you may choose to use the SUBNET criterion. For example, server farms are typically defined by subnets.
- If your environment is a build center, it may use the ROLE criterion so that users can select the OS to be installed.
For information about setting the ROLE, see ["Advanced Topic: Configuring ROM Objects for OS Management"](#) on page 224.
- If your environment is standardized by hardware, then you may use the MANUFACTURER or MODEL criterion. For example, one vendor makes all the laptops in your environment and a different vendor makes all the workstations in your environment, you may decide to use the MANUFACT class. These criteria will probably be used less often than the others because it may be unusual to use a certain model or manufacturer throughout your environment.

It is recommended to select a single criterion for policy. If you have used one criterion to determine policy, your OSs get deployed as expected.

If more than one criterion was used to determine policy and the machine is a bare metal machine, the user of the target device is given a list of operating systems to choose from.

The following is an overview of how the classes relate in order to determine the OS installed on a target device.

Determining OS on a Target Device



Advanced Topic: Preparing Content Using the CSDB Editor

Typically, the Sentient console is used to assign an operating system to a set of target devices and initiate the deployment. See ["Assigning OSs to Devices and Groups" on page 193](#).

In some cases, however, you may need to make use of advanced Sentient capabilities. You can use the CSDB Editor to create, modify, and prepare content in production environments. You must be familiar with the CSDB Editor to complete these tasks.

Before you begin preparing content, it is recommended that you review some typical scenarios and the procedures that you might follow when preparing to deploy OSs to your target devices. The table below provides sample scenarios and a summary of tasks that you can use in each of these situations. See the referenced descriptions listed with the individual operations to learn how to use the CSDB Editor to complete the operations.

Note: To use the following scenarios, you must be logged into the CSDB Editor as an administrator.

Advanced Administrative Procedures

If you want to...	Then...
Install an OS on a bare metal machine Note: This does not apply to Local Service Boot implementations.	1. Use the Sentient Console to create any necessary policy instances. If you are creating a manufacturer or model policy instance, see "Creating a Manufacturer or Model Instance" on page 204. 2. Use the Sentient Console to connect the OS service to the policy instances. 3. If you do not want to use the default behavior (the Undefined instance in the DEFAULT_BEHAVIOR class), you can modify the behaviors. See "About System Behaviors" on page 200. 4. Boot the target device. When the device boots up, the appropriate OS (according to policy) is installed and a ROM object is created.
Bring an unmanaged machine with an installed OS under Sentient management and install the appropriate OS as per policy. Reminder: The target device must have the Application Manager with the Sentient OS Manager feature installed.	1. Boot the target devices so that the discovery occurs. Note that the OS State is set to Desired, and the Current OS and Chosen OS are Unmanaged. 2. Use the OS Management Wizard in the Sentient Console.
Force a re-installation of the current OS without retaining any existing data	Use the OS Management Wizard in the Sentient Console. Be sure to check the Emergency Mode check box on the Deployment Options page when executing the OS Management Wizard.
Force the installation of a valid OS that you choose without retaining any existing data	1. Assign policy so that the new OS that you want to install is the <i>only</i> OS connected to policy. 2. Use the OS Management Wizard in the Sentient Console. Be sure to check the Emergency Mode check box on the Deployment Options page when executing the OS Management Wizard.
Initiate the installation of a different OS	1. Set the Select OS (PMACKOVW) behavior to <code>_NEVER_</code> to give the administrator control over policy. See "About System Behaviors" on page 200. 2. Assign policy so that the new OS that you

If you want to...	Then...
	want to install is the <i>only</i> OS connected to policy. Use the OS Management Wizard in the Sentient Console to re-evaluate the state of the OS and install a new one based on policy. Note that if you do not set the Behavior to NEVER, the user of the target device will be prompted to confirm if they want to reinstall the OS.
Allow the user to decide which OS to install	- Verify that your policy will result in more than one OS available for the target devices. - Set the PMSLCTOS behavior to <code>_LOCAL_</code>. See "About System Behaviors" on page 200. - Use the OS Management Wizard in the Sentient Console to re-evaluate the state of the OS and install a new one based on policy.
The following are additional options that can be used in many scenarios:	
Use an override Sysprep file	Connect a Sysprep instance to the operating system instance. See "Using an Override Sysprep File" on page 222 . When the OS is deployed to the target device, the override Sysprep file will be merged with the Sysprep file that is embedded in the OS
Add partitions	- Use the Drive Layouts Class to specify the type of partition. See "Defining Drive Layouts" on page 205. - Add a partition. See "Adding Partitions" on page 220. <i>All existing data will be lost.</i> - Assign the appropriate drive layouts to your target devices. See "Assigning Drive Layouts" on page 222.
Create a replace, cache, or merge type partition	- Use the Drive Layouts class to specify the type of partition. See "Defining Drive Layouts" on page 205. - Assign the appropriate drive layouts to your target devices. See "Assigning Drive Layouts" on page 222.

Logging On

To log on to the Sentient Enterprise Administrator CSDB Editor:

1. Go to **Start > All Programs > Sentient Enterprise Administrator > Sentient Enterprise Administrator CSDB Editor**.
2. In the **User ID** text box, type `admin`.
3. In the **Password** text box, type a password. Passwords are case sensitive. The pre-defined password is `secret`.

Caution: Be sure to change your password before moving the CSDB Editor into your production environment.

4. Click **OK**.

About the OS Management Classes

The following are the classes you may need to use when preparing operating system content.

Caution: The CSDB Editor is an open system. You must have a comprehensive understanding of how to use the CSDB Editor and the tasks that you want to perform to prevent unintended consequences.

Except for specific instance attributes detailed in this guide, do not change, edit, or delete any of the classes in the OS domain.

- Do not change any of the `_BASE_INSTANCE_` wiring
- Do not change (or otherwise add) `_NULL_INSTANCE_`
- Do not change `ZxxxPRI` attribute values
- Do not re-order the connections in any of the instances
- Do not change any of the expressions in any of the instances

Part of the implementation of OS management in Sentient is contained in the classes and instances of the OS domain. Any change to anything other than the instance attributes detailed in this guide may render the system unusable and void support.

Viewing the OS Management Classes

1. Open the CSDB Editor and go to PRIMARY.OS.
2. In the list view, the following classes appear.
 - Behavior (BEHAVIOR)
Lists the settings for how the OS management features behave. You can assign different system behaviors to different target devices. See ["About System Behaviors" on the next page](#).
 - Drive Layouts (DRIVEMAP)
This class lists the types of partitions that you can add or copy, and also allows you to configure new partitions. See ["Defining Drive Layouts" on page 205](#).
 - HW Config (LDS)
Stores instances that contain the information about how a target device's hardware must be configured in order for it to be ready for operating system installation.
 - HW Config Element (LME)
Stores instances that contain information about the resources required for a Hardware Configuration Management operation, the sequencing of operations, and how the operation is to be carried out.
 - AddOn Resources (ADDON)
If you use the option **OS Add-ons/Extra POS drivers** in the Publisher, the directories or files that you select will be published to the ADDON class. There is no need (nor any support) to edit these instances directly.

Note: OS services published to the CSDB with HPCA 7.50 or later will have a generic connection pointing to the ADDON class. Any directories or files published using the **OS Add-ons/Extra POS drivers** option in the Publisher will be included automatically in the OS deployment.

If you migrated from HPCA version 5.11 or 7.2x, you must manually add the connection to the OS.ZSERVICE instance:

OS.ADDON.<InstanceNameOfOSService>_*

For example:

OS.ADDON.WIN7X86_*

Place the value in the 6th `_ALWAYS_` connection field.

- Mobile File Resource (RMMFILE)
File resources for mobile devices.
- Operating Systems (ZSERVICE)
Stores the OS services to be deployed to your target devices.
- OS Packages (PACKAGE)
Used to combine multiple files into packages.
- OS Path (OSPATH)
A controlling class used by Sentient. Do not edit.

- **ELIGIBLE (ELIGIBLE)**
A controlling class used by Sentient. Do not edit.
- **OS Resources (FILE)**
OS resources, such as `WIN7.WIM`.
- **Partition Table Spec (PARTTION)**
Lists the specifications for the partitions that you may add in addition to the OS boot partition. See ["Adding Partitions" on page 220](#).
- **STATE (STATE)**
A controlling class used by Sentient. Do not edit.
- **Sysprep Files (SYSPREP)**
Lists the Sysprep files and `unattend.txt` files stored in your database. See ["Using an Override Sysprep File" on page 222](#).
- **UNIX Config Files (UNIXCFG)**
UNIX configuration resource class.

About System Behaviors

You can assign system behaviors to your target devices based on policy. If you do not assign a behavior to policy, `DEFAULT_BEHAVIOR` is the default.

For example, you may want to configure some managed devices for the user to acknowledge the change in the OS, while others may not require user acknowledgement.

Caution: You must be very careful if you are using more than one Behavior instance, because these instances determine the behavior of the system. You may have unintended consequences if this is not performed properly. For example, if you set the wrong policy, you may inadvertently allow users to make policy changes, or an unattended device may get stuck at a prompt.

It is highly recommended that you connect one Behavior instance to one Policy instance only.

One potential way to prevent errors would be to connect Behavior instances to mutually exclusive instances of different policies.

Setting the Behaviors

1. In the CSDB Editor, go to PRIMARY.OS.BEHAVIOR.
2. Create a new instance or modify an existing instance (see the table *Attributes of the Behavior Class*).

Note: To create or modify instances, see *Sentient Enterprise Console Online Help*.

3. After making changes, click **OK**.
4. Connect the BEHAVIOR instance to a POLICY instance.
 - Connect only one BEHAVIOR instance per POLICY instance.
 - If you are using a Core and Satellite environment, you may need to first remove the DEFAULT_BEHAVIOR connection from the ROLE base instance.

Attributes of the Behavior Class

Attribute	Description
Name of this instance	Instance Name
PMROLE	PMROLE must no longer be used—it cannot be used from the Sentient Console. The following information is provided for historical reference only. Indicates if the user is allowed to select a machine role. • _LOCAL_ Displays a user interface so a user at the target device can select a role for the device. The list of available roles, determined from the instances in the POLICY.ROLE class in the CSDB, appears. • _CENTRAL_ Disables the ability to select roles. A role selection remains in effect until you (the administrator) void or overrule the selection. Default: _CENTRAL_
PMSLCTOS	Specifies whether to provide the user with the choice of Operating Systems, if there are more than one OS/ZSERVICE instances available, to be installed. • _LOCAL_ (Default) Prompts the user with the choice of OS installation during deployment. • _CENTRAL_ Chooses the value automatically, based on the policy set by the administrator.
PMACKOVW	Specifies whether to prompt the user before overwriting or modifying the OS. • _ALWAYS_ (Default) Prompts the user before reinstallation of OS

Attribute	Description
	• _NEVER_ Does not prompt the user, but installs the OS Caution: NEVER is designed for use with unattended devices. Use this option with caution, as the user will not be prompted before the OS is overwritten. • _VALID_ This option is deprecated
PMINITL	Specifies if an OS must be installed over an existing file system on a recently discovered, but unmanaged device. The PMINITL attribute is referenced only if there is no <code>rombl.cfg</code> on the device. If there is a <code>rombl.cfg</code>, this indicates that the device is already under management and PMINITL is not referenced at all. • _LOCAL_ (default) Prompts the user • _KEEP_ Does not prompt the user and keeps the current OS • _REINSTALL_ Does not prompt the user and reinstalls the operating system, regardless of what exists
PMDISRCV	Specifies the action to be taken when there is no valid bootable partition. • If PMDISRCV = _CONFIRM_, the target device shuts down so that the administrator can recover data from the target device. • If PMDISRCV = _AUTO_, the appropriate OS is reinstalled.
RUNPARAM	Specifies the parameters that are appended to the radskman command line. This command line runs after the OS is installed, and it installs the target device's applications. For additional parameters, see the Persistent Systems Software Support web site. Be sure to specify the IP address or DNS name for your Configuration Server. If you do not modify this parameter, your target device will not be able to successfully run an Sentient OS connect. Do not remove the <code>cop=y</code> parameter; it is necessary to enable COP. In the RUNPARAM (RunOnce Parameter String), change <code>IP=RCSSERVER</code> to reference the appropriate Sentient server for your environment. If your server is running on a non-default port, also add: <code>,port=ConfigurationServerPortNumber></code> The default port is 3464.
ROMAPARAM	Typically, use this only if instructed by Technical Support.
BANDWIDTH	The bandwidth throttle used by each target device. For example, 1000K. You can specify bandwidth throttle in Kbs (K), MB/sec (M), or GB/sec (G). The default definition is in bytes/sec. The default value is blank (no

Attribute	Description
	bandwidth limitation), which means that the download process runs at the maximum speed of the network interface.
KBDMAP	Sets the keyboard mappings: • en (default) loads English keyboard mappings • fr loads French keyboard mappings • de loads German keyboard mappings For OS deployment using the Windows PE service OS, the following additional values are available: • it Italian • pt Brazilian Portugese • es Spanish
LANG	Specifies the language to be supported. • en_US = English • zh_CN = Simplified Chinese • ja_JP = Japanese • pt_BR = Brazilian Portuguese • fr_FR = French • de_DE = German • it_IT = Italian • es_ES = Spanish
ACKTMOUT	Specifies how long ACKTMOUT waits before assigning the default AUTOROLE. • Set ACKTMOUT = 0 to disable the timeout • Set ACKTMOUT = <i>number of seconds</i> to wait for the specified length of time before continuing This functionality should no longer be used. In a Core and Satellite environments, it cannot be used from the Sentient console.
AUTOROLE	AUTOROLE should no longer be used—it cannot be used from the Sentient Console. The following information is provided for historical reference only. The ROLE that is assigned if a timeout occurs.
SYSCHECK	Go to the RCA CSDB editor and navigate to OS> Operating System> Behavior> DEFAULT> DEFAULT_BEHAVIOUR and set the [_NEVER_ / _ALWAYS_ / _IGNORE_ / _VALID_] parameter value as required. • _NEVER_: Never validates whether a device meets the requirements for Windows 11 upgrade or installation.

Attribute	Description
	• _ALWAYS_: Validates whether a device meets the requirements and stops the process if the requirements are not met. • _IGNORE_: Validates whether a device meets the requirements and logs a warning message in case of the requirements are not met, but it allows the installation or the upgrade. • _VALID_: Validates whether a device meets the requirements and logs a warning message in case the requirements are not met, but it allows the users to decide whether they want to proceed with the installation or the upgrade. Note: It is applicable for OS manager Windows 11 upgrade. For more information, refer to the Managing Windows 11 Feature Upgrades and Service Updates section in the 11.0 System Admin Guide.

Creating a Manufacturer or Model Instance

You can assign OS policy based on various criteria. When you want the policy to be dependent on the device manufacturer or the device model, there is a certain naming convention that must be followed.

To create a manufacturer or model instance:

1. In the CSDB Editor, go to PRIMARY.POLICY.MODEL or PRIMARY.POLICY.MANUFACT.
2. Right-click the class name, and select **New Instance**.
3. Type the Display name and the Instance name.

Note: You must use the manufacturer or model information that is stored in the ROM object. The reason for this is that the instance name must correspond with the data derived from SMBIOS. For example, Hewlett-Packard would be HEWLETT_PA. You cannot use spaces and are restricted to ten characters.

When naming the model instance, it must be named as nvdmanufact_nvdmodel.

For example, if you have an HP Compaq dc7700 Small Form Factor machine, manufacturer (nvdmanufact) is displayed as HEWLETT_PA and the model (nvdmodel) is displayed as COMPAQ_DC7700_SMALL in the ROM object. The name of the Model instance for this machine must be HEWLETT_PA_COMPAQ_DC7700_SMALL.

4. Click **OK**.

Assigning Operating Systems

You must assign the appropriate operating systems to your target devices based on policy such as machine type, manufacturer, model, role, or subnet.

To assign operating systems:

1. In the CSDB Editor, go to PRIMARY.OS.ZSERVICE.
2. Select the appropriate OS service.
3. Connect the OS Service to a PRIMARY.POLICY instance.

Defining Drive Layouts

Sentient supports the ability to:

- Create one or more data partitions in addition to the boot partition
or
- Create a copy of your new OS image and its supporting files on a hidden partition to be used for recovery

For all supported operating systems, you can use the Drive Layouts class to specify the type of partitioning strategy used. For Windows Server 2012, Windows Server 2012 R2, Windows 10, Windows 8.1, Windows 8, Windows 7, Windows Vista, Windows 2008 R2, and Windows 2008, you can also specify disk space allocated to each partition. Partitioning is supported for the boot drive only.

For details, see the following topics:

- ["Partitioning Strategies" on the next page](#)
- ["Allocating Disk Space for Partitions" on page 208](#)
- ["Special Considerations for Dual-Partition Installations" on page 214](#)
- ["Specifying the Drive Layout" on page 220](#)

Caution: It is recommended to connect a Drive Layout instance to only one Operating System or Policy instance to prevent conflicting definitions.

It is possible that multiple Drive Layout instances may be resolved for an installation. Only the first resolved instance will be used, other instances will be ignored.

Partitioning Strategies

You can use the following attributes in the DRIVEMAP class to specify how Sentient must partition the hard disk prior to installing an operating system on a target device:

DRIVEMAP Attributes for Partitioning

Attributes Name	Default
Type	Merge
Reserved Space Size	0
System Partition Size	1024

The following table describes the possible values for the DRIVEMAP Type attributes:

DRIVEMAP Type Attribute

Type	Description
REPLACE (default in Classic environments)	Replaces the current partitioning on the target device with a single or dual-partition installation as defined for, or included with, the OS image being installed. If there are no DRIVEMAP instances connected to the OS being installed, this is the default method. IMPORTANT: If you use REPLACE, <i>all existing data will be lost</i> .
ADD	Same as REPLACE, and this option additionally creates one or more data partitions in an extended partition at the end of the hard disk. See "Adding Partitions" on page 220 for more information. IMPORTANT: If you use ADD, <i>all existing data will be lost</i> .
MERGE(default in Core and Satellite environments)	Use for migration purposes. Replaces or updates an OS on a machine where existing data must be preserved. MERGE overlays only the existing "System Reserved" (if applicable) and OS partition and does not touch data on any other partitions. • If the partitions to be installed are larger than the space already defined for these partitions, the installation fails. • If the target drive does not contain existing partitions (for example, bare metal), then MERGE auto-switches into REPLACE mode. See "Special Considerations for Dual-Partition Installations" on page 214 for additional information about using MERGE with Windows Server 2012, Windows Server 2012 R2, Windows 10, Windows 8.1, Windows 8, Windows 7, and Windows Server 2008 R2.
CACHE	Creates a hidden back-up partition at the end of the target drive. The size of the partition is dynamically determined by the size of the OS installation image. All files necessary to reinstall the OS is saved (in compressed form) in this partition. IMPORTANT: If you use CACHE, <i>all existing data will be lost</i> . For more information about restoring this image, see "Restoring Operating Systems" on page 191 .

Type	Description
PRES	Allows you to preserve a set of files and folders on a target device during the installation of a new operating system and restores them after OS installation. Note: This requires the ImageX method of OS deployment. Any attempt to use any other deployment method will result in an error. To do this: - Before the target device is rebooted to install the new OS, the files and folders to be preserved must be placed in the folder <code>C:\OSMGR.PRESERVE</code>. The folder name is in upper case and is case-sensitive. If you create the folder in small case, the preserve feature does not work. It is recommended that you use <code>NOVAPDC</code> to do this. However, any method (including manual) that results in the desired files and folders being placed in the named folder is acceptable. - During the resolution and deployment process, if this Partition Type is resolved for the target device, no disk repartitioning is performed. The existing (NTFS) root file system is kept intact, and all contents of the file system except for the contents of <code>C:\OSMGR.PRESERVE</code> are removed. - The new OS image is deployed to the (preserved) file system. - After the machine reboots into the newly deployed OS, the files and folders in <code>C:\OSMGR.PRESERVE</code> are available to be restored. It is recommended that you use <code>NOVAPDR</code> to do this. However, any method (including manual) that results in the desired files and folders being restored properly is acceptable. Note that all data in <code>C:\OSMGR.PRESERVE</code> remains until explicitly removed by the (user-defined) restore process. - If the target drive does not contain existing partitions (for example, bare metal) then PRES auto-switches into REPLACE mode. Note: You cannot use this PRES if your target device has a “System Reserved” partition in addition to the OS partition.

Allocating Disk Space for Partitions

The following information applies only to OS installations using ImageX and Windows Setup deployment methods (Windows PE Service OS only).

You can use the RSVDSPCE and SYSPSPCE attributes of the DRIVEMAP class to control how the hard disk is partitioned on a target device before the OS is deployed:

- **CSDB:OS.DRIVEMAP.RSVDSPCE**
Leave un-partitioned free space in the beginning of the drive selected for deployment. This free space can be used later for Microsoft BitLocker enablement (Windows Vista, Windows 2008) or for other purposes such as a recovery partition.
- **CSDB:OS.DRIVEMAP.SYSPSPCE**
Create a "System Reserved" partition of the specified size when installing Windows Server 2012, Windows Server 2012 R2, Windows 10, Windows 8.1, Windows 8, Windows 7, or Windows 2008 R2. If the value of this attribute is greater than zero, Windows Server 2012, Windows Server 2012 R2, Windows 10, Windows 8.1, Windows 8, Windows 7, or Windows 2008 R2 is installed in a dual-partition setup including a "System Reserved" partition of the specified size plus the operating system partition itself.

Note: You must specify the values of these attributes in Megabytes. For example, specifying 2000 means 2 GB.

RSVDSPCE and SYSPSPCE work differently depending on the DRIVEMAP Type and the OS being deployed. The following sections provide the details:

- ["Post-Windows 7 Operating Systems" on the next page](#)
- ["Pre-Windows 7 Operating Systems" on page 212](#)

Post-Windows 7 Operating Systems

For Windows Server 2012, Windows Server 2012 R2, Windows 10, Windows 8.1, Windows 8, Windows 7, and Windows Server 2008 R2 installations, you can specify the size of the System partition and unallocated reserved space using the RSVDSPCE and SYSPSPCE attributes.

Partitioning in Windows Server 2012, Windows Server 2012 R2, Windows 10, Windows 8.1, Windows 8, Windows 7, and Windows Server 2008 R2

DRIVEMAP Attribute	Description
SYSPSPCE	The SYSPSPCE attribute specifies the size of the System partition in MB. • If no value is specified for SYSPSPCE, the System partition is 1 GB. • If SYSPSPCE = 0 (zero), the System partition is not created. • If the value of SYSPSPCE specified is less than 1000 (1 GB), a warning is generated.
RSVDSPCE	The RSVDSpace attribute specifies how much unallocated disk space must be set aside before the OS partitions in a single-partition scenario or before the System and OS partitions in a dual-partition scenario. This free space can be used for two purposes. It can be used for BitLocker encryption, or it can be used to create a RECOVERY partition at a later time. When deploying Windows Vista, RSVDSpace can be used to reserve space on the hard disk to make the system BitLocker-ready without creating a System partition. For more information, see "Using Microsoft BitLocker" on page 232. When deploying Windows Server 2012, Windows Server 2012 R2, Windows 10, Windows 8.1, Windows 8, Windows 7, or Windows Server 2008 R2, the System partition defined by the SYSPSPCE attribute is used for BitLocker. If you use RSVDSpace to set aside space for a RECOVERY partition, that partition can subsequently be created, initialized, and populated in one of two ways: • Using an Sentient exit-point routine • Using a separate Sentient Service installed through the Sentient agent after the production OS is up and running

RSVDSPCE and SYSPSPCE Behavior for Windows Server 2012, Windows Server 2012 R2, Windows 8.1, Windows 8, Windows 7, and Windows Server 2008 R2

Type	RSVDSPCE Reserved Space Size	SYSPSPCE System Partition Size	Comments
REPLACE	Honored	Honored	
MERGE	Honored	Honored	See also "Special Considerations for Dual-

Type	RSVDSPACE Reserved Space Size	SYSPSPACE System Partition Size	Comments
			Partition Installations" on page 214
ADD	Honored	Honored	
CACHE	Honored	Not Applicable	OS installed in a single partition setup (no "System Reserved" partition)
PRES	Not Applicable	Not Applicable	No repartitioning done at all. Operating system partition is cleared with the exception of the Preserve directory.
Deploy CD Install OS from Cache	Not Applicable	Not Applicable	Requires prior OS deployment with DRIVEMAP type CACHE. OS is redeployed to the active OS partition.
Deploy CD Install OS from CD	Not applicable	Not Applicable	Wipes all partitions and installs the OS on a single partition

Example 1 – Reserve 2 GB for a RECOVERY Partition in a Dual-Partition Scenario

RSVDSPCE = 2048		SYSPSPCE = 1024
2 GB Reserved Space	1 GB System Partition	OS Partition (Windows 7)

Pre-Windows 7 Operating Systems

The following information pertains to the following operating systems:

- Windows XP
- Windows 2003
- Windows Vista
- Windows 2008

This information applies only to OS installations using ImageX and Windows Setup deployment methods (Windows PE Service OS only).

RSVDSPCE and SYSPSPCE Behavior for Pre-Windows 7 OSs

Type	RSVDSPCE Reserved Space Size	SYSPSPCE System Partition Size	Comments
REPLACE	Honored (see Comments)	Not Applicable	RSVDSPCE is only honored for Windows Vista and Windows Server 2008. It is not applicable to Windows XP or Windows Server 2003.
MERGE	Honored (see Comments)	Not Applicable	
ADD	Honored (see Comments)	Not Applicable	
CACHE	Honored (see Comments)	Not Applicable	
PRES	Not Applicable	Not Applicable	No repartitioning done at all. Operating system partition is cleared with exception of Preserve directory.
Deploy CD Install OS from Cache	Not Applicable	Not Applicable	Requires prior OS deployment with DRIVEMAP type CACHE. OS is redeployed to the active OS partition.
Deploy CD Install OS from CD	Not Applicable	Not Applicable	Wipes all partitions and installs OS on single partition

Example 2 – Reserve 2 GB in a Single Partition Scenario for Pre-Windows 7 OS

RSVDSPCE = 2048		SYSPSPCE = 1024	
2 GB Reserved Space		OS Partition (Windows Vista)	

Special Considerations for Dual-Partition Installations

Sentient can install Windows Server 2012, Windows Server 2012 R2, Windows 10, Windows 8.1, Windows 8, Windows 7, and Windows 2008 R2 (and later) operating systems using dual partitions or a single partition. The MERGE strategy works differently, depending on the values of the RSVDSPCE and SYSPSPCE attributes and the number of partition table slots available:

- If SYSPSPCE is greater than zero, one empty partition table slot is required to create the System partition.
- Similarly, if RSVDSPCE is greater than zero, one empty partition table slot is required to set aside the unallocated disk space.
- If both SYSPSPCE and RSVDSPCE are greater than zero, two empty partition table slots are required to create the specified layout.
 - If no empty slots are available, neither the System partition nor the unallocated space is created.
 - If only one empty slot is available, the unallocated space is set aside, but the System partition is not created.

The examples that follow depict a single-partition Windows XP installation being upgraded to a dual-partition Windows 7 installation with a 1 GB System partition.

Example 1 – Single Partition Windows XP to Dual Partition Windows 7

RSVDSPCE = 0	SYSPSPCE = 1024	Empty partition table slots = 1

Original Layout:

OS Partition (Windows XP)	Data 1

New Layout:

1 GByte System Partition	OS Partition (Windows 7)	Data 1

If an empty partition table slot is not available, Sentient creates a single partition Windows 7 installation.

Example 2 – Same Upgrade, No Free Partition Table Slots

RSVDSPCE = 0	SYSPSPCE = 1024	Empty partition table slots = 0

Original Layout:

OS Partition (Windows XP)	Data 1	Data 2	Data 3

New Layout:

OS Partition (Windows 7)	Data 1	Data 2	Data 3

Example 3 – Single Partition Windows XP to Dual Partition Windows 7 with Unallocated Space

RSVDSPCE = 1024	SYSPSPCE = 1024	Empty partition table slots = 2

Original Layout:

OS Partition (Windows XP)	Data 1

New Layout:

1 GB Reserved Space	1 GB System Partition	OS Partition (Windows 7)	Data 1

However, if only one empty partition table slot is available, only the unallocated space is set aside.

Example 4 – Same Upgrade, Only 1 Empty Partition Table Slot

RSVDSPCE = 1024	SYSPSPCE = 1024	Empty partition table slots = 1

Original Layout:

OS Partition (Windows XP)	Data 1

New Layout:

1 GB Reserved Space	OS Partition (Windows 7)	Data 1

If no empty partition table slots are available, a single partition installation is implemented.

Example 5 – Same Upgrade, No Empty Partition Table Slots

RSVDSPCE = 1024	SYSPSPCE = 1024	Empty partition table slots = 0

Original Layout:

OS Partition (Windows XP)	Data 1

New Layout:

OS Partition (Windows XP)	Data 1

Specifying the Drive Layout

To specify a drive layout:

1. In the CSDB Editor, go to PRIMARY.OS.DRIVEMAP
2. Create a new instance
3. Open the instance, and double-click **Type** to specify the type of partition to be created. The Editing window appears.
4. In the text box, type ADD, REPLACE, CACHE, MERGE, or PRESERVE (see ["Partitioning Strategies" on page 206](#))
5. In the Editing window, click **RSVDSPCE**. Specify a value in MB.
6. Click **SYSPSPCE** and specify a value in MB
7. Click **OK**

Adding Partitions

You can create a new layout that contains a boot partition and one or more logical data partitions at the end of the hard disk in a single, extended partition. These partitions are in addition to the OS boot partition. Partitions are added from the “back” of the disk to the “front.”

Caution: All existing data will be lost.

Note: There is a limit of four *physical* partitions on a hard drive, and only one partition may be an extended partition (which may contain any number of logical drives).

Also, if you start with a single physical drive, such as:

Partition	Logical Drive
Primary	C
Extended	D
	E
	F

If you then add a second hard drive, the drive letter mappings are reassigned so that the primary partitions are in an alphabetical sequence. For example:

Drive 1

Partition	Logical Drive
Primary	C

Partition	Logical Drive
Extended	E
	F
	G

Drive 2

Partition	Logical Drive
Primary	D
Extended	H
	I
	J

Caution: The partition is added after the boot partition. Make sure you allow enough space for the OS. Note that if the total requested space exceeds the capacity of the drive where the OS is being installed, the installation fails.

To create partitions, follow these steps:

1. In the CSDB Editor, go to PRIMARY.OS.PARTTION
2. Create a new instance
3. Open the instance
4. Set the PARTTION class attributes as required

PARTTION Class Attributes

Attribute	Description
PARINFO	Identifies the name of the partition
SIZE	Specifies the partition size specified as a percentage of the hard drive or in MB. These values equal the total hard drive space.
UNITS	Indicates if the partition size is being specified as a percentage or in megabytes
FORMAT	Specifies whether or not to format the drive
PARTYPE	Indicates the type of partition: NTFS, FAT32, EXT2, EXT3, or QNTFS. EXT2 and EXT3 are not supported under the WinPE Service OS. Note that QNTFS performs a quick format without zeroing out the partition.

5. Connect the PARTTION instance to the corresponding DRIVEMAP instance.

Assigning Drive Layouts

After you have created your Drive Layout (DRIVEMAP), you must assign the appropriate drive layouts to your target devices based on policy such as machine manufacturer, model, role, or subnet.

To assign drive layouts:

1. In the CSDB Editor, go to the appropriate POLICY instance, such as a SUBNET instance.
2. Connect the appropriate DRIVEMAP instance to the POLICY instance.
In Core and Satellite environments, remove the DEFAULT_DRIVEMAP from the ROLE base instance. Only one connection is allowed.

Note: You can add, merge, replace, or cache partitions. You cannot do more than one of these actions.

Using an Override Sysprep File

You can assign a `Sysprep.inf` that is separate from the gold image to allow the same image to be set up differently on target devices. The override `Sysprep.inf` is merged with the embedded `Sysprep.inf`. During the merge, the values in the override `Sysprep.inf` take priority. If a value is not specified in the override `Sysprep.inf`, the keyword is removed.

In the [GUIRUNONCE] section of the `Sysprep.inf`, the lines in the file are merged based on their position in the file. Two edit functions are supported in this section. If you type a + in the override `Sysprep.inf`, it keeps the corresponding line from the embedded `Sysprep.inf`. If you type a - in the override `Sysprep.inf`, it removes the corresponding line from the embedded `Sysprep.inf`.

Here is an example of a sysprep file that is embedded in the image, an override sysprep file, and the result of the merging of these files using the edit functions.

Example of Resulting Sysprep File Using Edit Functions

Sample of sysprep file in the image	Override sysprep file	Sample of resulting sysprep file
[Unattended] OemSkipEula = NoExtendOemPartition = 0 [Identification] JoinWorkgroup = "WORKGROUP" [guirunonce] C:\TEMP\ KEEPRUNNINGTHIS.CMD C:\TEMP\ ANDRUNTHIS.CMD C:\TEMP\	[Unattended] OemSkipEula = YesExtendOemPartition = 1 [Identification] JoinWorkgroup = JoinDomain = "TESTDOM1" [guirunonce]+ C:\TEMP \RUNTHISONETOO.CMD- C:\TEMP \STOPRUNNINGTHIS.CMD	[Unattended] oemskipeula= Yesextendoempartition =1 [Identification] joindomain= "TESTDOM1" [guirunonce] C:\TEMP \KEEPRUNNINGTHIS.CMD C:\TEMP \ANDRUNTHIS.CMD C:\TEMP

Sample of sysprep file in the image	Override sysprep file	Sample of resulting sysprep file
STOPRUNNINGTHIS.CMD		\RUNTHISONETOO.CMD

Caution: The `Sysprep.inf` file must not be greater than 800 KB in size.

Creating an Override Sysprep.inf

1. Modify `Sysprep.inf` to contain the appropriate information.
2. Use the Publisher to publish the new `Sysprep.inf` file to the OS domain, Sysprep Files (SYSPREP) class.

Note: In the Publisher, from the **Type of Data to Publish** drop-down menu, select **OS Image**. Then, you can select the appropriate `Sysprep.inf` file that you want to use.

3. Use the CSDB Editor to connect the PRIMARY.OS.SYSPREP instance to the appropriate OS (PRIMARY.OS.ZSERVICE instance). You can only attach one Sysprep file to an OS. If the OS does not have this connection, the embedded `Sysprep.inf` file is used.

Note: Currently, the COMPNAME and DOMAIN from the ROM object displayed in the Manager is used in `Sysprep.inf`, whether `Sysprep.inf` was embedded in the image or published separately.

Note: Consider running a manual test of `Sysprep.inf` to verify the accuracy of the file prior to using the Image Preparation Wizard. Note that if you run Sysprep and have `extendoempartition = 1`, the partition is extended after Sysprep runs. To deliver the same OS with varying setup behaviors, you can create multiple OS services. Each OS service can contain the same OS image, yet each may have a different `Sysprep.inf` attached to it.

Advanced Topic: Configuring ROM Objects for OS Management

In earlier versions of Client Automation, the Management Portal User Interface (UI) was used to configure ROM objects representing devices (or device groups) in the Management Portal repository. This UI is no longer available in Sentient.

Sentient now provides a tool called `osmkit.tkd` that you can use to do the following operations:

- Create, delete, or update a device or device group
- Bring devices under OS management (create a ROM object for the device)
- Set any ROM object attribute (including ROLE)
- Assign policy to a device or device group

You can perform these operations on one or more devices. This is helpful if you want to automate OS management tasks.

The `osmkit.tkd` script reads an XML configuration file and uses web services calls to update ROM objects. You need not—and must not—modify the script. You can customize the XML configuration file. Sample configuration files are provided.

Prerequisites

Before you can use the `osmkit.tkd` tool, you must perform the following:

1. In a command line window, go to the following directory on your Sentient Core server:
`<InstallDir>\OSManagerServer\osmkit`
2. Run the following commands:
`copy ..\nvdkit-rca-osm.exe .\nvdkit.exe`
`copy ..\nvdcrt.tkd .\nvdcrt.tkd`

Syntax

You can use one of the following syntax to configure ROM objects for OS Management:

- `nvdkit osmkit.tkd -host <host> -port <port> -usessl <0/1> -userid <userid> -password <password> -input <input> -preview <0/1>`

Parameter	Mandatory	Description
-host	yes	Host name of the Sentient Core server
-port	yes	Port name for the Sentient Core server (3466 by default)
-usessl	no	If set to 1, the script uses SSL for communication. Default value is 0.
-userid	yes	Sentient user ID
-password	yes	Password for this Sentient user. OSMKIT also supports encrypted password. You <i>must</i> encrypt password using NVDKIT only.
-input	yes	Name of the XML configuration file, (must be located in the <code>osmkit</code> directory)
-preview	no	If set to 1, the script reads the XML configuration file but does not actually make any changes. If set to 0, the script makes the changes specified. The default value is 1.

Example:

```
nvdkit osmkit.tkd -host rcaserver -port 3466 -usessl 0 -userid admin  
-password secret -input myconfig.xml -preview 0
```

- Create a customized configuration file and place it in the `osmkit` directory. Use the following syntax to use your customized configuration file:

```
nvdkit.exe osmkit.tkd -cfg <custom.cfg>
```

In this instance, `<custom.cfg>` is the customized configuration file that you create based on the following sample configuration file.

Sample Configuration File

```
host          localhost  
port          3466
```

```
userid          admin
pass            "{AES256}3gMlspmbrGbqVXNPDx8tWg=="
directorysource internal
input           test.xml
preview         1
usessl          0
```

Working with the Configuration File

In an Sentient Core server installation, the sample XML configuration files for `osmkit.tkd` are located here:

```
<InstallDir>\OSManagerServer\osmkit
```

There are three types of actions that can be specified in the configuration file:

- **Device:** Device action are specified using the `<device>` element. These actions operate at device level and pertain to a device. You can identify a device by specifying a Common Name using the "name" attribute or by matching the attribute to one of the parameter, such as `cn`, `compguid`, `compname`, `computerbl`, `dnshostname`, `detecteddnshostname`, `hostname`, `ipaddress`, `nvdhwlana`, `nvdmachid`, `smsystemserialnumber`, and `smsystemuuid`. The following table lists the attributes for `<device>` element:

Device Attributes

Attribute	Description
name	Specify Common Name of the device that you want to create, delete, or update
mode	Specify create to create device objects in RMP Specify delete to delete device objects in RMP Specify update to modify one or more attributes in the device objects in RMP
failonerror	Specify true to stop the processing when an error is encountered during any phase of the current operation. Valid values are true and false .
debug	Specify 1 to debug logs for the operation. Valid values are 1 and 0 .

- **Policy:** Policy actions are specified using the `<policy>` element. These actions operate at device level and pertain to a device or group of devices. The following table lists the attributes for `<policy>` element:

Policy Attributes

Attribute	Description
target	Specify device or groupmembers
name	Specify Common Name of the device
mode	Specify assign to assign individual policies to a device or group of devices Specify unassign to remove individual policies from a device or group of

Attribute	Description
	devices
key	Specify instance in the D.C.I Notation (Domain.Class.Instance.) For example, SOFTWARE.ZSERVICE.TEST
prio	Specify priority.Default value is May . Valid values are May , MayNot , Should , ShouldNot , Must , and MustNot .
debug	Specify 1 to debug logs for the operation. Valid values are 1 and 0 .
failonerror	Specify true to stop the processing when an error is encountered during any phase of the current operation. Valid values are true and false .

- **OS:** OS actions are specified using the <os> element. These actions operate at ROM object level and pertain to a device or group of devices. The following table lists the attributes for <os> element:

OS Attributes

Attribute	Description
target	Specify device or groupmembers
name	Specify Common Name of the device
mode	Specify invalidate to configure a device or group of devices so that their Operating System definition is invalidated Specify reevaluate to configure a device or group of devices so that an Operating System connect triggers an Operating System evaluation and deploy an optionally selected Operating System. Specify manage to bring one or more devices or groups of devices under OS Management Specify update to update ROM Object attributes for the device or group of devices
debug	Specify 1 to debug logs for the operation. Valid values are 1 and 0 .
failonerror	Specify true to stop the processing when an error is encountered during any phase of the current operation. Valid values are true and false .

Querying Default Groups

Sentient enables you to query default groups available in **Device Categories** listed under **Management** tab - **Zone:RCA** and perform operations on the devices under these categories. You can perform operations on one or more than one device by setting the attribute `groupype` to value as listed in the following table:

Attribute Values for Querying Device Categories

Default Groups	groupype Value
Device Architecture	model
Device Manufacturers	manufacturer
Infrastructure Services	infrastructure
Managed Services	managementservices
Operating Systems	operatingsystem
OS Management	osmanagement
Subnets	subnet
VM Services	vmsservices

Example: Updating a Device to OS Management Group

To update a device to **OS Management** -> **No Resolved OS** group under **Zone:RCA** - **Device Categories**, add the following lines of code to the customized XML file:

```
<osmkit>

  <os target="groupmembers" groupype="osmanagement"
name="cn=noresolvedos,cn=osm,cn=xref,cn=hp,cn=radia " mode="update"
debug="1" failonerror="true">

    <attributes>

      <curros>_UNMANAGED_OS_</curros>

      <slctdos>_UNMANAGED_OS_</slctdos>

      <rslvdos>_UNMANAGED_OS_</rslvdos>

      <osstate>_DESIRED_</osstate>

    </attributes>

  </os>

</osmkit>
```

You can update devices to other default groups by adding the above line of codes for each group.

Note: When querying the **Device Categories** listed under **Management** tab - **Zone:RCA**, you must provide distinguished name for attribute `name`.

Configuration Examples

The following examples show how to customize your XML configuration file to perform the various supported functions of `osmkit.tkd`.

Example 1: Create a device

```
<device name="AA-1C-C4-18-F0-9B" mode="create" failonerror="true"
debug="1">
  <attributes>
    <nvdhctlana>AA-1C-C4-18-F0-9B</nvdhctlana>
    <ipaddress>4.3.2.1</ipaddress>
  </attributes>
</device>
```

Example 2: Delete a device

```
<device name="AA-1C-C4-18-F0-9B" mode="delete" failonerror="true"/>
```

Example 3: Remove the OS policy assignment for all members of testgroup

```
<policy target="groupmembers" name="testgroup" mode="unassign"
key="OS.ZSERVICE.TEST_SERVICE" prio="May" debug="1"
failonerror="true"/>
```

Example 4: Reset all members of testgroup to neutral “managed” OS state

```
<os target="groupmembers" name="testgroup" mode="update" debug="0"
failonerror="true">

  <attributes>

    <curros>_UNMANAGED_OS_</curros>

    <slctdos>_UNMANAGED_OS_</slctdos>

    <rslvdos>_UNMANAGED_OS_</rslvdos>

    <osstate>_DESIRED_</osstate>

  </attributes>

</os>
```

Disk Encryption

In previous versions of Sentient, a partition that could not be read was determined to contain no meaningful data and would trigger automated disaster recovery.

Sentient can detect when a partition has been encrypted using the following products:

- WinMagic SecureDoc
- PGP Whole Disk Encryption
- Check Point PointSec Full Disk Encryption

- McAfee Safeboot
- Sophos Encryption

Encrypted drive support changes some behaviors of the system:

1. Partition data that cannot be read is assumed to be valid if an encryption product is detected.
2. Automated disaster recovery is not possible using the Behavior setting, Disaster Recovery (PMDISRCV). If you want to perform disaster recovery, you must use the OS Management Wizard with the Emergency Mode option selected in the Sentient Console to reinstall the OS.

Note: After recovering your operating system you must deploy the encryption product components and initiate the encryption process.

3. For kiosk-type machines booting from a CD/DVD, the CD/DVD must be removed following the deployment to prevent the machine being booted from the CD repeatedly.

Prerequisites

Set the BIOS to boot from the local drive first.

Note: Do not capture an image from an encrypted hard drive.

Encryption Support Mode Parameter (ENCMODE)

By default, Sentient will automatically detect the supported encryption products listed above and adjust its behavior to ensure that the system does not perform an unwanted re-installation.

- For network (PXE) boots, the ENCMODE attribute is set to AUTO in the [OS Manager] section of the default file.
- For CD/DVD boots, the ENCMODE attribute is set to AUTO in the [OS Manager] section of `rombl.cfg` which resides in the root of the deployment CD.

You can change how encryption is handled using this ENCMODE parameter.

If ENCMODE is not present, the default value AUTO, is used. To change the value, you may need to add the ENCMODE attribute and the desired value.

The following table describes the values that can be assigned to ENCMODE in the format ENCMODE=*value*.

ENCMODE Attribute Values

Value	Definition
NONE	Do not support encryption. Use this value to enforce the behavior of HPCA 7.2 and below, where a partition that could not be read was determined to contain no meaningful data and treated as an automated disaster recovery situation (depending on the behavior settings).
AUTO (default)	Automatically detect supported encryption products.

Value	Definition
ENC	Assume all partitions are encrypted. Use this for unsupported encryption products because the auto detection feature is not used.

Note: It is recommended that you use a Client Automation service (ZSERVICE) to deploy the encryption product components and initiate the encryption process. It is also recommended that you prioritize the service to ensure that the encryption service is installed first to keep the amount of time the system runs unencrypted to a minimum.

Using Microsoft BitLocker

Microsoft BitLocker encryption technology is significantly different than other 3rd-party encryption products supported by Sentient. BitLocker is an integral part of Vista and newer Microsoft operating systems. It is based on a split partition layout that contains a system partition (typically drive S:) and the operating system partition (drive C:). The system partition is always unencrypted.

When using BitLocker, you must prepare your systems at the partition level so that it is ready to be enabled with BitLocker.

By using Sentient's Reserved Space attribute in the DRIVEMAP class, you can install and prepare systems with the assurance that the Microsoft BitLocker enablement and subsequent encryption will succeed. Next, you must enable BitLocker. See Microsoft's documentation for enablement instructions.

Caution: For Hardware Configuration Operations triggered by policy changes that are sensed during an OS Connect, Sentient will temporarily disable BitLocker. After the Hardware Configuration Operations have been completed, Bitlocker will be re-enabled, ensuring that the preboot integrity trust chain has not been compromised. For Hardware Configuration Operations triggered using the OS Management Wizard with the Emergency Mode option selected in the Sentient Console, you (the administrator) must handle any potential trust chain issues. See the *Sentient Enterprise Console Online Help* for more information about the Repair Device task.

Reserved Space – RSVDSPCE in DRIVEMAP class

The Reserved Space attribute (RSVDSPCE) in the DRIVEMAP class must contain a value expressed in MB.

If you specify this value for its intended use, use a value equal to or greater than 1500. This is the size that Microsoft recommends for the BitLocker S: partition.

A value of 0 (default) will cause Sentient to not leave any gap. Non-fatal warnings will be issued in the OS deployment log when the value is smaller than 1500 and greater than 4000.

When Sentient partitions the disk, it will leave un-partitioned space on the disk equal to the size in MB specified in the RSVDSPCE attribute. This space can then be used later by BDEHDCFG.EXE to prepare the system for BitLocker. This step is not included and must be done separately. Consult the Microsoft documentation for how to enable BitLocker on a deployed system.

The RVSDSPCE attribute is not supported on pre-Vista operating systems. Any value specified will be reset to 0 during deployment, a warning will be issued and no space will be reserved.

Local Service Boot and OSM Client Method Updates

The Local Service Boot service and the OS Manager Application Manager agent have both been updated to recognize and support a BitLocker prepared and/or enabled dual partition scheme.

Partitioning Notes (DRIVEMAP class)

In case of a Merge DRIVEMAP scenario in a BitLocker prepared or encrypted system, the OS Manager service OS agent has been updated to correctly identify both the system and the operating system partition and leave the other partitions intact. When re-creating the OS partition, space will be left unallocated for the system partition. Only the OS partition will be recreated.

The Preserve DRIVEMAP type cannot be used with the BitLocker dual partition scheme.

Introduction to Hardware Configuration Management

Hardware Configuration Management allows you to create and apply operations to hardware on a target device. For example, although some target devices may be ready to have the operating system installed out of the box, there may be other situations when you must identify and apply critical operations before proceeding with the operating system installation. Examples of the types of operations necessary are upgrading the BIOS firmware or configuring a Disk Array Controller (DAC). These operations prepare the hardware configuration on the target device. After the hardware configuration is complete (that is, the hardware configuration is in its desired state), the target device is ready for the operating system to be installed.

Use Hardware Configuration Management where hardware configuration changes are applied to a target device *after* an operating system is installed.

About the Process Flow

Below is an overview of the process flow of Hardware Configuration Management:

1. Create one or more Hardware Configuration Elements (HWCEs)
Each element contains information about the resources required for the operation, sequencing of operations, and how the operation must be carried out.
 - a. Gather the resources for your HWCEs. You may need to go to the vendor's web site to get the correct binaries and create the appropriate configuration files to apply to the target device.
 - b. Use the Sentient Administrator Publisher to publish the HWCEs to the Sentient Enterprise Administrator Configuration Server Database (CSDB). The HWCEs are published to the PRIMARY.OS.PACKAGE class.
 - c. Use the Sentient Administrator CSDB Editor to create and configure the HWCE instances in the PRIMARY.OS.LME class.

- d. Use the Sentient Administrator CSDB Editor to connect the PACKAGE instance to the HWCE instance.
2. Create a Hardware Configuration object
A Hardware Configuration object contains information about configuring the hardware of a target device for operating system installation readiness. It contains one or more HWCEs.
 - a. Use the Sentient Administrator CSDB Editor to create a Hardware Configuration object and specify the filters that identify the devices to apply the hardware configuration.
 - b. Connect the appropriate HWCEs to the Hardware Configuration object.
3. Set policy to ensure deployment of hardware configuration and operating system.
4. The next time the target device connects; it detects the hardware configuration and applies the tasks connected to this hardware configuration.
 - If there is more than one task, they will be applied in the order specified by the administrator.
 - If a task requires a reboot, the Sentient OS Manager will reboot the target device and proceed to the next task.
 - If the operating system must be re-installed after a change, then it is re-installed.

Best Practices

The following are some recommended best practices to follow when using Hardware Configuration Management:

- Deploy in a test environment before deploying Hardware Configuration Management in a production environment.
- Verify that the hardware configuration elements are accurate with respect to scripts that are running in the apply method.

Maintenance

If you receive a fix from CPE, place it in

`\OSManagerServer\OSM\SOS\linux\testing\work` and then reboot the machine.

Perform a One-Time Hardware Maintenance Operation

Using the Sentient Console, you can create a job that uses a Hardware Configuration Element to perform special hardware maintenance operations on a client device. This may be necessary before you can install, update, or repair the OS on certain devices – for example, if you need to trigger a RAID (redundant array of independent disks) verify or re-synch after an active hot spare (AHS) has been changed.

Note: For more routine low-level operations – such as a BIOS firmware upgrade or disk array controller (DAC) configuration – you must use the normal LDS/LME management process.

Performing a One-Time Hardware Maintenance Operation

1. On the **Management** tab, go to the Directories area, and expand the zone that you want to use.
 - To specify one or more individual target devices, click **Devices**.
 - To specify a group, click **Groups**.
2. In the Directory Object table, select the devices (or groups) that you want to work with.
3. In the drop-down menu for one of the selected devices (or groups), select the **Perform a one-time Hardware Maintenance** item in the OS Management submenu. This launches the Hardware Maintenance Wizard.
4. Select **Emergency Mode** if you want to install (or re-install) the OS without attempting to capture and preserve any existing data – for example, in a disaster recovery scenario.
5. Select **Wake on Lan** if you want Sentient to trigger management operations on a machine that is currently turned off.
6. From the Available Maintenance Options list, select the hardware configuration element that you would like to use.
7. Specify the **Start Date** and **Start Time** that this OS deployment job should start.
8. Click **Next**.
The Summary page opens. This page enables you to view all the settings that you have specified for this hardware maintenance job, including the list of target devices.
9. Click **Submit** to create the job.

A new RMP type job should appear under **Current Jobs** on the Management tab (see Managing Jobs in *Sentient Enterprise Console Online Help*).

Using Hardware Configuration Management

In this section, you will learn how to create and apply operations to hardware on a target device, such as a ProLiant server. In this example, you must make the following changes to the hardware configuration before you can install the operating system:

- Update an old version of BIOS firmware (BIOS)
- Configure a disk array controller (DAC)
- Assemble several raw disks into logical volumes (LVOLS)

Caution: If you are using Microsoft's BitLocker, see ["Using Microsoft BitLocker" on page 232](#)

Using Shadow Hardware Configuration Elements (OS.LME)

Gather the Resources for your HWCEs

The table [Hardware Configuration Element Example Definitions](#) defines each of the HWCEs to be used in this example. First, let us review the columns in the table.

- **Hardware Configuration Element** provides a general description for this HWCE. Type a descriptive instance and friendly name for the element in the Sentient Administrator CSDB Editor.
- **Operation** describes the type of operation that will be applied to the target hardware.
- **Resource** lists the files that must be gathered into a single folder. These are the files that performs the operations.

Note: Be sure that text files are in the UNIX® text file format.

- **Apply methods** are the commands that initiate the operations.
- **Post-requisite HWCE** is also known as a consequence. If a HWCE is applied to a target device, some HWCEs may have to be reapplied after the original is applied. For example, if you use an HWCE to change the BIOS firmware, the settings will most likely reset to their default values. Therefore, use a HWCE to reset the default values back to the settings required for your environment. This element will be considered the consequence (or Post-requisite HWCE) for the BIOS firmware HWCE.
- **Conditional Pre-requisite HWCE** is also known as a dependency. Use this field to set up the order in which the HWCEs are applied. This is most effective when you apply your hardware configuration and need to apply the HWCEs in a specific order.

Hardware Configuration Element Example Definitions

HWCE	Operation	Resource Files	Apply Method	Post-requisite HWCE	Conditional Pre-requisite HWCE
Update BIOS	Update firmware	Biosfw.img Biosflash.exe	./biosflash.exe biosfw.img	None	None
Disk Array Controller (DAC)	Configure DAC	Array.conf Acrutil.exe arraycfg.sh	./arraycfg.sh Note: This script contains the command acrutil.exe array.conf	LVOL	None
Logical Volumes (LVOL)	Create logical volumes	Volume.conf Volutil.exe volume.sh	./volume.sh Note: This script contains the command volutil.exe	DRIVE MAP	None

HWCE	Operation	Resource Files	Apply Method	Post-requisite HWCE	Conditional Pre-requisite HWCE
			<code>volume.conf</code>		

The BIOS HWCE updates the firmware on the target device. To perform this operation, the command line listed in the Apply Method column initiates the specified resource files. The BIOS does not have any dependencies or consequences, therefore changes to the BIOS may happen at any time in the sequence of events.

However, the DAC HWCE specifies LVOL as a consequence. This means that the DAC must be configured on the target device before the LVOL and that LVOL must be reapplied after any change to the DAC.

The LVOL HWCE has a consequence of DRIVEMAP. This consequence is a reserved type (see ["About the Hardware Configuration Element Class" on page 243](#) for more information about reserved types) that causes an operating system installation after the OS drive is repartitioned. Therefore, this operation will invalidate any pre-existing operating systems.

When all three operations have been completed successfully, the target device is considered to have its hardware configuration completed and is ready for its operating system to be installed.

Creating and Configuring the HWCEs

In the previous section, the resource packages, that will be downloaded as part of the hardware configuration management actions, are created. The LMEs and/or LDS instances must be created.

In this section, you learn how to configure HWCEs that identify the resources and methods to use to bring the hardware configuration for a target device to its desired state.

This section provides simple step-by-step instructions to gain an understanding about how to create and configure hardware elements. See ["About the Hardware Configuration Element Class" on page 243](#) for detailed information about the fields and the information you provide into the fields.

To configure the Hardware Configuration Element:

1. Go to **Start > All Programs > Sentient Enterprise Administrator > Sentient Enterprise Administrator CSDB Editor**.
2. Log in as the administrator (by default, the user ID is admin and the password is secret).
3. Go to PRIMARY.OS.LME.
4. Right-click the class name and select **New Instance**.

5. Type the display name and the instance name.
When creating the HWCE instances, you may want to use the following Instance and Friendly Names.

For the BIOS, set the:

- Display Name to **Pro32 WS Bios Rev 1.00**
- Instance Name to **P32_BIOS_100**

For the Disk Array Controller, set the:

- Display Name to **Pro32 WS Array Mirrored (1 Spare) Rev 1.00**
- Instance Name to **P32_DAC_MIRR_100**

For the Logical Volume, set the:

- Display Name to **Pro32 WS Array 1 Drive Rev 1.00**
- Instance Name to **P32_LVOL_1DRV_100**

6. Click **OK**.
7. Double-click the instance that you want to configure.
8. Enter the information required to configure the BIOS element as described in the table [Configuring the BIOS HWCE](#).

See "[About the Hardware Configuration Element Class](#)" on page 243 for a complete description of each of the fields below. As identified in this appendix, you *must* type entries for the following fields:

- Friendly Name
- Class
- Type
- Instance Number
- Apply Method
- Apply Behavior
- Post Behavior
- Failure

Configuring the BIOS HWCE

Field	Value
Friendly name	Pro32 WS Bios Rev 1.00
Version	1.00
Class	_REGULAR_
Type	BIOSFLASH
Instance Number	0

Field	Value
Service OS Needed to Run Method	_SVC_LINUX_
Apply Method	./biosflash.exe biosfw.img
Apply Behavior	_NEVER_
Post Behavior	_CONTINUE_
Failure	_CRITICAL_
Post-requisite HWCE	N/A
Conditional Pre-requisite HWCE	N/A

9. Click **OK**.
10. Connect the PACKAGE instance that you created in step b to the PRIMARY.OS.LME instance that you are creating in this step.

Caution: Never connect a PACKAGE instance other than the ones created using the HWCE Publisher. You cannot connect an OS package, a SYSPREP package, a DRIVEMAP, or BEHAVIOR instance, or any other package instance that deals with OS deployment in itself.

11. Repeat the steps above to configure the HWCEs for the Disk Array Controller (DAC) and the Logical Volumes (LVOL). Use tables [Configuring the DAC HWCE](#) and [Configuring the LVOL HWCE](#) as a guide for the information to be entered.

Configuring the DAC HWCE

Field	Value
Friendly name	Pro32 WS Array Mirrored (1 Spare) Rev 1.00
Version	1.00
Class	_REGULAR_
Type	ACRCONFIG
Instance Number	0
Service OS Needed to Run Method	_SVC_LINUX_
Apply Method	/arraycfg.sh
Apply Behavior	_NEVER_
Post Behavior	_CONTINUE_
Failure	_CRITICAL_
Post-requisite HWCE	LDRIVCFG
Conditional Pre-requisite HWCE	N/A

Configuring the LVOL HWCE

Field	Value
Friendly name	Pro32 WS Array 1 Drive Rev 1.00
Version	1.00
Class	_REGULAR_
Type	LDRIVCFG
Instance Number	0
Service OS Needed to Run Method	_SVC_LINUX_
Apply Method	./volume.sh
Apply Behavior	_NEVER_
Post Behavior	_CONTINUE_
Failure	_CRITICAL_
Post-requisite HWCE	DRIVEMAP
Conditional Pre-requisite HWCE	N/A

Creating the Hardware Configuration Object (OS.LDS)

The hardware configuration is the configuration that a target device requires to be in its desired state. In this section, you configure the Hardware Configuration object.

To create the hardware configuration:

1. Go to **Start > All Programs > Sentient Enterprise Administrator > Sentient Enterprise Administrator CSDB Editor**.
2. Log in as the administrator (by default, the user ID is admin and the password is secret).
3. Go to PRIMARY.OS.LDS.
4. Right-click HW Config (LDS) and select **New Instance**.
5. Type the display name, such as `Pro32 WS CAD Setup Rev 1.00` and the Instance name, such as `P32_CAD_100`.
6. Click **OK** and then double-click the instance.
7. Use the MANUFLTR (Reg Expr to Filter Manufacturer) field *and* the MODLFLTR (Reg Expr to Filter Model) field to restrict the hardware that the hardware configuration is applied to.

Caution: You *must* type information in the filter fields as a regular Tcl expression otherwise the hardware configuration will never be applied to your target device. Filter entries are case-sensitive. If you need assistance with regular Tcl expressions, refer to a manual such as *Practical Programming in Tcl and Tk* by Brent B. Welch, Prentice Hall

PTR (3rd edition or newer).

- a. Double-click the MANUFLTR (Reg Expr to Filter Manufacturer) attribute and type a regular Tcl expression to indicate the manufacturer that you want to apply this hardware configuration to. If you do not provide information in this field, no match will be made.
 - If you type a value `.*`, the hardware configuration will be applied to any target device.
This is not recommended because your hardware configuration may cause damage to some of your target devices.
 - List the items as a series of regular expressions separated by a space.

Some example values for the manufacturing filter might be:

`HP.* Compaq.* COMPAQ.*`

or

`Dell.* DELL.*`

- b. In the MODFLTR (Reg Expr to Filter Model) attribute, type an expression to indicate the model that you want to apply this hardware configuration to.
 - If you type a value `.*`, the hardware configuration will be applied to any target device.
This is not recommended because the hardware configuration targeted at a particular manufacturer may cause damage to some of your target devices. It is important to consider how to use your filters and be sure to filter based on both manufacturer and model.

Some example values for the model filter might be:

`DL360-G2.* DL360.* PROLIANT.*DL360.*`

or

`Poweredge.* .*2450.*`

These patterns match the models of DL360 servers or Poweredge 2450 server models.

Caution: Be careful not to type an expression such as `DL360-C2*` because in this case, the asterisk is not used as a wildcard as you would expect. The asterisk causes a match to any model that has repeated 2s in it. For example, some potential matches would be `DL360-C22`, `DL360-C222`, `DL360-C2222222222`, and so on.

Caution: The manufacturer and model filters must match the hardware configuration applied to the target device.

8. To track the versions of your configuration elements, use the VERSION attribute.
9. Click **OK**.

To connect the HWCEs to the Hardware Configuration Package:

1. Go to **Start > All Programs > Sentient Enterprise Administrator > Sentient Enterprise Administrator CSDB Editor**.
2. Log in as the administrator (by default, the user ID is admin and the password is secret).

3. Go to PRIMARY.OS.LME and drag and drop the instances to the HW Config (LDS) instance. In this example, use the three elements created above.

Viewing the Assigned Hardware Configuration Objects

After an Sentient OS connect has occurred and the hardware configuration has been deployed to a target device, you can review the Assigned Hardware Configuration Objects that are successfully applied. To do this, go to the Device Properties sheet in the Sentient console.

Modifying the Hardware Configuration on a Device

Once the hardware configuration is in its desired state, it may remain in that state for long. For example, you may want to update your BIOS with the latest version of firmware. To do this, perform the following:

1. Publish the resources for the updated firmware. For example, publish the resources with the description Pro32 WS Bios Rev 2.00 Resources and the instance name P32_BIOS_200.
2. Use the CSDB Editor to make any necessary modifications to the HWCE for the updated firmware. See *To configure the Hardware Configuration Element in " Using Shadow Hardware Configuration Elements (OS.LME)" on page 235* for details.
3. Create a new Hardware Configuration object, such as Pro32 WS CAD Setup Rev 1.10 (instance name P32_CAD_110), and connect the following HWCEs to this object.
 - Pro32 WS Bios Rev 2.00
 - Pro32 WS Array 1 Drive Rev 1.00
 - Pro32 WS Array Mirrored (1 Spare) Rev 1.00
4. Change policy to deploy the new Hardware Configuration object Pro32 WS CAD Setup Rev 1.10.
5. At subsequent Sentient OS connect, it detects the hardware configuration and applies the tasks connected to this hardware configuration.
6. In the Sentient console, go to the Device Properties sheet for the target device to see the Current Hardware Configuration. The new Hardware Configuration object, P32_CAD_110, will be listed. Access the ROM object to see the Current HWCEs. In this example, the elements, P32_DAC_MIRR_100 and P32_LVOL_1DRV_100 remain unchanged and only the new BIOS, P32_BIOS_200 is applied to reach the new desired state.

If there were no consequences or dependencies, the elements DAC and LVOL can be considered as not modified. However, if there was some consequence, then one or both of the elements may be reapplied to the device.

Publishing Hardware Configuration Elements

In this section, you will use the Sentient Enterprise Administrator Publisher to publish the HWCEs to the Sentient Enterprise Configuration Server Database.

Note: Before you publish your HWCEs, gather your resource files into a single folder.

To publish an HWCE:

1. Go to **Start > All Programs > Sentient Enterprise Administrator > Sentient Enterprise Administrator Publisher**.
2. Type your User ID and Password.
3. From the Publishing Options drop-down menu, select **HW Configuration**.
4. Click **OK**.
5. Select the folder that contains the resources required to create your HWCE. In the example, `C:\HWCEs\BIOS` is selected.

Caution: Make sure that you gathered the correct files that match the system to which you intend to deploy this. If you choose the wrong files, you may leave your system in a damaged state. Also, all the files to be published must be located under `C:\HWCEs\BIOS`. The Publisher does not look into subdirectories of `C:\HWCEs\BIOS`.

6. In the Description field, type a description of the elements being published. For this example, type `Pro32 WS Bios Rev 1.00 Resources`.
7. In the Package Instance Name field, type the instance name for the package. For this example, type `P32_BIOS_100`.
8. Click **Next**.
9. Review the information and then click **Publish**. The package resources are published in a non-compressed format.
10. When the Publisher is done, click **Finish**.
11. Click **Yes** to confirm to close the Publisher.

Use the CSDB Editor to view the package that has been created in PRIMARY.OS.PACKAGE.

Repeat the above steps to create packages for the Disk Array Controller (DAC) and the Logical Volumes (LVOL). Ensure to use descriptive instance names and provide detailed descriptions. For example:

For the Disk Array Controller, set the:

- Description to `Pro32 WS Array Mirrored (1 Spare) Rev 1.00 Resources`
- Instance Name to `P32_DAC_MIRR_100`

For the Logical Volume, set the:

- Description to `Pro32 WS Array 1 Drive Rev 1.00 Resources`
- Instance Name to `P32_LVOL_1DRV_100`

About the Hardware Configuration Element Class

This section provides detailed information about the attributes used to create Hardware Configuration Elements (HWCEs).

Hardware Configuration Element Class

Name	Attribute Description	Default Value	Potential Values
Name (required)	Friendly name	N/A	N/A
Version	User-defined version number. For reference only.	1.00	N/A
Class (required)	Identifies if the HWCE is a Regular HWCE or a Shadow HWCE.	_REGULAR_	_REGULAR_ A regular HWCE is an object that identifies the resources and methods required to bring a target device to a state where it is ready to have an operating system installed. _SHADOW_ A Shadow HWCE is an exception to how the state of a device is typically managed. A Shadow HWCE is applied to your target device one time and is then deleted.
Type (required)	User-defined identifier A hardware configuration typically has only one HWCE of this type. If there is more than one HWCE of the same type, then only the first encountered one is applied. If a system has identical hardware components that a HWCE can be applied to, then set more than one HWCE to the same type and set the INSTANCE attribute to 0. See " Instance Number (0 - > apply to all) (required)" on page 246 and " About the Instance Number and Apply Method Attributes" on page 249.	N/A	Examples of user-defined types BIOSFLASH Flashes the BIOS of a device. NVRCONFIG Configures the non-volatile RAM of a system. The purpose of this is to batch configure a system's parameters such as bootpath, PXE enable, OS type, and so on. There are many different parameters that can be set on a ProLiant server to customize its configuration. ACRFLASH Flashes the firmware of an array controller.

Name	Attribute Description	Default Value	Potential Values
			ACRCONFIG Configures the array controller with respect to RAID volumes that are built from raw local disks or raw fiber channel LUNs (==FC disk drives). LDRVCONFIG Creates the logical drives (for example, C:) from either raw local disks or by carving off a part of a RAID volume that has been set up. NICFLASH Flashes the firmware of network interface controllers (NIC). NICCONFIG Configures any programmable elements of NICs. HBAFLASH Flashes the firmware of a SCSI or fiber channel host bus adapter (HBA) (disk controller). HBACONFIG Configures SCSI or fiber channel host bus adapter with regard to getting raw disks or FC LUNs (disks) assigned. READ_CONFIG Used for pre-configured HWCEs that read or capture the array/hardware/iLO/system configuration of the target device into a file that is uploaded to the OS Manager.

Name	Attribute Description	Default Value	Potential Values
			Reserved types <i>Do not use the OS or DRIVEMAP types for your HWCEs because they are reserved for internal use only.</i> OS Causes an OS installation DRIVEMAP Causes an OS installation
Instance Number (0 -> apply to all) (required)	A unique number used to establish a convention between the apply methods and the application of the apply methods to a set of HWCEs. Information in this field does not do anything until the person writing the apply method scripts uses it to establish conventions. See " About the Instance Number and Apply Method Attributes" on page 249 for more information.	0	0 or greater depending on the apply method scripts written. Typically, the value of 0 is used to indicate that a HWCE is to be applied to <i>all identical components</i> of a device. If you do this, the apply method script must interpret 0 as an indication to apply the script to all instances of the hardware configuration elements. If you set this to 1, the apply method may be applied to only one hardware configuration element. Again, the apply method script must support this. You can use values greater than 0 to indicate that the hardware configuration element must be applied to one or more identical pieces of hardware. See " About the Instance Number and Apply Method Attributes" on page 249 for more information.
Service OS Needed to Run Method	System defined service OS identifier	_SVC_LINUX_	_SVC_LINUX_ to run the Linux Service Operating System. _SVC_PEX86 to run the

Name	Attribute Description	Default Value	Potential Values
			WinPE 2.0 Service Operating System. Note: Set the Service OS for each HWCE to ensure it runs in the required Service Operating System.
Apply Method (required)	A command line that initiates the HWCE. Note that the text files must be in the UNIX text file format. Also, note that if you have an apply method that reboots the machine and you want messages to be seen by the user, you must redirect any output to /dev/console. For example, if your apply method flashes BIOS and you want the user to see a message indicating that the BIOS is being flashed and not to touch the power button, redirecting the output will ensure that the user sees the message before the reboot. See " About the Instance Number and Apply Method Attributes" on page 249 for more information.	N/A	N/A
Apply Behavior (required)	Defines the reboot behavior on completion of the HWCE apply method. Note that apply behavior must be in sync with the apply method otherwise you may run into problems deploying the hardware configuration. An example	_NEVER_	_ALWAYS_ The apply method always triggers a reboot after the operation without returning control to the Sentient OS Manager User Agent. _OK_ After success, the apply method triggers a reboot

Name	Attribute Description	Default Value	Potential Values
	of an inconsistency is if your apply method includes a reboot command, but the Apply Behavior is set to <code>_NEVER_</code> .		without returning control to the Sentient OS Manager User Agent only. <code>_NEVER_</code> The apply method never reboots and always returns control to the Sentient OS Manager User Agent. <code>_ERROR_</code> <i>For future use.</i>
Post Behavior (required)	After a HWCE is processed successfully, the machine is instructed being configured the next steps regarding its boot state.	<code>_CONTINUE_</code>	<code>_REBOOT_</code> On successful completion of an apply behavior operation, a reboot of system is triggered. <code>_CONTINUE_</code> On successful completion of an apply behavior operation, the system does not reboot. It continues onto the next HWCE or OS installation operation after HWCE processing returns.
Failure (required)	The command or shell script returns an error and this indicates how to handle that error.	<code>_CRITICAL_</code>	<code>_CRITICAL_</code> Any failure of this HWCE operation is considered critical and terminates HWCE /OS install processing. <code>_IGNORE_</code> Any failure of this HWCE operation is considered a success for the purposes of HWCE processing.
Post-requisite HWCE	The HWCEs listed here must be applied after HWCE is configured. These may also be referred to as consequences.	N/A	Type a friendly name, Type, or a combination of Type_ Instance Number. A space is used as the separator between multiple items.
Conditional Pre-requisite HWCE	If there are HWCEs that must be applied, they must be applied before the	N/A	Type a friendly name, Type, or a combination of Type_ Instance Number.

Name	Attribute Description	Default Value	Potential Values
	HWCE is configured. These may also be referred to as dependencies.		A space is used as the separator between multiple items. For example, the following may be a list of dependencies: BIOSFLASH SMART5ICFG NVRAMFLASH_1 BIOSFLASH refers to the TYPE. SMART5ICFG refers to a specific HWCE. NVRAMFLASH_1 refers to the INSTANCE.
Connections	Connections to published packages of resources	N/A	N/A

About the Instance Number and Apply Method Attributes

This section provides information about the relationship between the instance number and the apply method.

The instance number is a unique, user-defined number used to establish a convention between an apply method (for an HWCE) and identical hardware in the target system. Information in this field does not *do* anything unless the user writing the apply method script uses it to establish conventions. For example, if you determine that setting the instance number to 0 means that the HWCE must be applied to all identical hardware in the target system, the user writing the apply method must establish this convention in the script. Also, the user creating the HWCEs must define what is considered "identical hardware" based on the service OS, hardware configuration, and apply method.

Typically, instance values greater than zero designate that the hardware configuration element must be applied to one or more identical instances of a system's hardware components. The specific instances are defined by the script writer based on the service OS, hardware configuration, and apply method. For example, the apply method for a hardware configuration element with an instance number set to 1 may be intended to be applied to the first piece of matching hardware detected during an operating system boot. Or, it could mean that the apply method for the hardware configuration element with an instance number set to 2 may be intended to be applied to the second piece of matching hardware detected during an operating system boot. An instance number of 2 could also mean that this operation must be applied to all even numbered disks detected during operating system boot.

Remember, the value of the instance number is meaningless unless there is a convention between the Sentient use of the value and the apply method scripts that determine what is to be done with it.

Below are two examples of potential usage of these attributes.

Example 1

A system contains three identical disks that have been assigned the same HWCE. The instance number attribute in the HWCE is set to 1.

In this example, the apply method was written so that:

- An instance number set to 1 indicates that the operations must only be applied to one disk in the system. Whether the apply method is applied to disk 1, 2, or 3 depends on how the apply method was written.
- An instance number set to 0 indicates that the apply method is to apply the appropriate operations to each disk assigned to this hardware configuration element.

To do this, the apply method script contains a loop that applies the method to all three disks. This loop ensures that the operation is applied to any number of disks that are assigned to this hardware configuration element.

If the same system contained 10 disks³/₄five of type A and five of type B³/₄you might want to apply the hardware configuration element only to type A disks. If this was the case, the script could be written so that if the instance number is set to 0, the HWCE would only be applied to type A disks.

Example 2

Two network interface cards (NICs) are eligible to PXE network boot but only one may be started as PXE bootable at a time.

Two different Hardware Configuration Elements (HWCE) can be created with the type set to NICCONFIG. The first HWCE enables the PXE boot and the second HWCE disables the PXE boot. All other parameters for the HWCEs are set identically.

In this case, you would not want to set the instance number to 0 for the HWCE that enables the PXE boot. If your script indicates the operation is applied to all HWCEs by setting the instance number to 0, then all NICs are enabled causing issues.

However, you can set the instance number to 0 for the HWCE that disables the PXE boot. If your script indicates that the operation is applied to all HWCEs by setting the instance number to 0, then all NICs are disabled making it useful.

Also, because you do not want the enable PXE and disable PXE HWCEs to overlap, you can create the apply method script to use an instance number set to 1 to designate the first NIC and use an instance number set to 2 to designate the second NIC.

About the Hardware Configuration Class

This appendix provides detailed information about the attributes used to create Hardware Configuration objects.

The Hardware Configuration Class

Name	Attribute Description		
Stop Expressions	For internal use only	N/A	N/A

Name	Attribute Description		
Connections	Allows you to manually create connections to hardware configuration elements	N/A	N/A
Name	Name of the Hardware Configuration	N/A	N/A
ZSVCPRI	Service Create Ordering For internal use only	N/A	N/A
Version	User-defined version number. For reference only.	1.00	N/A
MANUFLTR	Reg Expr to Filter Manufacturer Restricts the hardware that the hardware configuration is applied to, based on manufacturer Provide information in the filter fields as a regular Tcl expression otherwise the hardware configuration does not apply to your target device It is recommended that you create a thorough plan for the proper filters See "Creating the Hardware Configuration Object (OS.LDS)" on page 240 for detailed information	""	Any regular Tcl expression
MODLFLTR	Reg Expr to Filter Model Restricts the hardware that the hardware configuration is applied to, based on model Provide information in the filter fields as a regular Tcl expression otherwise the hardware configuration does not apply to your target device It is recommended that you create a thorough plan for the proper filters See "Creating the Hardware Configuration Object (OS.LDS)" on page 240 for detailed information	""	Any regular Tcl expression

Multicast and OS Management

Sentient supports reliable delivery multicast so that you can rollout large numbers of OS images concurrently with improved performance.

In general, the same concepts apply when using the Multicast Server for the Application Manager or for OS management.

This topic covers how to use multicast with Sentient OS management.

Prerequisites

- An understanding of the Multicast Server.
- A basic understanding of OS management in Sentient.

Requirements

- The image will be downloaded only if the Service Multicast Eligible option is selected for the OS Service. To do this, use the Portal to navigate to the appropriate Operating System service.
 - a. Click **Modify Instance**.
 - b. In the workspace, click **Advanced**.
 - c. Scroll to the bottom of the screen and make sure that Service Multicast Eligible is selected.

Configuring Multicast for OS Management

To configure reliable delivery multicast:

1. Go to the appropriate BEHAVIOR instance.
2. In the workspace, click **Advanced**.
3. Click **Modify Instance**.
4. Modify the ROMAPARM Parameters field as follows:
`-multicast multicastIPAddress:3466 -mcastretrycount 1-
mcastretrywait 240`

Description of ROMA Parameters

Parameter	Description
multicastIPAddress	This parameter specifies the Multicast Server host. You can also use the host name. 3466 is the default Multicast Server port.
mcastretrycount	This parameter specifies the number of times that the client will retry multicast if there is a failure. The default value is 1.
mcastretrywait	This parameter specifies how long to wait before the client will start the retry. The default value is 240 seconds.

5. Modify the following file as needed:
`<InstallDir>\MulticastServer\etc\mcaster.cfg`
 - `root`
Specifies the root directory from which the Multicast Server will retrieve resources.
 - `address`
Specifies a range of multicast IP addresses available for use with dynamic windows.
 - `Minref`
Specifies the minimum number of clients that are required to contact the multicast server

to start a multicast session. By default, `minref=2`. You may want to change this to take advantage of multicast's functionality. You may want to set `minref=1` for debugging purposes.

■ CWINDOW

Specifies the length of the collection window; how long to wait for clients to register for a given OS service before finalizing the setup of a multicast session. Change the value for this parameter based on your requirements.

See "[Implementation and Diagnostics](#)" on page 1454 for more information about the parameters in this file.

6. If you made changes to `mcast.cfg`, restart the Multicast Service to implement your changes.

Caution: You may notice a `multicast.rc` file in this folder:

`<InstallDir>\MulticastServer\etc`

Do *not* make any changes to this file.

Improving Performance and Reliability for Multicast with OS Management

The default values of the multicast parameters provide a good combination of reliability and performance in many environments. Optimal performance (transfer speed) is relative to your network environment. Therefore, you must determine what is optimal for your environment and then use the parameters defined in this topic to increase reliability and performance.

The fundamental problem surrounding the reliability and performance issues of the multicast transfer is packet loss. Because multicast is a UDP based protocol, delivery of packets is not guaranteed.

External factors that contribute to packet loss are:

- Network conditions. The amount of traffic on the network, the number of routers between the server and client, and faulty network connections, all can contribute to packet loss during multicast transfers.
- Agent conditions. The relative CPU, I/O and network performance of the agents can contribute to packet loss specific to the clients in question. If an agent is unable to read packets fast enough, some of those packets will be missed.

In any environment, packet loss is inevitable. The key is to find the balance between minimal packet loss and high data transfer rates in order to optimize actual throughput.

Terminology

It is important to understand of how multicast handles the transfer of images. A sender (server) sends packets to a receiver (agent). The agent receives the data. If the data has not been received in its complete form, the client sends a resend request to the server. The server resends the packets to attempt to complete the transfer successfully. Below you will be introduced to some of the terminology that you will see used throughout this topic.

actual throughput

The size of the operating system image divided by the time it takes to transfer the image.

agent (receiver)

The agent that receives the multicast transmission.

image

The data that is transmitted from the server to its clients in a single multicast session. For OS management, this is an operating system image.

multicast transfer

The process of sending data from the server to the client.

packet

A unit of information sent over a computer network.

packet loss

When the agent does not receive one or more packets sent by the server.

performance

The time it takes to transfer the image.

raw data transfer rate

The total number of packets (fixed size of data) sent over time, including packets that have been resent.

reliability

The likelihood that the multicast transfer will complete successfully.

resend block

A group of packets to be resent as a result of a resend request (NACK).

resend request/negative acknowledgment (NACK)

A message sent from the client to the server indicating the client did not receive a specific piece of data .

server (sender)

The agent that transmits the data to its clients via multicast. For OS management, this data is an operating system image.

About the Multicast Parameters

This section describes the multicast parameters whose values may need to be modified in order to increase performance and/or reliability.

Multicast Parameters

Parameter	Used By	Definition	Default Value
<code>gddelaybp</code>	Sender	Inter-packet delay. The number of milliseconds to wait after sending a packet before sending the next one.	0.0625
<code>lingercount</code>	Sender	The number of times to check for resend requests (NACKs) after the last packet has been sent before determining that the transfer is complete.	512
<code>lingerdelay</code>	Sender	The delay, in milliseconds, between checking for resend requests (NACKs) after the last packet has been sent.	32.0
<code>lprcount</code>	Sender	The number of times the last packet of the image is retransmitted in order to increase the probability that the receiver sees the last packet. Note that the receiver recognizes the last packet because it contains a flag indicating that it is the last packet.	4
<code>lprdelay</code>	Sender	The delay, in milliseconds, between each attempt to resend the last packet.	.25
<code>maxrsndreq</code>	Receiver	The maximum number of resend requests (NACKs) that can be issued for a given block. A block contains a number of packets. The size of a block is defined by the <code>numpktblks</code> parameter described below.	4098
<code>nacdelay</code>	Receiver	The delay, in milliseconds, between resends of a specific NACK.	0.5
<code>nacresend</code>	Receiver	The number of times to resend each NACK.	2
<code>netinactto</code>	Receiver	Network inactivity time-out. The number of minutes of network inactivity allowed between received packets before the receiver fails.	5
<code>numpktblks</code>	Sender or Receiver	Defines the size of the pool from which resend requests are fulfilled.	64
<code>pktsperblk</code>	Sender or Receiver	Specifies the number of packets within a resend block. This is the minimum number of packets that will be resent as a result of a NACK. The total number of these packets is considered a resend block. This value must be a multiple of 32. If you do not follow this requirement, your value will be adjusted and noted in the <code>gdmcsend.log</code> and the OS	256

Parameter	Used By	Definition	Default Value
		Manager System Agent logs.	
recvtimeout	Receiver	The maximum time, in minutes, that is allowed for the total data transfer before it is considered a failed transfer.	45
throtfreq	Sender	Throttle frequency. Specifies how often to check to see if the inter-packet delay should be adjusted.	8
throthighth	Sender	Throttle high threshold. The number of average resends per block that will trigger an increment of the inter-packet delay.	-1 (disabled) Note: To enable this, set it to a positive integer.
throtincr	Sender	Throttle increment. The value, in milliseconds, that is automatically added to (or subtracted from) the current inter-packet delay each time the throttle is adjusted. See "Auto Throttle" on page 259 for more information.	0.01
throtlowth	Sender	Throttle low threshold. The number of average resends per block that will trigger a decrement of the inter-packet delay.	-1 (disabled) Note: To enable this, set it to a positive integer.
throtmax	Sender	Throttle maximum. The maximum inter-packet delay, in milliseconds, that can be set by the throttle.	0.5
throtmin	Sender	Throttle minimum. The minimum inter-packet delay, in milliseconds, that can be set by the throttle.	0.0
ttl	Sender	Time to live. The number of subnets that the packet will reach. Every time a packet reaches a switch the ttl value is decremented until it reaches 0. If the value is 0, the packet cannot cross the switch. This limits how far the packets can spread from the sender.	3

How the Parameters Influence Multicast Data Transfer

This section provides a more in-depth description of the parameters, including the influence they have on the multicast data transfer and their interaction with each other.

Understanding Inter-packet Delay

The raw data transfer rate of the sender is influenced by the inter-packet delay parameter (`gddelaybp`).

Note: `Gddelaybp` represents the number of milliseconds to wait after sending a packet before sending the next.

Increasing the inter-packet delay will decrease the raw data transfer rate of the sender. In general lower transfer rates will result in less packet loss. If the transfer rate is too low, it will have a negative impact on the actual throughput.

To give you a feeling for the impact this parameter can have on the actual throughput, consider the example of transferring a one gigabyte image using a 1 millisecond inter-packet delay. One gigabyte is 1,073,741,824 bytes. Assuming each packet is 1024 bytes, the image can be transferred in 1,048,576 packets at best. Given a one millisecond delay for each packet, the delays alone would total more than 1048 seconds. This means that it would take over 17 minutes to transfer the image, assuming no packet loss at all. In actuality, some packets probably will be lost, requiring some of the data to be resent; each resend packet consuming at least one millisecond.

Approaching this from the other direction, to transfer one gigabyte image in under five minutes. Five minutes equals 300,000 milliseconds. Dividing that by 1,048,576 packets gives us about 0.3 milliseconds per packet. So, before transferring the image in under five minutes, the inter-packet delay must be less than 0.3. Unfortunately, lowering this value will more than likely result in greater packet loss and in turn, more resent packets.

To what degree lowering the inter-packet delay results in greater packet loss depends on the network and client conditions. While some conditions may support very low inter-packet delay values with minimal packet loss, others may not. Normally, when the conditions cannot support a given raw data transfer rate, the actual throughput will suffer due to the number of resends required to complete the transfer. In extreme cases however, the transfer may fail.

About the Buffer Settings

While the buffer settings do not have an impact on the raw data transfer rate, they can have significant impact on the reliability and actual throughput of the transfer.

The buffer, as defined by the `numpktblks` and `pktsperblk` parameters, influences the following characteristics of the multicast transfer:

- The maximum number of packets the receiver can handle before it has the opportunity to write out the packets received first. For slower clients, there may be periods during the transfer where packets are being received faster than they can be written out, or an unfulfilled resend

request may prevent a buffer from being written out, causing received packets to backup. During these periods, the overall size of the buffer (`numpktblks * pktsperblk`) defines the number of packets that can be received before the backup is alleviated. If the buffer limit is exceeded before the backup is alleviated, the transfer will fail.

- On the sender side, the number of packet blocks (`numpktblks`) defines the size of the pool from which resend requests are fulfilled. If a resend request is made for a block that is no longer in this pool, the server will not be able to fulfill the request.
- On the receiver side, the number of packet blocks, `numpktblks`, defines the size of the pool of blocks for which resend requests can be made.
- The size of each packet block (`pktsperblk`) defines the minimum number of packets that will be resent as a result of a resend request (NACK). The optimum packet block size depends on the overall distribution of lost packets. If lost packets are few and far between, then smaller packet blocks will minimize the overhead associated with the acquisition of each lost packet. If lost packets tend to be grouped together, then larger packet blocks may minimize the number of resend requests (NACKs) required to acquire the missing packets.

Handling Special Packets

Multicast, being a UDP based protocol, does not guarantee delivery of packets. The protocol used to send resend requests from the receivers to the sender is based on UDP as well, so delivery of resend requests is not guaranteed. However, relying on the resend requests ensures the delivery of the packets. In addition, the last packet sent from the sender is used to trigger resend requests from the receiver as needed. If the last packet is lost, receivers will not know to request resends for the missing packets, including the last one.

As a resend request cannot be relied upon to ensure that a resend request is received, it is required to fall back on a more fundamental way to minimize the probability that these special packets will be lost. To do this, a fixed number of duplicates for each of these types of packets is sent to ensure that at least one of them will be received by the clients. The parameters used to do this are:

- `nackresend` defines the number of times each NACK packet is retransmitted.
- `nackdelay` defines the delay between each retransmission.
- `lprcount` defines the number of times the last packet of the image is re-transmitted.
- `lprdelay` the delay between each retransmission.

The more clients participating in the multicast session, the lower the need for many NACK resends. Assuming many of the lost packets will be common to a large number of receivers, more often than not, multiple receivers will NACK the same blocks.

Handling the End of Image

After the multicast server has sent the last packet of the image, it needs to wait to see if there are any remaining NACKs that need to be serviced before exiting. The `lingercount` and `lingerdelay` parameters govern how this is done.

Note: `Lingercount` - The number of times to check for resend requests (NACKs) after the last packet has been sent before determining that the transfer is complete.
`Lingerdelay` - The delay, in milliseconds, between checking for resend requests (NACKs) after the last packet has been sent.

Basically, the server checks for NACKs *lingercount* times and waits *lingerdelay* milliseconds between each check. If the server does not see a NACK in that period, it exits. If it does receive NACKs, it services them and starts checking all over again.

If these parameters are set too low, the server may exit before it receives the remaining NACKs from its clients. If this happens, the transfer to the clients with unfulfilled NACKs will fail. In the event of failure, the transfer will be retried if you have set `mcastretrycount` to a value greater than 0.

Auto Throttle

The intent of this feature is to prevent adverse network and/or client conditions from causing the actual throughput from degrading to unacceptable levels, not to optimize throughput; although, in some cases, it may accomplish just that.

This feature attempts to keep the average NACKs per block within a predefined band. This is accomplished by modifying the inter-packet delay (`gddelaybp`) whenever the average NACKs per block falls outside the band. The band is defined by high (`throthighth`) and low (`throtlowth`) throttle threshold values, where the high threshold is the maximum desired NACKs per block and the low threshold the minimum.

After each packet block is sent for the first time, the n-moving average for the last *n* packet blocks is computed, where *n* is the number of packet blocks currently configured (`numpktblks`). When the throttle is checked, this moving average is compared to the high and low throttle thresholds, and the inter-packet delay is adjusted accordingly. If the moving average is greater than the high throttle threshold, a configurable value (`throtincr`) is added to the inter-packet delay. If the moving average is less than the low throttle threshold, the same configurable value is subtracted from the inter-packet delay. High (`throtmax`) and low (`throtmin`) limits for the inter-packet delay are also defined. If a throttle adjustment would cause the inter-packet delay to exceed either of these limits, the adjustment will not be made.

The throttle is checked after every `throtfreqpacket` blocks are sent. Here, `throtfreq` is the configurable throttle frequency. Actually, this is the throttle period, as it defines the number of packet blocks between throttle adjustments. The intent here is to give any previous adjustments an opportunity to influence the results, before checking the throttle again.

Analyzing Problems

This section describes how to identify, analyze and resolve multicast data transfer problems.

About the Logs

The sender's log file, `gdmcsend.log`, is typically stored here:

`<InstallDir>\MulticastServer\logs`

The receiver log is typically appended to the end of the OS Manager System Agent log for the device. Is this what this agent is currently called?

Poor Performance

As mentioned before, poor multicast transfer performance is usually due to poor network and/or agent conditions. Such conditions result in the generation of an excessive number of resend requests (NACKs) from one or more of the clients, slowing down the entire transfer.

Before you can resolve the performance issue, you must first determine the root cause of the problem. To do so, examine the contents of the multicast sender's log file, `gdmcsend.log`. Review the following steps to guide you in determining the cause of the problem.

1. Determine is the average number of resends per block for the transfer in question. Look for the line in the log file in the form:

```
Avg resends per block = 0.00283688
```

Averages less than one are very good. This indicates that most of the packet blocks were sent only one time, with relatively few resends. Large values may indicate a problem. What to consider large depends on the value of the inter-packet delay, `gddelaybp`. Remember, there is a trade-off between raw data transfer rates and packet loss, so you can expect more NACKs when the inter-packet delay is small.

2. If the average resends per block indicates that there is a problem, examine the per-client statistics for the transfer. In the same log file, look for lines in the form:

```
Client stats:
```

```
Client: 16.119.237.171 (0xabed7710) NACKs = 19714
```

```
Client: 16.119.237.207 (0xabed7710) NACKs = 102
```

```
Client: 16.119.237.122 (0xabed7710) NACKs = 17
```

```
Client: 16.119.237.217 (0xabed7710) NACKs = 8
```

Each client is identified by its IP address. The client that has been issued the most resend requests (NACKs) appears at the top of the list.

If there are one or more agents that top the list whose NACK count far exceed those of the other agents, it is a strong indication that the problem is specific to the agents in question. After the problematic agents have been identified, you can try to determine what sets them apart from the others. Some considerations:

- a. Are the problematic clients on a different subnet than the others? If so, the problem may be specific to that subnet. Check the routers in the path from the server to the clients to see if any have seen a large number of errors on any of their ports. If so, it can be a router, port, or cabling problem.
- b. Are the agents in question slower than the others? Slow clients may be unable to keep up with high raw data transfer rates, causing them to miss more packets and in turn, NACK more often. If this is the case, you have a few options:
 - Increase the inter-packet delay (`gddelaybp`) in order to lower the raw data transfer rate, so the slower agents will be better able to keep up. Even with the lower transfer rate, if the number of NACKs from these agents is significantly reduced, the actual throughput may increase.
 - Whenever possible, do not include these clients in multicast sessions with faster agents. Put them in their own multicast session, or use unicast to deploy images to them.

- c. If the clients are of comparable speed, the local network connections or cabling may be at fault. Check the cables and connections closest to the agents to see if they are causing the problem.
3. If all of the clients show a large number of NACKs, the problem is probably more systemic.
 - a. The network may have been especially congested during the time of the transfer. Performing the transfer when the network is less busy may yield better results.
 - b. Check the relevant network routers, connections and cabling as described above. This time, make sure to check the cables and connections from the server to the network.
 - c. It could be that all of the machines are just too slow to keep up with the current raw data transfer rate. Increase the inter-packet delay to see if fixes the problem.

In some cases, enabling the auto-throttle feature is a better alternative than manually increasing the inter-packet delay. After the proper threshold values are set, the auto-throttle will adjust the inter-packet delay as needed.

Client Time-Out

Agents can time out for one of two reasons:

- **Total image transfer time-out** occurs when the total time it takes to transfer the image exceeds the value of the `recvtimeout` parameter.
- **Network inactivity time-out** occurs when the time between received packets exceeds the value of the `netinact` parameter.

When a client times out, the type of time-out can be determined by examining the client's log file.

Total Image Transfer Time-Out

In the log file, a total image transfer time-out is indicated by a message in the form:

```
Module has timed out (timeout = nnn)
```

where *nnn* is the time-out value that has been exceeded.

Extreme cases of poor performance can lead to this type of failure, when the performance degrades to the point where the image cannot be transferred in the time defined by the `recvtimeout` parameter. When this is the case, the same techniques described in "[Poor Performance](#)" on the previous page, can be used to identify and resolve the problem.

Network Inactivity Time-Out

A log file message in the following form is indicative of a network inactivity time-out:

```
Inactivity timeout has been exceeded.
```

This type of failure can be caused by almost anything that disrupts the flow of data from the server to the client. Premature termination of the multicast sender and various network problems can occasionally be at fault.

In some cases, it can result from the loss of one or more strategic packets. For example, the client in question may not have seen the last packet of the image. If this is the case, it will not know it needs to NACK the missing data. Having sent the last block and not seeing any NACKs, the server will not send more data. Expecting more data, the client will wait for the next packet until `netinact` has been exceeded.

It can be determined if the client missed the last packet of the image by examining the log files. In the sender's log file, `gdmcsend.log`, look for two lines in the form:

```
Last block: 3524
```

```
Packets in last block: 54
```

If they exist, then you know the sender sent the last packet.

Now, in the client's log file, look for a line like:

```
Last buffer size = nnn
```

If this line is not there, then you know the client did not see the last packet.

To remedy this problem, increase the value of the `lprcount` parameter. This will cause the last packet of the image to be retransmitted more times, increasing the probability that the client will see at least one of the redundant packets.

Buffer Overflow

The primary causes of buffer overflow are slow clients and missing data.

Slow Client

If the client is too slow, it may not be able to write out data fast enough, causing its buffer capacity to be exceeded. To determine if this is the case, look to the client's log file.

First, look for a line in the form:

```
Current block: 3289, High block: 3353
```

In this example, the value of the `numpktblks` parameter is 64. The fact that the difference between the current block (3289) and the high block (3353) is 64 indicates that all the buffers are in use.

Following this line are entries for every block that is not full. If there are no such entries or just a few near the high block range, it shows that most of the buffers are full, but the agent has not had the chance to write them out yet. For example, if the following line is:

```
Block: 3353, 32 packets of 256
```

It shows that all but the high block are full. This indicates that the agent may be too slow for the current raw data transfer rate. Here, you may want to consider increasing the inter-packet delay to see if the agent can better keep up with the lower raw data transfer rate.

Missing Data

On the client, if a block is missing data, it cannot be written out. After that block becomes current, writing will stop and will not resume until the missing data is filled in. In the meantime, the remaining buffers are used to hold the incoming data. If the missing data is not filled in soon enough, the buffers may overflow. Normally, the client will NACK the missing data and the holes will be filled in long before this happens.

In the client's log file, the indicators of this condition are similar to those of the slow client case. The line:

```
Current block: 3289, High block: 3353
```

should look essentially the same, showing all of the buffers in use.

In this case however, the following line will show that the current buffer is not full:

```
Block: 3289, 32 packets of 256
```

Now the question becomes, why is this data missing? The agent should have sent a NACK requesting that this block be resent and the data should have been resent by the server.

There are two possibilities: the NACK was never sent or the server never received it.

First, let us see if the block was indeed NACK'ed. In the client's log file, look for the statistics associated with the block in question:

```
Block: 3289, 32 packets of 256Resends requested: 1
```

Here you see one NACK was sent for the block.

Now, see if all of the NACKs the client sent got through to the server. In the client log file, there should be a line in the form:

```
Total resend requests = 8
```

Here, you see that the agent sent eight NACKs to the server. In the server log file, look at the per-agent data. After the line:

```
Client stats:
```

is a list of agents and the number of NACKs the server has received from each. Using the agent's IP address, find the line associated with the client in question. It should look something like this:

```
Client: 16.119.237.171 (0xabed7710) NACKs = 8
```

Here you can see that the server did receive all the NACKs the client sent. If these numbers were not the same, it would indicate that one or more NACKs had been lost. In that case, you should increase the value of the `nackresend` parameter. This will cause each NACK packet to be retransmitted more times, increasing the probability that the client will see at least one of the redundant packets.

For the case where the server has seen all the NACKs sent from the client, it probably indicates that the client did not issue a NACK when it needed to.

In the agent log file, look for the following line:

```
Max resend hits = n
```

Here, *n* is the number of times the client did not issue a NACK because the value of the `maxresendreq` parameter had been exceeded. If you cannot remedy the cause of the excessive number of NACKs, you may want to increase the value of `maxresendreq`, thus enabling the client to NACK a given block more times.

Test Modules

The following commands are provided as test tools that you can use to manually test different combinations of parameters, rather than running tests in the full Sentient environment.

Using GDMCSEND

Caution: The `gdmcsend` command can be run from a Windows environment only.

`gdmcsend` is the server side multicast send command.

In the following folder on the installation media, there is a script called `gdmsend.cmd` that can be used for testing:

```
Infrastructure\extended_infrastructure\multicast_server\multicast_
test_modules\
```

To start the multicast test sender module:

1. Copy the multicast test send modules (`gdmcsend.exe`, `gdmcsend.cmd`, and `TESTDATA0004`) from the following directory on the infrastructure CD to a temporary directory:
`extended_infrastructure\multicast_server\multicast_test_module`
2. Rename `TESTDATA0004` to `GDMCTESTDATA`.
3. Edit `gdmsend.cmd` and change `DP` on line 19 from `0.0` to `0.5`.
4. Edit `gdmsend.cmd` and change `OFFSET` on line 49 from `60` to `0`.
5. Run `gdmsend`.

If you want to modify the script, use a text editor to open the file and modify the parameters. Then, you can run this file to test the changes you made.

Note: When setting values for parameters that apply to both `gdmcsend` and `gdmrecv`, the values must match.

The following are two forms of the command and the valid options for each. Explanations of the parameters follow.

Use this command if you are using reliable delivery resend mode.

```
gdmcsend -rm D|B -ma multicast_address -mp multicast_port -np nac_
port-f file_name -npb nblocks -ppb npackets[-dp1 delay] [-dp delay] [-
dl delay] [-lc n] [-lf log_file][-nr n] [-ttl n] [-lpr n] [-lprd
delay] [-offset n_bytes][-ni ip_address][-tf throttle_frequency] [-ti
```

```
throttle_increment)[-tmax throttle_maximum] [-tmin throttle_minimum] [-tthigh high_throttle_threshold] [-ttlow low_throttle_threshold]
```

Use this command if you are using the fixed resend mode, which resends each packet block a fixed number of times.

```
gdmcsend -rm F -ma multicast_address -mp multicast_port -f file_name-ppb npackets -nr number_of_resends[-dp1 delay] [-dp delay] [-lf log_file] [-nr n] [-ttl n] [-lpr n] [-lprd delay] [-offset n_bytes] [-ni ip_address]
```

gdmcsend Command Options

Option	Corresponding parameter in mcast.cfg	Description	Default
<code>-dl <i>linger_delay</i></code>	<code>lingerdelay</code>	The delay, in milliseconds, between checking for resend requests after the last packet has been sent.	64.0
<code>-dp<i>delay</i></code>	<code>gddelaybp</code>	Delay, in milliseconds, after sending each packet.	0.0625
<code>-dp1<i>delay</i></code>	N/A	Delay, in milliseconds, after sending the first packet.	5
<code>-f <i>filename</i></code>	N/A	Name of the file containing the data to be sent.	N/A
<code>-lc <i>n</i></code>	<code>lingercount</code>	Linger count. The number of times to check for resend requests (NACKs), after the last packet has been sent.	256
<code>-lf <i>log_file</i></code>	N/A	The name of the log file. The log file is stored in the directory where you execute the command. You may use this parameter to change the name of the log file or provide an absolute or relative path.	<code>gdmcsend.log</code>
<code>-lpr<i>n</i></code>	<code>lprcount</code>	Last packet resend. The number of times to resend the last packet.	4
<code>-lprd<i>delay</i></code>	<code>lprdelay</code>	Last packet resend delay. The delay, in milliseconds, between last packet resends.	0.25
<code>-ma <i>multicast_address</i></code>	N/A	Multicast address. The address to which the data is sent.	N/A
<code>-mp<i>multicast_port</i></code>	N/A	Multicast port. The port to which the data is sent.	N/A

Option	Corresponding parameter in mcast.cfg	Description	Default
<code>-ni ip_address</code>	N/A	Network interface. The IP address identifies the specific local network interface to use when sending data.	selected automatically
<code>-np nac_port</code>	N/A	NACK port. The port from which resend requests are read.	9514
<code>-npb nblocks</code>	N/A	Number of packet blocks. The number of packet blocks available to be resent.	N/A
<code>-nr n</code>		The number of times to resend each packet. This option only applies when resend mode (<code>-rm</code>) is set to F .	0
<code>-offset n_bytes</code>	N/A	Skip the first <i>n_bytes</i> bytes of the file.	0
<code>-ppb npackets</code>	N/A	Packets per block. The number of packets in each packet block (must be a multiple of 32).	N/A
<code>-rm F B D</code>	N/A	Resend mode. F = fixed Each packet block is resent a fixed number of times (as specified by the <code>-nr</code> option). B = backup Resend all blocks from the lowest number requested to the current block (last block sent by the sender). D = discrete Resend only requested blocks.	B
<code>-tf throttle_frequency</code>	<code>throtfreq</code>	The minimum number of packet blocks between throttle adjustments.	8
<code>-ti throttle_increment</code>	<code>throtincr</code>	The value, in milliseconds, that is added to (or subtracted from) the current inter-packet delay each time the throttle needs to be adjusted.	0.01
<code>-tmax throttle_maximum</code>	<code>throtmax</code>	The maximum value of the inter-packet delay before throttling will stop.	0.5
<code>-tmin throttle_</code>	<code>throtmin</code>	The minimum value of the inter-packet delay before throttling will	0.0

Option	Corresponding parameter in mcast.cfg	Description	Default
		stop.	
<code>-tthigh <i>high_throttle_threshold</i></code>	<code>throthighth</code>	The average number of resends per block that will trigger an increment of the inter-packet delay.	-1 (throttling disabled)
<code>-ttlow <i>low_throttle_threshold</i></code>	<code>throtlowth</code>	The average number of resends per block that will trigger a decrement of the inter-packet delay.	-1 (throttling disabled)
<code>-ttl <i>n</i></code>	<code>tthl</code>	Time to live. The number of subnets that the packet will reach.	3

Using GDMCRECV

`Gdmcrecv` is the client side multicast receive command.

The `gdmcrecv` command can only be run from the Service Operating System as booted from the OS Manager CD-ROM in TESTMODE. If necessary, use a nano editor to modify the shell script, `gdmcrecv.sh`.

Note: When setting values for parameters that apply to both `gdmcsend` and `gdmcrecv`, the values must match.

The following are two sample commands. Explanations of the parameters follow.

Use this command if you are using reliable delivery resend mode.

```
gdmcrecv -rm D|B -ma multicast_address -mp multicast_port -np nac_port-na nac_address -npb nblocks -ppb npackets[-t timeout_minutes] [-nit timeout_minutes][-mr max_resend_req] [-nd nac_delay] [-nr nac_resends][-lf log_file] [-bt block_threshold] [-ni ip_address][-pmf freq] [-stderr]
```

Use this command if you are using the fixed resend mode which resends each packet block a fixed number of times.

```
gdmcrecv -rm F -ma multicast_address -mp multicast_port -ppb npackets[-t timeout_minutes] [-nit timeout_minutes][-lf log_file] [-ni ip_address]
```

`gdmcrecv` Command Options

Option	Corresponding Parameter in meast.cfg	Description	Default
<code>-bt <i>block_threshold</i></code>	N/A	Block threshold. When the number of used blocks exceeds this value,	0

Option	Corresponding Parameter in meast.cfg	Description	Default
		resend requests are sent even if all data has been received in order to slow down the sender.	
-lf <i>log_file</i>	N/A	Name of log file. The log file is stored in the directory where you execute the command. You may use this parameter to change the name of the log file or provide an absolute or relative path.	gdmcrecv.log
-ma <i>multicast_address</i>	N/A	Multicast address. The address from which data is read.	N/A
-mp <i>multicast_port</i>	N/A	Multicast port. The port from which data is read.	N/A
-mr <i>max_resend_req</i>	maxrsndreq	The maximum number of times a resend can be requested for each block.	128
-na <i>nac_address</i>	N/A	NACK address. The address to which resend requests are sent.	N/A
-nd <i>nac_delay</i>	nacdelay	The delay, in milliseconds, between sending resend requests.	0.5
-ni <i>ip_address</i>	N/A	Network interface. The IP address that identifies the specific local network interface to use to receive data.	selected automatically
-nit <i>timeout_minutes</i>	netinactio	The time to wait, in minutes, between received packets before failing.	5
-np <i>nac_port</i>	N/A	NACK port. The port to which resend requests are sent.	9514
-npb <i>nblocks</i>	numpktblks	Number of packet blocks. The maximum number of packet blocks that can be serviced by resend requests at any point in time.	N/A
-nr <i>nac_resend</i>	nacresend	The number of times each NACK should be resent.	4
-pmf <i>freq</i>	N/A	Progress meter frequency. The	0

Option	Corresponding Parameter in meast.cfg	Description	Default
		progress meter is updated after every freq packet blocks have been written out. A value of zero disables the progress meter.	
<code>-ppb</code> <i>npackets</i>	<code>pktsperblk</code>	Packets per block. The number of packets in each packet block (must be a multiple of 32 and match the value used by the sender).	
<code>-rm F B D</code>	N/A	Resend mode. F = fixed Each packet block is resent a fixed number of times (as specified by the <code>-nr</code> option). B = backup Resend all blocks from the lowest requested to the current. The receiver will only send resend requests (NACKs) for the lowest block needed. D = discrete Resend only requested blocks. The receiver will send resend requests (NACKs) for every block needed.	B
<code>-stderr</code>	N/A	Write log messages to <i>stderr</i> (standard error), as well as the log file.	FALSE
<code>-t timeout_</code> <i>minutes</i>	<code>recvtimeout</code>	The maximum time, in minutes, before the data transfer fails.	45

Customizing OS Deployment by Using Exit Points and Add-Ons

Sentient provides two features that you can use to dynamically customize your OS deployments:

- **Add-On Packages** enable you to deploy arbitrary sets of data during image deployment.
- **User Exit Points** enable you to execute custom code at various stages of the deployment

These feature can be used in both ImageX and Windows Setup deployments. They enable you to build a more flexible and controlled OS deployment environment, where static OS images can be transformed during deployment to meet the complex requirements of your enterprise.

For example, you can use these features to inject drivers during the OS deployment. To do this, you must:

1. Publish the drivers.
For instructions, see *Publishing OS Add-Ons and Extra Production OS (POS) Drivers* in the *Sentient Enterprise Console Online Help*.
2. Use either Windows Setup or ImageX for deployment.
3. For Windows Setup deployments, use an `unattend.xml` file that contains a reference to `C:\osmgr.hlp\drivers`. The sample `unattend.xml` files provided by Sentient contain this reference.
For ImageX deployments, a reference to `C:\osmgr.hlp\drivers` is added to the registry on the reference machine before the OS image is captured.

User Exit Points

Caution: Although exit points provide a way to customize OS deployments, extreme caution needs to be exercised when implementing such exit points. You must take care not to interfere with the Sentient OS deployment. Thorough testing of such custom solutions is mandatory.

In addition to the existing Personality and Data Capture exit points, Sentient has enabled several new, formal exit points to allow customization of the OS deployment process. These exit points are defined for the following purposes, but could be used for alternative processing. The following table details when they are run:

- Before disk partitioning occurs
- Before the OS and its resource files are downloaded
- Before the OS will be installed
- After the OS has been installed but before the reboot occurs

Add-On Methods

Sentient also allows for methods and/or data to be downloaded to the service OS RAM drive or to the production OS during image installation. This data is called Add-On. Any number of Add-On methods can be run during the deployment.

Add-Ons are typically used to dynamically inject non-critical device drivers into a Vista, Windows 2008 Server or later OS image prior to its deployment, but they are not limited to such use. They are enabled by simply publishing one or more files using a new ADD-ON publish feature in the Publisher.

Drivers (or other methods) must be published. After publishing, the Add-On package is connected to a Service. Formal exit-point command files are also published as an Add-On and connected to the appropriate Service. The Publisher has been extended to allow a formal publishing session to the OS Manager ADDON class. See *Publishing OS Add-Ons and Extra Production OS (POS) Drivers* in the *Sentient Enterprise Console Online Help*.

The Add-Ons feature is integrated with the normal image deployment process. Published Add-Ons are downloaded as needed along with the Service's resources. The Service OS Add-Ons are downloaded before Production OS Add-Ons.

Add-Ons that run in the Service OS have the extension `.sdd` (Service add-on), and Add-Ons that run under the Production OS use the `.pdd` extension (Production add-on). Both `.pdd` and `.sdd` files are created as TAR files (compressed archive in `.tar` format with path-information).

Exit Point and Add-On processing can be used with ImageX and Windows Setup deployment methods.

Publishing Add-On Methods

The Sentient Publisher now has a drop-down option to publish an OS ADDON. Choose this option when publishing Service OS or Production OS methods and any associated data, like device drivers.

Publishing Add-On Methods

Add-Ons are published to the Configuration Server Database (CSDB) into the OS.ADDON class as new instances that are connected to an OS ZSERVICE. Publishing always occurs for a specific OS service. The wiring is done automatically.

For new installations of HPCA version 7.50 and later, the wiring setup is done as part of the creation of a new OS service. For migrated environments, administrators will need to connect the service and any associated ADD-ONS manually.

Deployment of the OS is triggered by launching an OS Deployment job. See “OS Management” in the *Sentient Enterprise Console Online Help* for more information.

Add-Ons and user exits points are processed during the OS deployment process.

Agent Execution of Add-On Methods

Service OS Add-On (`.sdd`) files are downloaded and extracted into the root of `X:\` (typically the local RAM drive in the Windows PE service OS) as the first step of the install WIM phase. The OS Manager Agent aborts if the available free-space on this drive drops below 20 MB while downloading and extracting any `.sdd` files.

Production OS Add-On (`.pdd`) files—for example, device driver files—are downloaded and extracted into the root of the new OS partition, always `C:\`, after downloading the `.WIM`. When computing the size for the new OS partition, the OS Manager Agent will take into consideration the uncompressed size of all resolved `.pdd` archives (from the ADDON resource meta-data attribute ZRSCSIZE). Each `.pdd` file is stored and named after its object-ID and not the original name it was published as. The agent executes user-provided exit-point scripts during its normal flow of operation if well-known exit-point specific script files exist after `.sdd/.pdd` extraction.

The processing sequence for a typical OS Management deployment follows the sequence below when installing an OS. Note that if an exit point is defined to reside in both the `X:\work` and `C:\`, then the exit method will be called twice. Be certain to publish a directory structure that contains a `\work` subdirectory (see examples above).

Add-Ons are extracted in the order they are resolved by the Configuration Server. There is no sequencing. All exit-point scripts are first being searched for in `X:\Work` and executed if available.

- Service OS methods, `.sdd` extensions, run from the `x:\` drive
- Production OS methods, `.pdd` extensions, run from the `c:\` drive

Note: Exit points and any associated data is NOT deleted automatically.

Agent Execution of Add-On Methods – Important Information

- Exit-point execution and error-handling is similar to LME Apply-method execution.
- After looking for and executing either the `PreInstall.cmd` and `PreReboot.cmd` exit points in the `X:\Work` directory, the agent will also look under `C:\` directory and executes exit points found in this directory in addition to the exit points in the `X:\Work` directory.
- The agent does a drive-/partition re-synch after having run `PrePartition.cmd` and `PreDownload.cmd`.
- The Sentient Publisher is extended to create new `OS.ZSERVICE` instances of deployment-types `ImageX` and `Windows Setup` with a connection to `OS.ADDON.PDD_<Servicename>_*` by default
- The tar archive(s) published as `PDD_<Servicename>_<Suffix>` will be used as Production OS Add-On(s) (for example, for additional drivers) without the need to use the CSDB Editor for standard cases. The `<Suffix>` should be used to classify Add-Ons.
- Using the CSDB Editor, there can be more `.pdd` and/or `.sdd` ADDON connections and the standard `OS.ADDON.PDD_<Servicename>_*` connection can be replaced by something more selective (for example, leveraging model information).
- For publishing type Production OS add-ons, optionally ask for a corresponding service name/suffix and create an instance name in the form of `PDD_<Servicename>_<Suffix>`.
- The directory `C:\osmgr.hlp\drivers` is the suggested additional driver library location. Each driver `.pdd` ADDON needs to extract its contents into one driver-/driver-and-version specific subdirectory under `C:\osmgr.hlp\drivers`.
The current `unattend.xml` template will be changed to always include the `C:\osmgr.hlp\drivers` directory in the search-path for Plug-and-Play drivers.
For captured images, the `C:\osmgr.hlp\drivers` directory will have been added to the registry before capturing (in HPCA version 7.90 and later).

Caution: Subdirectories below `C:\osmgr.hlp\drivers` must not contain multiple dots in their names!

The folder `c:\osmgr.hlp` is automatically deleted during the first connect after deployment. However, if you have additional drivers published as PDD, the folder `c:\osmgr.hlp\drivers` is not deleted. You must not delete this folder manually because the drivers might be in use (or may be used later) by the Operating System.

- Older, pre-existing OS services need to re-publish their `unattend.xml` template (to add the driver path) and add the `OS.ADDON.PDD_<Servicename>_*` connection to leverage driver-injection.
As an alternative to re-publishing the modified `unattend.xml` template, a sample `PreInstall.cmd` script could be provided (and added to any `.pdd` package) to extend the PnP search-path for `C:\osmgr.hlp\drivers` by modifying the xml file on the fly.

OS Deployment Processing Using User Exits

There are 3 phases during a normal deployment process:

- Pre-OS deployment phase where pertinent information is extracted from the device prior to its being provisioned
- OS deployment phase
- Post-OS deployment phase where the machine may be added to a directory and any extracted information is restored

Pre-OS Deployment Phase

You can use the Sentient Personality Backup and Restore feature to capture user files and settings for later restoration after the device has had its OS provisioned. For more information, see *Personality Backup and Restore* in the *Sentient Enterprise Console Online Help*.

OS Deployment Phase

1. Start Service OS
2. User exit: `PrePartition.cmd`
 - Runs before partitioning is completed (ImageX or Windows Setup only)
 - `PrePartition.cmd` can be defined as a Service OS Add-On (.sdd). It cannot be defined as a Production OS Add-On (.pdd).
 - Typically used to partition disk drive(s). Agent does a drive-/partition re-synch after having run `PrePartition.cmd` user exit.
 - OS is not available
 - Network is available
 - RAM drive of the WinPE service OS is available
You can use the environment variable `SystemDrive` to find the drive letter of the Windows PE RAM drive.
3. Partition Disk Drive(s)
4. User exit: `PreDownload.cmd`
 - Runs after partitioning, before downloading/extracting OS and other resource files
 - `PreDownload.cmd` can be defined as a Service OS Device Driver (.sdd) Add-On. It cannot be defined as a Production OS Device Driver (.pdd) Add-On.
 - May be used to modify the environment after the disk has been partitioned
 - Agent does a drive-/partition re-synch after running `PreDownload.cmd` user exit.
 - OS is not available
 - Network is available
 - RAM drive of the WinPE service OS is available

5. Download OS and other resource files
6. User exit: `PreInstall.cmd`
 - Runs after downloading/extracting resource files and before running Windows Setup or ImageX extract that installs the OS
 - `PreInstall.cmd` can be defined as a Service OS Add-On (`.sdd`) or a Production OS Add-On (`.pdd`).
 - May be used to customize the environment before the OS has been installed (for example, Customize OS install configuration files or add or replace OS files)
 - Could be used to replace the `install.wim` file from the Configuration Server with one from another location, if desired
 - OS is not available
 - Network is available
 - RAM drive of the WinPE service OS is available
7. Install OS using ImageX or Windows Setup
8. User exit: `PreReboot.cmd`
 - Runs after Windows Setup returned or ImageX extract and before triggering reboot
 - `PreReboot.cmd` can be defined as a Service OS Add-On (`.sdd`) or a Production OS Add-On (`.pdd`).
 - May be used to modify the environment after the OS has been installed and to prepare for methods to be run immediately or after the reboot has occurred (for example, Registry run or runonce keys)
 - WinPE OS is available and is the running OS
 - Facilities and interfaces limited to what is provided by the WinPE service OS. See the Windows AIK for WinPE features and limitations.
 - Native OS is installed, but not yet actively running
 - SYSPREP has not yet run
 - Network is available
 - RAM drive of the WinPE service OS is available
9. Reboot device
10. Post-reboot device start-up
 - Completes native OS installation
 - Runs SYSPREP
 - Install Sentient Enterprise Agent (runsync)
 - Sentient Agent runs first connect to Configuration Server
 - Metadata download
 - OS Manager client methods are run
 - OS state set to `_DESIRED_`
 - Runs User exit: `novapdr.cmd` - Optional

- Typically used to restore user files and personality captured before the device was provisioned.
- Full native OS is available (for example, Windows 7)
- Network is available

Customizing the Windows Answer File

This appendix contains the following topics:

- ["Customizing the unattend.xml File" below](#)
- ["XML File Processing in Sentient " on page 281](#)
- ["About the .subs" on page 283](#)

These topics pertain to the process of capturing and publishing operating system images so that they can be deployed to managed devices in unattended mode (requiring no user interaction on the client devices).

Customizing the unattend.xml File

Sentient provides an answer file that you can use for unattended OS installations. This answer file is called `unattend.xml`.

Each operating system and architecture (for example, 32-bit or 64-bit) has its own `unattend.xml` file. The files are located in subdirectories of:

```
<InstallDir>\Data\OSManagerServer\capture-conf
```

The header at the beginning of the file shows you the OS, architecture, and deployment method to which the file applies.

If you want to use the `unattend.xml` file that Persistent Systems provides, you must modify it for your environment before you publish the OS image. Here are some settings that you will want to customize:

- ["ProductKey" on the next page](#)
- ["TimeZone" on page 278](#)
- ["RegisteredOwner and RegisteredOrganization" on page 278](#)
- ["JoinDomain" on page 279](#)
- ["MetaData" on page 280](#)

Caution: At a minimum, you must specify a valid product key (see ["ProductKey" on the next page](#)). Modifying the other settings discussed here is optional.

Use a text editor to modify a copy of the pertinent `unattend.xml` file. You can name this copy anything that you like as long as it has the `.xml` file extension. When you publish the OS image, you will specify where your customized answer file is located.

Note: The Windows Automated Installation Kit (AIK) includes a file called `Unattend.chm`. This is a compiled online help file that contains reference information about the contents of the `unattend.xml` file. See this help file for more detailed information about the settings discussed here and the other settings available that you can customize. To open the file, simply double-click `Unattend.chm`.

ProductKey

The `<ProductKey>` element appears in different places in the `unattend.xml` file depending on the specific OS image, architecture, and deployment method that you are using. The `<ProductKey>` is a string with 29 characters that is delimited like this:

XXXXX-XXXXX-XXXXX-XXXXX-XXXXX

Note: For all DVD installations, be sure that `/IMAGE/INDEX` is pointing to the correct image on the DVD (see ["MetaData" on page 280](#)).

Retail Editions

For retail editions of Windows (for example, Windows 7 Ultimate), make the following modifications:

- Put a valid product key in the `<Key>` element inside the `<ProductKey>` element. For example:

```
<UserData>

<AcceptEula>true</AcceptEula>

<ProductKey>

<Key>XXXXXX-XXXXXX-XXXXXX-XXXXXX-XXXXXX</Key>

<WillShowUI>OnError</WillShowUI>

</ProductKey>

</UserData>
```

This element is located in the "Microsoft-Windows-Setup" component in the "WindowsPE" in pass.

- Remove the entire `<ProductKey>` element located in the "Microsoft-Windows-Shell-Setup" component in the "specialize" pass:

```
<ProductKey>XXXXXX-XXXXXX-XXXXXX-XXXXXX-XXXXXX</ProductKey>
```

Business Editions

For business editions of Windows (including Business, Enterprise, Professional, or Server editions), make the following modifications:

- Remove all characters in the `<Key>` element located in the located in the "Microsoft-Windows-Setup" component in the "WindowsPE" in pass (see example above):
`<Key></Key>`

- Put a valid product key in the `<ProductKey>` element located in the "Microsoft-Windows-Shell-Setup" component in the "specialize" pass:

```
<ProductKey>XXXXXX-XXXXXX-XXXXXX-XXXXXX-XXXXXX</ProductKey>
```

If you are using a Volume License Multiple Activation Key (MAK), use that in the `<ProductKey>` element.

Note: In the Windows AIK, the `<Key></Key>` element supports an empty value, but the `<ProductKey>` element does not—hence `<ProductKey>` element must be deleted if it is not being used (see ["Retail Editions" on the previous page](#)).

64-Bit Platforms

When you are using a DVD with the Windows Setup deployment method on some 64-bit architectures, be sure to make the following modifications:

- Remove all characters in the `<Key>` element located in the located in the "Microsoft-Windows-Setup" component in the "WindowsPE" in pass (see example above):

```
<Key></Key>
```

- Put a valid product key in the `<ProductKey>` element located in the "Microsoft-Windows-Shell-Setup" component in the "specialize" pass:

```
<ProductKey>XXXXXX-XXXXXX-XXXXXX-XXXXXX-XXXXXX</ProductKey>
```

- Make sure that `/IMAGE/INDEX` points to the correct image on the media (see ["MetaData" on page 280](#)).

- Change "amd64" to "x86" in the following component specifications in the "WindowsPE" pass:

```
<component name="Microsoft-Windows-International-Core-WinPE"
processorArchitecture="amd64" ...
```

```
<component name="Microsoft-Windows-Setup"
processorArchitecture="amd64" ...
```

- During publishing, when you are prompted for the source directory, specify the one from the 32-bit media for the same operating system.
 - Special instructions for Windows 2008 R2 x64:
 - Use the Windows 7 Enterprise Edition 32-bit installation media.
 - Before you publish the OS image, follow these steps:
 - From the Windows 7 32-bit installation media, copy the *mediaDrive*:\sources folder to `c:\temp`
 - Remove the Windows 7 media, and load the Windows 2008 R2 x64 media.
 - From the Windows 2008 R2 x64 installation media, copy the *mediaDrive*:\sources\license folder to `c:\temp\sources`

If prompted to overwrite existing files, do so.

This ensures that the Windows 2008 Server R2 EULAs are available from the Windows 7 installation folder.

Note: For more information, see the “ProductKey” topic in the `Unattend.chm` help file included in the Windows AIK.

TimeZone

The `<TimeZone>` element appears in different places in the `unattend.xml` file depending on the specific OS image, architecture, and deployment method that you are using.

For example, in the `unattend.xml` file for a captured Windows 7 (x86) image, there are two places where the `<TimeZone>` element appears:

- In the `Microsoft-Windows-Shell-Setup` component under `<settings pass="oobeSystem">`
- In the `Microsoft-Windows-Shell-Setup` component under `<settings pass="specialize">`

Change the `<TimeZone>` to match the target devices to which the OS will be deployed. For example:

```
<TimeZone>Eastern Standard Time</TimeZone>
```

It is important that the spelling of the time zone exactly match the spelling used in the Windows Registry. For more information, see the “Language Pack Default Values” topic in the `Unattend.chm` help file included in the Windows AIK.

Note: Greenwich Mean Time is now known as Coordinated Universal Time.

Note: On a computer running Windows 7 you can use the `tzutil` command to list the time zone for that computer.

RegisteredOwner and RegisteredOrganization

These elements appear in different places in the `unattend.xml` file depending on the specific OS image, architecture, and deployment method that you are using.

For example, in the `unattend.xml` file for a captured Windows 7 (x86) image, there are two places where these two elements appear:

- In the `Microsoft-Windows-Shell-Setup` component under `<settings pass="oobeSystem">`
- In the `Microsoft-Windows-Shell-Setup` component under `<settings pass="specialize">`

Change these elements to the name of your company (or the entity to whom the operating system is registered). For example:

```
<RegisteredOrganization>Hewlett-  
Packard</RegisteredOrganization>
```

```
<RegisteredOwner>Hewlett-Packard</RegisteredOwner>
```

These strings can be up to 256 characters in length.

See the “RegisteredOrganization” and “RegisteredOwner” topics in the `Unattend.chm` help file included in the Windows AIK for more information.

JoinDomain

You can instruct target devices to either join a domain or a workgroup after the OS is installed. Workgroup mode is the default. The targets can join a domain using a secure method.

Secure Domain Join

To join a domain using a secure method, open the .SUB file of a published image or of an image that is going to be published, and make the below changes:

1. Open the substitutes file from `<InstallDir>/Data/OSManager/capture-conf/<OS>/<ARCHITECT>` (this is not applicable for an image that is already published)

2. Modify COMPDOMN in

```
//un:settings[@pass='specialize']//un:component[@name='Microsoft-  
Windows-UnattendedJoin']  
[@processorArchitecture='amd64']/un:Identification/un:Credentials/u  
n:Domain,COMPDOMN
```

AND

```
//un:settings[@pass='specialize']//un:component[@name='Microsoft-  
Windows-UnattendedJoin']  
[@processorArchitecture='amd64']/un:Identification/un:JoinDomain,CO  
MPDOMN
```

3. Encrypt Username and Password by performing the following steps:
 - a. Go to `<RCInstallPath>/OSManagerServer` directory

- b. Double-click `nvdkit-rca-osm.exe`

- c. Type the following command for Username encryption:

```
password encrypt<ENC_USERNAME> AES
```

The Username is case sensitive and once encrypted, it will look something like this:

```
<AES256>yR3qcC+TWFOF6B0JGk5x1Q==
```

- d. Type the following command for Password encryption:

```
password encrypt<ENC_PASSWORD> AES
```

Your password will be encrypted and will look something like this:

```
<AES256>kITMqDenvFUdpBaYt8XBg==
```

- e. Copy the encrypted username and password and paste it in the substitutes file.

4. Modify ENC_USERNAME in

```
//un:settings[@pass='specialize']//un:component[@name='Microsoft-  
Windows-UnattendedJoin']  
[@processorArchitecture='amd64']/un:Identification/un:Credentials/u  
n:Username,ENC_USERNAME
```

Note: Ensure ENC_USERNAME is the Admin User of the domain.

5. Modify ENC_PASSWORD in

```
//un:settings[@pass='specialize']//un:component[@name='Microsoft-  
Windows-UnattendedJoin']  
[@processorArchitecture='amd64']/un:Identification/un:Credentials/u  
n:Password,ENC_PASSWORD
```

Note: Ensure ENC_PASSWORD is the password of the Admin User.

Note:

- The user specified must have an access level sufficient to join the domain.
- If any of this information is missing or incorrect, the device will join a workgroup instead of a domain.
- If the target device was previously managed by Sentient, and the device was previously a member of a domain, the stored domain information will override the contents of the <Domain>, <JoinDomain>, <Username>, and <Password> elements in the unattend.xml file.
- Any information that is set centrally—for example, by using an OS management script to set the domain—will override information in unattend.xml.

MetaData

If you are deploying an operating system image directly from a DVD, you must specify the location of that image within the WIM file on the DVD. In the WIM file, this information is organized like this:

```
<WIM>  
  
<IMAGE INDEX="2">  
  
<NAME>MyWIM</NAME>  
  
<DESCRIPTION>MyCustomWindowsImage</DESCRIPTION>  
  
</IMAGE>  
  
</WIM>
```

In the unattend.xml file, the image information is specified in the <MetaData> element in the Microsoft-Windows-Setup component hierarchy under <settings pass="WindowsPE">. For example:

```
<MetaData>
```

```
<Key>/IMAGE/INDEX</Key>  
  
<Value>2</Value>  
  
</MetaData>
```

The `<Key>` element indicates which data item in the WIM file to match. It can be any of the following:

- `IMAGE/INDEX`
- `IMAGE/NAME`
- `IMAGE/DESCRIPTION`

The `<Value>` element indicates what the value of this data item should be. Here, the image to be deployed has an `IMAGE/INDEX` value of 2 in the WIM file.

You can extract a list of the images in a WIM file by using the following command:

```
imagex /info WIMFileName > c:\info.txt
```

Here, *WIMFileName* is the name of the WIM file (for example, `install.wim`). Be sure to redirect the output of the command to a text file (as shown here) so that you can easily search through the results.

For more information, see the *MetaData* topic in the `Unattend.chm` help file included in the Windows AIK.

XML File Processing in Sentient

The `unattend.xml` file that you publish is overlaid on top of any `unattend.xml` file that is present in the image that was published.

Before Sentient starts the image install, the published XML is combined with the `substitutes` file to generate the final `unattend.xml`.

This combining of files is done by Sentient before it starts the actual image installation. The previously exposed `substitutes` file is now used behind the scenes. Each operating system and architecture (for example, 32-bit or 64-bit) has its own file. The files are located in subdirectories of:

```
<InstallDir>\Data\OSManagerServer\capture-conf
```

The correct file is selected automatically depending on the processor architecture of the image being published.

The following table lists the settings in the `unattend.xml` file that are updated when the `substitutes` file is published.

Caution: The settings in blue (`CommandLine`, `Path`, and both instances of `PartitionID`) are required for Sentient to work. They cannot be removed.

Substitutes File

Settings Pass	Component	Path	Setting	Override Value
windowsPE	Microsoft-Windows-Setup	DiskConfiguration/Disk/ModifyPartitions/ModifyPartition	PartitionID	DISKPART volume ID to which Sentient will install the OS
windowsPE	Microsoft-Windows-Setup	ImageInstall/OSImage/InstallTo/	PartitionID	DISKPART volume ID to which Sentient will install the OS
windowsPE	Microsoft-Windows-Setup	ImageInstall/OSImage/InstallFrom/	Path	WIM file to use for installation
oobeSystem	Microsoft-Windows-Shell-Setup	AutoLogon/	Domain	Computer name (for auto-logon)
specialize	Microsoft-Windows-Shell-Setup	AutoLogon/	Domain	Local computer name (for auto-logon)
specialize	Microsoft-Windows-UnattendedJoin	Identification/Credentials/	Domain	Centrally set domain using getmachinename.tcl or pre-existing device entry in the Sentient Core Console
specialize	Microsoft-Windows-UnattendedJoin	Identification/	JoinDomain	Centrally set domain using getmachinename.tcl or pre-existing device entry in the Sentient console
specialize	Microsoft-Windows-Shell-Setup		ComputerName	Computer name
oobeSystem	Microsoft-Windows-Shell-Setup	FirstLogonCommands/SynchronousCommand	CommandLine	Path to agent install media installer

You can, if required, customize the `substitutes` file to disable certain customizations or to add new ones. You cannot however remove or change the `PartitionID` or `CommandLine` settings.

About the .subs

Caution: Sentient now enables you to specify the source of this information when you run the Publisher.

Note: This topic does not apply to Windows XP or Windows 2003.

The Sentient Publisher is backward compatible. It supports publishing saved OS images that consist of a `.WIM` file, a `.EDM` file, a `.XML` file, and a `.SUB` file.

If you choose to manually pre-create `*.SUBS` and `*.XML` files, they must have the same prefix as the `*.WIM` file. For example: `vista.WIM`, `vista.SUBS`, and `vista.XML`. All three files must be stored in the same directory.

Note: When you run the Sentient Publisher, if it finds a `*.SUBS` and `*.XML` file in the same directory as the `*.WIM` file, it will not prompt you for an `unattend.xml` file.

Sentient provides samples of these files on the Image Capture media in subdirectories of the following folder:

```
\samples\unattend
```

If you choose to use the sample files, rename them and then modify them as needed—for example, setting the `<TimeZone>` and the `<ProductKey>`.

The `*.XML` file is an answer file that contains standard information as well as placeholders for information that will be included from `*.SUBS`. You can use the Microsoft Windows System Image Manager (SIM) tool to make additions to the `*.XML` file. If you do so, you must first open the corresponding `*.WIM` file before opening `*.XML`.

Caution: If you choose to use `*.XML` and `*.SUBS` files, you must specify your Windows installation product key in the `*.XML` file.

Do not delete any XML values from this file! If you modify the `*.XML` file incorrectly, you may cause your installation to fail.

If you see errors in the Messages section in the SIM tool similar to "...The value `$$SUBSTR$$` is invalid.." you can ignore them.

When you save the file, you may also see a message similar to "There are validation errors in the answer file. Do you want to continue?" Click **Yes** to continue.

The `*.SUBS` file is the "substitutes" file that lists each XML item to be modified in `*.XML` and what its value should be. The lines in the `*.SUBS` file are known as XPATHs.

Note: Information entered in the *.SUBS file takes precedence over information in the *.XML file.

Example of Substitution

If you want to see how substitution works, you can review the following example which will show how the JoinDomain attribute gets changed from “anything” in the *filename*.XML file to “VistaTeam” in the unattend.xml file.

Note: Code that appears within <> should appear all on one line in the *.XML file.

1. Locate the appropriate unattend*.xml and substitutes files for your operating system, target device architecture, and deployment method. These files are located under samples\ on the ImageCapture CD.
2. Make a copy of the unattend*.xml file, and name it *filename*.XML, where *filename* matches the name of your .WIM file. Store the copy in the same directory as your .WIM file.
3. Make a copy of the substitutes file, and name it *filename*.SUB. Store the copy in the same directory as your .WIM file.
You should now have the following three files in one directory:

- *filename*.WIM
- *filename*.XML
- *filename*.SUB

4. Locate the XML element for JoinDomain in the *filename*.XML file. It should look similar to this example:

```
<?xml version="1.0" encoding="utf-8"?>

<unattend xmlns="urn:schemas-microsoft-com:unattend">

  <settings pass="specialize">

    <component name="Microsoft-Windows-UnattendedJoin"
      processorArchitecture="x86" publicKeyToken="31bf3856ad364e35"
      language="neutral"
      versionScope="nonSxS"
      xmlns:wcm="http://schemas.microsoft.com/WMIconfig/2002/State"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

      <Identification>

        <JoinDomain>anything</JoinDomain>

      </Identification>

    </component>

  </settings>
```



```
<cpu:offlineImage cpu:source="wim://radiafcovcm/c$/vista_
inst/vista.wim#Windows Vista ULTIMATE" xmlns:cpu="urn:schemas-
microsoft-com:cpu"/>

</unattend>
```

5. Modify the following XPATH element in the *filename*.SUB file. Note that this XPATH element appears on a single line in the *filename*.SUB file.

```
//un:settings[@pass='specialize']/un:component[@name=Microsoft-
Windows-UnattendedJoin']
[@processorArchitecture='x86']/un:Identification/un:
JoinDomain,VistaTeam
```

During deployment of the operating system, the *filename*.SUB and *filename*.XML files will be combined to create an *unattend.xml* file that is used to provide information during all phases of the Windows setup. In this example, the JoinDomain attribute will be set to **VistaTeam**.

Building a Windows PE Service OS

This section includes the following topics:

- ["About the Windows PE Service OS Screen" below](#)
- ["About the Custom Build Script" on page 287](#)
- ["Prerequisites" on page 287](#)
- ["Adding Drivers to the Windows PE Service OS" on the next page](#)
- ["Building a Custom Windows PE Service OS" on page 290](#)
- ["Using Customized build.config Files \(Advanced Option\)" on page 295](#)

About the Windows PE Service OS Screen

A Service OS is a pre-installation environment that is based on a lightweight operating system such as Linux or Windows PE. The Service OS does the following things:

1. Boots into the target hardware
2. Loads all the drivers that are needed in order for that hardware to function correctly
3. Downloads and runs Sentient programs which, in turn, download and install OS images

The Service OS is used to perform the following types of operations:

- Operations to hardware on a target device (for example, a BIOS update or hardware configuration)
- Provisioning target devices (for example, deploying an OS)
- Capturing an OS image

Whenever a Service OS starts, the Service OS screen appears on the pertinent device. When an OS image is being captured, for example, the Service OS screen appears on the reference machine. When an OS is being deployed, the Service OS screen appears on the target device.

The Windows PE Service OS screen shows you the status of the operation. The right side of Windows PE Service OS screen shows you a scrolling log of the steps that are being performed.

- A green checkmark icon indicates that a particular step either is in progress or has been successfully completed.
- A yellow triangle icon is a warning that something may be wrong.
- A red X icon indicates that this step in the capture or deployment has failed.
- A blue question mark (?) icon indicates that input is required.

Information about the current step always appears at the bottom of the list of messages. A scroll bar appears on the far right if there is not enough room to list all of the messages.

If the operation is successful, a green check mark appears on the left side of the Service OS screen with further instructions.

At this point, you can take one of two actions:

- Click **Reboot** to reboot the device into the installed operating system.
- Click **Shutdown** to shut the device down.

After you click one of these buttons, a status message will appear briefly under the progress bar on the right side of the screen before the reboot or shutdown occurs.

If the operation is not successful, a red X appears on the left side of the Service OS screen with information about the nature of the failure. If the operation fails, you can use the scroll bar on the right side of the screen to view information about the hardware detected and determine where in the process the failure occurred. At this point, you can take one of three actions:

- Click **Reboot** to reboot the device into the installed operating system.
- Click **Shutdown** to shut the device down.
- Click **Exit to console** to close the Service OS screen and reveal the console window.

After you click one of these buttons, a status message will appear briefly under the progress bar on the right side of the screen before the action that you selected occurs.

Note: Deployment on UEFI devices creates three partitions. These are EFI system partition (ESP), Microsoft reserved partition (MSR), and OS partition.

To capture, publish, and deploy, only ImageX method is supported. CD boot is not supported on UEFI devices.

Adding Drivers to the Windows PE Service OS

You can add drivers to the Windows PE Service OS when you run the build scripts. For example, if you have a driver that requires a reboot, you must do it in “offline” mode. This means that the build script will pause, and you can make any necessary changes at that time. This is described in detail in the steps below.

Note: Additionally, you can add drivers to Windows PE while it is running (“online” mode). The drivers must be fully contained without need for a reboot, and the device must have

connectivity to the Sentient server.

During the startup of the Windows PE Service OS, any drivers that exist in `<InstallDir>\OSManagerServer\SOS\WinPE\drivers` will be downloaded and installed using `drvload.exe`.

About the Custom Build Script

It provides a script that enables you to:

- Add font support for Chinese and Japanese.
- Update the Windows Preinstallation Environment (PE) Service OS when a new `winpe.wim` file is made available through an updated Windows Automated Installation Kit (AIK).
- Add extra drivers or packages that do not exist in the Windows PE Service OS provided.
- Use the information in this chapter in conjunction with your knowledge of the Microsoft Windows AIK to rebuild the Windows PE Service OS with the drivers and packages necessary for your environment.
- Create a new `ImageCapture.iso` if you have updates that need to be applied, such as a change to the default Service OS or to the configuration of the boot menu.
- Create a new `ImageDeploy.iso` if you have updates that need to be applied such as a change to the default Service OS or to the configuration of the boot menu.

Prerequisites

Before you can use the script provided by Persistent Systems to build a custom Windows PE Service OS, you must satisfy a number of prerequisites. See the following topics for details:

- ["Process Knowledge" below](#)
- ["Administrator Machine" on the next page](#)
- ["Media" on the next page](#)
- ["Files and Directories" on page 289](#)
- ["Support for Other Languages" on page 289](#)
- ["Advanced Option" on page 290](#)

Caution: Do not attempt to run this script on a machine where incompatible software is installed. See the prerequisites for the ["Administrator Machine" on the next page](#).

Process Knowledge

You will need a basic understanding of Microsoft's preinstallation customization process to add drivers and other information to the Windows PE Service OS.

Administrator Machine

To run the script, you will need an “administrator” machine with the 32-bit version of the Windows Automated Installation Kit (AIK) installed. This is the machine that you will use to build the customized Windows PE Service OS.

Caution: Do NOT use a machine where any of the following are installed:

- Sentient Boot Server
- Sentient Core or Satellite server
- Cygwin

The supported operating systems when you use ADK 8.1 are:

- Windows 7 family
- Windows 8 family
- Windows 8.1 family
- Windows 10
- Windows Server 2008 R2 family
- Windows Server 2012 family
- Windows Server 2012 R2 family

The supported operating systems when you use WAIK 3.1 are:

- Windows Server 2003 with SP2
- Windows Vista SP1
- Windows 7 SP1
- Windows 7 family
- Windows Server 2008 R2 SP1
- Windows Server 2008 family
- Windows Server 2008 R2 family

Media

You will need the following media (DVD or CD-ROM):

- Sentient product media
- Sentient Image Capture media
- Sentient Image Deploy media

Files and Directories

- You will need the build scripts bundle, `build_scripts.zip`, from the Sentient product media.
- If you are generating a new `ImageCapture.iso` or `ImageDeploy.iso`, you must do the following to include the updated files required.
 - a. Create a build items directory on the "Administrator Machine" on the previous page, such as `c:\build_items`.
 - b. *Optional:* Copy any updated files that you have received from it to this build items directory. Create subdirectories as needed, based on the structure of the Image Capture or Image Deploy media.
If any of the required files are not in this directory, you will be prompted to insert the previous Image Capture or Image Deploy media so the files can be copied.
 - c. *Optional:* Include `rombl_capture.cfg` and `rombl_deploy.cfg` in the build items directory for use on the appropriate ISO. These files contain information such as the menu timeout settings and the default Service OS.
To create these files, copy `rombl.cfg` from the previous `ImageCapture.iso` or `ImageDeploy.iso`, and modify and rename the files as necessary.

If you do not include these files in the build items directory, the script prompts you for the previous CD-ROM and retrieves the files from the media. If you choose not to insert a CD-ROM, a standard `rombl.cfg` file will be created automatically.

Support for Other Languages

If you want to add support for Chinese or Japanese without making additional changes to the ISO:

- Remove any existing `winpe.wim` files from the `build_items` directory.
- Copy `winpe_cjk.wim` from the `\custom_build\lang_support` directory on the product CD-ROM to the `build_items` directory.
- Rename `winpe_cjk.wim` to `winpe.wim`.
- See "Building a Custom Windows PE Service OS" on the next page to run the script.

Caution: To use the Chinese or Japanese enabled `winpe.wim` file without rebuilding the `winpe.wim` file, be sure to type N when prompted to recreate the `winpe.wim` file.

- If you are using the ImageDeploy CD to install from CD—or you are installing from a cache and want messages to appear in your local language—copy the `\custom_build\lang_support\i18n` directory from the product media to the build items directory. You may remove the `.msg` files that are not needed for your local language.

Advanced Option

Caution: The following information is intended for experienced Sentient administrators only. Do not attempt to customize an existing `winpe.wim` file unless you have a strong understanding of both OS Management under Sentient and the Microsoft Windows AIK tools.

If you are using a pre-existing `winpe.wim` file:

- It is strongly recommended that the pre-existing `winpe.wim` was built using the same version of the Windows AIK that is installed on the machine where you are executing the build scripts.
- The `winpe.wim` file must have the following packages installed:
 - For Windows AIK version 1.1
 - WinPE-HTA-Package
 - WinPE-Scripting-Package
 - WinPE-XML-Package
 - WinPE-WMI Package
 - For Windows AIK version 2.0 and 3.1:
 - WinPE-hta.cab
 - WinPE-scripting.cab
 - WinPE-wmi.cab
 - WinPE-setup.cab
 - WinPE-legacysetup.cab
 - WinPE-setup-client.cab
 - WinPE-setup-server.cab
- If your `winpe.wim` file was prepared using the `peimg /prep` command, see the Microsoft documentation for the Windows AIK, `peimg`, and ImageX for restrictions (only applies to Windows AIK 1.1).

Building a Custom Windows PE Service OS

The following topics show you how to obtain and use the script that Sentient provides to build a custom Windows PE Service OS. You can also use the Sentient OS Manager Deployment toolkit to build a custom Windows PE Service OS.

Note: To build custom Windows PE Service OS, it is recommended to use the Sentient OS Manager Deployment Toolkit from Sentient version 9.20 onwards.

- To obtain the script and prepare to run it, see ["Get the Script" below](#).
- To launch the script and specify the information that it requires, see ["Run the Script" below](#).

Note: Be sure to review and satisfy the ["Prerequisites" on page 287](#) before you invoke the script.

- After you run the script, see ["Additional Information" on page 294](#).

Using the Script

The script that you will need to build a custom Windows PE Service OS is located on the Sentient installation media.

Get the Script

Follow the procedure below to obtain the script and prepare to run it on your ["Administrator Machine" on page 288](#). To obtain the script and make it available on the Administrator Machine

1. Copy the following file from the installation media to a location on the ["Administrator Machine" on page 288](#) (where the Windows AIK is installed).:
`<InstallDir>\media\ISO\roms\build_scripts.zip`
2. Unzip this file to a directory of your choice (such as `C:\Build_scripts`).

Run the Script

Note: This procedure assumes that you have satisfied the prerequisites (see ["Prerequisites" on page 287](#)) and obtained the script (see ["Get the Script" above](#)).

For each question that the script asks, the default response is shown in brackets on the far right side of the window. For example:

Type a number between 1 and 9 [1]:

In this example, the default response is 1. To accept the default, press **Enter**.

Perform the following steps to build a custom windows PE service OS:

1. Go to a Windows command prompt, and change to the `<version>` folder in the directory that you just created.
Here, `<version>` is the SOS version number. For example:
`C:\ Build_scripts\<version>`
2. Type **run**
In a few moments, a list of Sentient versions is displayed.
3. Type the number corresponding to the Sentient version that you want to use.

4. When asked whether you want to create a new WIM file, type `Y` or `N`.

Note: If the script that you are running was provided with an OS management service OS update (driver patch), you cannot create a WIM file. You can only create ISO files.

If you typed `Y`, follow these steps to specify the WIM file options:

- a. You will be prompted to type the path to your Windows AIK tools directory. For example, `C:\Program Files\Windows AIK\Tools`
- b. When asked whether you want to use the `winpe.wim` file from the Microsoft Windows AIK, type `Y` or `N`.

Note: It is strongly recommended that you use the `winpe.wim` file from the Microsoft Windows AIK.

If you type `N`, you will be reminded to ensure that your pre-existing `winpe.wim` file is built according to specifications. Then, you will be prompted to specify the fully qualified path of the pre-existing `winpe.wim` file.

- c. When asked whether you want to include font support for Chinese, and Japanese, type `Y` or `N`.
 - d. When asked whether you want to pause the WIM creation process to add extra drivers or packages, type `Y` or `N`.
 - e. When asked whether you want to provide a path to a directory containing additional drivers to be added during the WIM creation process, type `Y` or `N`.
 - f. If you typed `Y`, you will be asked to type the fully qualified path to the directory containing the drivers.
5. The next group of questions determines whether you want to create a new Image Capture ISO or Image Deploy ISO and which Service OS to include.
- You should create a new Image Capture ISO (type `Y`) if any of the following conditions are true:
 - You have received updated files from Persistent Systems Software Support.
 - You have rebuilt `winpe.wim`, and you are using the ISO to perform the capture.
 - You need to change the configuration (`rombl.cfg`).
 - You should create a new Image Deploy ISO (type `Y`) if any of the following conditions are true:
 - You have received updated files from PSL Software Support.
 - You have rebuilt `winpe.wim`, and you are booting from the CD during deployment.
 - You need to change the configuration (`rombl.cfg`).

Follow these steps to specify the ISO options:

- a. When asked whether you want to create a new Image Capture ISO, type `Y` or `N`.
- b. When asked whether you want to create a new Image Deploy ISO, type `Y` or `N`.

- c. If you answered **Y** to question, you will be asked which Service OSs to include on the ISO. Type the appropriate selection. Then, press **Enter**.
- d. When asked if you want to create a new `rombl.cfg` or use a pre-existing `rombl.cfg` file, choose one of the following actions:
 - To create a new `rombl.cfg` file, type **1**, and press **Enter**.
 - To use a pre-existing `rombl.cfg` file and skip to the next step, type **2**, press **Enter**.
- e. When asked which Service OS you want to boot by default, type the appropriate selection. Then, press **Enter**.
- f. Specify how the boot menu should be handled in each ISO that you are creating. There are three choices:

0	Hide the boot menu from the user of the target device. The default service OS that you specified in steps above will be used.
-1	Show the boot menu, and wait for a user response. The response will override the default Service OS setting.
Number greater than zero	Show the boot menu, and wait this number of seconds for a user response before booting into the default service OS specified in the step "When asked which Service OS you want to boot by default, type the appropriate selection. Then, press Enter." above.

- g. When asked if you want to change the port used to connect to the Sentient infrastructure, type **Y** or **N**. The default port is 3466.
- h. When asked if want to specify the ISO boot load value that gets included in the ISO boot sector, type **Y** or **N**.

Caution: Use this option only if you experience problems using the default value and you have been instructed by Persistent Systems Software Support to change it.

Certain hardware models require a boot load segment of 0x2000 because of a BIOS issue. Other models cannot boot from the CD when the boot load segment is something other than the default loader segment of the EI Torito ISO format: 0x0000.

To specify the boot load segment setting, type **1**, **2** or **3**:

1	RCA default (0x2000) – works with most BIOSs
2	ISO default (0x0000) – gets translated to 0x07c0 by most BIOSs
3	Manually type a value

Then press **Enter**. If you typed **3**, specify the boot load segment setting as a hexadecimal string beginning with 0x.

- i. When prompted for the fully qualified path to the build items, type the directory name (such as `C:\build_items`), and press **Enter**.

This completes the questions pertaining to the Image Capture and Image Deploy ISOs.

6. When prompted for the fully qualified path for the temporary work directory, type a directory name (such as `C:\build_work`). This directory will be referred to as the *<work-dir>* in later steps.

Note: If the directory already exists and has information in it, you will be asked whether you want to delete the information or not. If you choose No, you will be asked to type a directory again. If you prefer to exit, press **Ctrl + C** to exit the process. If you choose Yes, the information will be overwritten.

7. When prompted for the fully qualified path for the output directory, type a directory name such as (`C:\build_output`).

Note: If you are prompted to create ISOs for CAS, type N.

8. The build process takes some time, as you will see from the on-screen messaging. When it is finished, you will see a message indicating that the Service OS creation process completed successfully and be returned to a command prompt.

After the build is completed, go to the directory where the `Windows PE.wim` was stored, such as `C:\WinPE_output`, and perform the following action:

Boot Method for Target Devices	Action Required
PXE	Copy <code>winpe.wim</code> from the output directory to <i><InstallDir>\BootServer\X86PC\UNDI\boot</i>
LSB	Use the CSDB Editor to replace <code>winpe.wim</code> in the LSB package.
CD	Create a new ISO using the Windows PE scripts.

If you chose to create `ImageCapture.iso` or `ImageDeploy.iso`, they will be stored in the same output directory.

Additional Information

After you provide all the information that the custom Windows PE service OS build script requires (see "Run the Script" on page 291), the following tasks occur:

1. If files that are required to build the ISO are not in the build items directory, you must insert the CD/DVD, and the files will be copied. If you choose not to insert the CD/DVD, the build process will terminate.
2. The information that you entered is saved, and the Windows PE directory creation begins.
3. If you indicated that you wanted to pause the WIM creation process to add extra drivers or packages, the process will pause after the Windows PE directory is created and the contents of `winpe.wim` are extracted into the WIM directory (for example, `C:\build_work\WIM`). There are two ways to do this:

Method A: Use a Windows AIK tools to make your modifications.

If you are using Windows AIK version 1.1, use the `peimg.exe` command. The default location of this executable file is:

```
C:\Program Files\Windows AIK\Tools\PETools\peimg.exe
```

If you are using Windows AIK version 2.0 or 3.1, use the `dism.exe` command. The default location of this executable file is:

```
C:\Program Files\Windows AIK\Tools\Servicing\dism.exe
```

See the Windows AIK documentation for information about how to use these commands (or use the `/help` command line option).

Method B: Add the drivers to a driver list.

After you see a message indicating that all required information is gathered, the `build.config` file will be created in the `C:\Build_scripts` directory to store the information that is needed to build the `winpe.wim` and the ISOs. You can use a text editor to open this file and add the appropriate drivers below the empty `DRIVERS` list.

For example:

```
declare DRIVERS = " cdrom.inf \  
e:\\tmp\\work\\WIM\\windows\\inf\\adp94xx.inf \  
e:\\tmp\\work\\WIM\\windows\\inf\\3com*.inf "
```

Note: Because the back-slash (`\`) is a special character, you must “escape” it by using two back-slashes, as shown in this example.

Note that all lines except the last end with a back-slash. In this case, the back-slash indicates a continuation of the declaration.

If you do not specify a directory, the script will search for the driver in the `<work-dir>\WIM\Windows\inf` directory.

If you prefer, you can provide a fully qualified path that specifies the location and driver, such as `c:\\anydirectory\\mydrivers.inf`

You can also specify a path with a filename containing a wild card, such as `c:\\anydirectory\\md*.inf`, which will install all `md*.inf` files found in `c:\\anydirectory`.

After you are finished, type **run** to continue, and the drivers will be added to `winpe.wim`.

If you run the script again in the future, you will be prompted about whether you want to keep the `build.config` file or replace it with a new one. Also, the script will pause automatically. If you do not have additional packages or drivers to add, simply type **run** to continue.

Using Customized build.config Files (Advanced Option)

If you choose, you can take an existing `build.config` file and save it with another name. You may want to do this if you need to maintain various sets of configurations, or if you are testing based on an existing configuration. You can add drivers to the file as specified above.

Place the file in the directory where you unzipped the `build_scripts.zip` file, such as `C:\build_scripts`.

When you run the script, instead of typing `run` use the following command:

```
run.cmd -f mybuild.cfg
```

If you do not include the `-f` parameter, the default `build.config` file will be created and used.

Using the Sentient OS Manager Deployment Toolkit

In addition to the Build Script, you can use the Sentient OS Manager Deployment Toolkit to build WinPE Wim files, Image Capture ISO, and Image Deploy ISO files.

Note: Currently, the creation of USB media is not supported by the Sentient OS Manager Deployment Toolkit.

Prerequisites

To use the Sentient OS Manager Deployment Toolkit, ensure the following prerequisites are in place:

- ADK 8.1 or WAIK must be installed on the machine
- .net Framework must be available

Note: It is not recommended to upgrade from ADK 8.1 to ADK 10.0. However, both ADK 8.1 and ADK 10.0 can co-exist.

To build a custom Windows PE Service OS using this toolkit, perform the following steps:

1. Go to `<InstallDir>\Media\iso\roms` and access the **deployment-toolkit-ui** folder.
2. Run **DeploymentToolKit.exe** using administrative privileges.

The Sentient OS Manager Deployment Toolkit window appears. By default, the General tab is enabled.

3. Under WAIK/ADK Option, choose one of the following:
 - ADK
 - WAIK

Depending upon the selection, the default location of WAIK/ADK is displayed. If you have installed WAIK/ADK in a different location, click  to browse to the non-default location.

To Build Custom WinPE Wim Files

1. In **Product Version**, select the Sentient version for which the WinPE is being built.
2. Under Build Options, select **WinPE**.

3. Under **Folder Options**, specify the following:
 - Temporary Folder - Browse and select a directory location to store the temporary files that are created during the build process.

Note: Files or folders in this directory get automatically deleted. Also, at the end of the processing, the contents of this folder might get deleted.

- Output Folder - Browse and select a directory location to store the files that are built using the Sentient OS Manager Deployment Toolkit.
4. Click **Next**.

Depending on the Build Option selected, the appropriate tab on the left panel is enabled. For example, if you select WinPE in the build option, on clicking **Next**, the WinPE tab is enabled. The WinPE Build Options window enables you to add details to build WinPE for x86 (32 bit) and WinPE for x64 (64 bit for UEFI devices).

5. Under WinPE Wim x86 Build Options, specify the details to build WinPE for x86 (32 bit):
 - a. Select **WinPEx86** to build WinPE for x86 (32 bit).
 - In **Base WinPE Wim**, the base winpe file is selected from ADK/WAIK by default. To use a non-default or a custom winpe as a base, click  to browse to the location.
 - In **New WinPE Wim**, specify the name and location of the new WINPE.wim file.
 - b. Select **Add Drivers** and provide the path under additional drivers path.
 - c. Select **Add Font Support** to include font support for Chinese and Japanese.
6. Under WinPE Wim x64 Build Options, specify details to build WinPE for x64 (64 bit for UEFI devices):
 - a. Select **WinPEx64 (UEFI)** to build WinPE for x64 (64 bit).
 - i. In **Base WinPE Wim**, the base winpe file is selected from ADK/WAIK by default. To use a non-default winpe or a custom winpe as a base, click  to browse to the location.
 - ii. In **New WinPE Wim**, specify the name and location of the new WINPE.wim file.
 - b. Select **Add Drivers** and provide the path under additional drivers path.
 - c. Select **Add Font Support** to include font support for Chinese and Japanese.
7. Click **Next**.

The Preview tab is enabled providing the summary of the options selected and displaying the items that will be built.

8. Click **Start** to begin the process to build the Wim files.

To change any of the details specified, click **Back** and make the appropriate changes.

To Create Image Capture ISO and Image Deploy ISO Files

1. On the General tab, under Build Options, select **ISO Media**.

The ISO tab is enabled.

2. On the ISO tab, under Build Options, select the following:
 - **Image Capture:** To build ImageCapture.iso files
 - **Image Deploy:** To build ImageDeploy.iso files
 - **Build New Rombl.cf:** To build new rombl.cfg file that will be a part of the ImageCapture.iso and ImageDeploy.iso files
3. Select **Use Non Default Boot Load Segment** to specify the boot load segment that gets included in the ISO boot sector. If the boot segment is not specified, it takes the value as 0x7C0.
4. In **ISO Output Directory**, by default, the Output directory specified in the General tab appears.
5. In **Updated Items Directory**, specify the directory containing the updated files.
6. In **Path to x86 WinPE**, specify the location of 32 bit WinPE WIM file that is going to be a part of the ISO.

Note: Using the Sentient OS Manager Deployment Toolkit, you cannot create ImageCapture.iso and ImageDeploy.iso files for UEFI devices. For information on capturing images for UEFI devices, see *Capturing Images for UEFI Devices* section in the *Sentient Enterprise System Administrators Guide*.

7. Depending upon the requirement, choose the following options:

Tabs	If you select this option	Specify this
Image Capture	CD/DVD	CD/DVD Drive Letter: The drive containing ImageCapture.iso. This will be used as the base file to create the ImageCapture.iso file.
	ISO File	Path to ISO: Path of the ImageCapture.iso. This file will be used as the base file to create the new ImageCapture.iso file. Note: If you have installed the OSMFirstTimeSetup tool, use the ISOs created from this tool. Do not use the MR version ISOs while creating Image Capture ISO using Sentient OS Manager Deployment Toolkit.
Image Deploy	CD/DVD	CD/DVD Drive Letter: The drive containing the ImageDeploy.iso. This will be used as the base file to create the ImageDeploy.iso file.
	ISO File	Path to ISO: Path of the ImageDeploy.iso. This file will be used as the base file to create the new ImageDeploy.ISO file.
	Add OS Resources	Note: If you have installed the OSMFirstTimeSetup tool, use the ISOs created from this tool. Do not use the MR version ISOs while creating Image Deploy ISO using Sentient OS Manager Deployment Toolkit.

		Resources Path: The root directory containing exported OS resources if these resources are to be included in ImageDeploy.iso. The default path is <code><InstallDir>\Data\ServiceDecks\CDDeployment</code>.
ROMBL Cfg		Add WinPE Service OS: To add WinPE section in the rombl.cfg Add Linux Service OS: To add linux SOS section in rombl.cfg Default SOS: Default SOS for ImageCapture.iso and ImageDeploy.iso files. You can select one of the following: ■ WinPE ■ Linux Menu Timeout: Timeout value in seconds during SOS selection. The default SOS is Linux. If there is a delay in providing the value within the specified timeout, the file boots into Linux SOS. ISVR Port: ISVR value for the new rombl.cfg file. The default port is 3466.

8. Click **Next**.

The Preview tab is enabled providing the summary of the options selected and displaying the items that will be built.

9. Click **Start** to begin the process to create the ISO files.

To change any of the details specified, click **Back** and make the appropriate changes.

Configuration Parameters in Deployment Toolkit

The following table lists the parameters that can be configured through the Deployment Toolkit UI. Alternatively, you can configure these parameters in `<today's date>.ini` file that is created under `<InstallDir>\Media\iso\roms\deployment-toolkit-ui\Config` folder when you run the DeploymentToolKit.exe:

Configurable Parameters	Enables you to	Applicable Values/Example
Product_Version_Idx	Specify the Sentient product version for which the WinPE is being built	RCA <product version>
WIM_Create_WIMfile	Create wim files	1, 0
ISO_Create_Media	Create ISO media	1, 0
Temp_Work_Dir	Specify the directory location to store the temporary files that are created during the build process	For example, <code>c:\Build_Scripts\temp</code>

ADK	Indicate if ADK is installed	1, 0
Kit_Tools_Dir	Specify the ADK installed path	For example, C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit
PE_CREATE_X86_WIM	Create x86 wim file	1, 0
PE_CREATE_X64_WIM	Create x64 wim file	1, 0
PE_OUTPUT_DIR	Specify the output directory that contains the created wim files	For example, c:\Build_Scripts\out
PE_Temp_Mount_Dir	Specify the temporary mount point for winpe creation	For example, c:\Build_ Scripts\temp\WinPE
PE_USE_FORCEUNSIGNED_FLAG	Force unsigned drivers to be incorporated in the winpe wim file. Set this flag in the <today's date>.ini file. This parameter is not configurable through the Deployment Toolkit UI.	1, 0
PE_BASE_WINPE_WIM	Specify the path of base winpe to be used for updating	For example, C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Windows Preinstallation Environment\x86\en- us\winpe.wim
PE_Add_Extra_Driver	Specify if there are additional drivers to be added in the winpe	1, 0
PE_Extra_Driver_Dir	Specify the path to the additional drivers to be added to winpe wim	For example, c:\Build_ Scripts\Drivers\x86
ISO_Create_CaptureISO	Create imagecapture.iso	1, 0
ISO_Create_DeployISO	Create imagedeploy.iso	1, 0
ISO_Capture_Source	Specify the base ISO to be used for updating imagecapture.iso	For example, C:\Build_ Scripts\deployment- toolkit-ui\CAE_

		ImageCapture.iso
ISO_Deploy_Source	Specify the base ISO to be used for updating imagedeploy.iso	For example, C:\Build_Scripts\deployment-toolkit-ui\CAE_ImageDeploy.iso
ISO_Output_Dir	Specify the path to the updated ISO files	For example, c:\Build_Scripts\out\ISO

Supported Locales

This chapter shows you how to set the locale for the service operating system (SOS) and OS Manager System Agent messaging.

Caution: If you do not need the SOS and OS Manager System Agent messaging to be localized, do not make any of the changes discussed in this chapter.

Note: When you are creating an OS image with Sentient, the locale for your reference and target devices must match. For example, if you want to create a Simplified Chinese OS image, you must run the OS Image Capture Tool on a Simplified Chinese reference machine.

Supported Languages

The following languages are supported:

- Brazilian Portuguese
- English
- French
- German
- Italian
- Japanese
- Simplified Chinese
- Spanish

Changing the Locale

To add support for a specific locale in a PXE environment:

1. Use a UNIX® based text editor to open this file:
`<InstallDir>\BootServer\X86PC\UNDI\boot\linux.cfg\default`

Note: Do not use editors that automatically convert to Windows format, such as Notepad, to modify the Boot Server configuration files. You can use Nano or WordPad.

The file looks similar to the following:

```
[OS Manager]DFLTSVOS=_SVC_LINUX_ISVR=10.10.10.1:3466

[_SVC_LINUX_]KERNEL=bzImageAPPEND initrd=rootfs.gz root=/dev/ram0
rw quiet pci=nommconf

[SVC_PEX86]PEBCD=rombl.bcdPEAPPEND=initrd=winpe.wim
```

2. For the Linux Service OS (SOS), add the LANG parameter to the end of the APPEND line.

For example:

```
APPEND initrd=rootfs.gz root=/dev/ram0 rw quiet pci=nommconf
LANG=zh_CN
```

For the WinPE SOS, add the LANG parameter to the end of the PEAPPEND line. For example:

```
PEAPPEND=initrd=winpe.wim LANG=zh_CN
```

The following languages are available:

Available Languages

Language	LANG Value
Brazilian Portuguese	pt_BR
English	en_US
French	fr_FR
German	de_DE
Italian	it_IT
Japanese	jp_JP
Simplified Chinese	zh_CN
Spanish	es_ES

3. Save and close the default file.

Note: If you specify `LANG=CJK`, the Linux SOS will start up with English messages until it switches to the locale specified in the pertinent BEHAVIOR instance in the CSDB (see ["Setting the System Language Parameter" on the next page](#) or the locale specified in the `ROMBL.CFG` file for the LSB case.

Setting the System Language Parameter

In this section, you will set the System Language parameter in the Behavior instance. Doing so sets the locale for the service operating system and OS Manager System Agent messaging. This affects PXE environments, LSB environments, and restoring operating systems from a CD-ROM or DVD.

To set policy to enable support for other languages:

1. Log in to the CSDB Editor.
2. Go to the appropriate PRIMARY.OS.BEHAVIOR instance.
3. Double-click the **Locale used in Service OS** attribute. The Editing dialog box opens.
4. In the **Locale used in Service OS** box, type in the code for the language that you want. For more information, see the codes listed in the table *Available Languages* in the section ["Changing the Locale" on page 301](#).
5. Click **OK** to save your change and close the dialog box.
6. Drag and drop the BEHAVIOR instance to the appropriate POLICY instance.

Double-Byte Support for Sysprep or unattend.txt files

If you are using double byte characters, the `unattend.txt` file must be encoded in UTF-8 coding. For Sysprep files, follow double-byte character rules as stated by Microsoft.

AppEvents

The following AppEvents are stored in the Events section in the ROM object:

App Events

Message	Description
CD install, no CD drive	A CD-based installation was requested but no CD-ROM drive exists on the machine.
Partition error	The OS Manager System Agent was unable to retrieve partition information (file retrieval problem).
Boot partition problem	The OS Manager System Agent was unable to determine the boot partition after the disk was partitioned.
Error Installing MBR	The OS Manager System Agent encountered an error while installing the Master Boot Record (MBR).
Error installing image	The OS Manager System Agent received an error while installing the OS image.
unattend.txt error	The <code>unattend.txt</code> file could not be retrieved from the server.

Message	Description
Sysprep.inf error	The <code>Sysprep.inf</code> file could not be retrieved from the server.
OS install Successful	OS was successfully installed.
NOOP install Successful	No OS install was required. Hardware Configuration Elements may have been processed and Sentient may have been updated to indicate that the machine is in desired state with respect to the OS currently installed OS.
HW config element apply failed	The application of a HW Configuration Element failed. Errors or warnings may be available in the log file.
Shadow HW config element apply failed	The application of a Shadow Hardware Configuration Element failed. You can find errors or warnings in <code>osselect.log</code> .
Admin activity required - Invalidate OS state	A Hardware Configuration Element failed or the installation of the OS failed. The OS state will be set to INVALID due to the failure.
Admin activity required - Multiple HW configurations resolved and central control	More than one HW Configuration was determined by policy. The target device could not determine which of these HW Configurations to use to reach desired state. The administrator or user must select the HW Configuration that needs to be applied to reach desired state.
Admin activity required - no eligible OS, unusable machine, machine shutdown	During policy resolution, no eligible OS was found for the device. The device may have no local OS or the device may be managed but the OS must be repaired (<code>_INCONSISTENT_OS</code>). The device is unusable and Sentient does not know how to proceed. Therefore, the device has been turned off until the administrator changes policy and sends a WOL to the machine.
Admin activity required - Multiple OSs resolved and central control	Multiple OSs were resolved for this device and administrative action is required because the user was not given the option to select the OS.
Admin activity required - Multiple OSs resolved and central control	During policy resolution, several eligible OSs were found for the device. However, the behavior setting does not allow for user selection of the OS. Therefore, the administrator must intervene and determine what OS should be installed on the device. Until then, the device is usable as long as the <code>OSSTATE</code> is not set to INVALID.
Admin activity required - No OS has been selected	During policy resolution, no eligible OS was found for the device. The device may have no local OS or the device may be managed but the OS is in need of repair (<code>_INCONSISTENT_OS</code>). The device is unusable and Sentient does not know how to proceed. Therefore, the device has

Message	Description
	been turned off until the administrator changes policy and sends a WOL to the machine.
Admin activity required - OSSTATE set to _INCONSISTENT_	On a managed device that was in its desired state, <code>Rombl.cfg</code> was lost. This may indicate serious corruption and therefore, Sentient changed the value of OS State to _INCONSISTENT_ and will allow the device to be used "as is". If possible, during the next Sentient OS Connect, <code>Rombl.cfg</code> will be recreated. If this does not happen, the administrator should force a reinstall of the OS.
Admin activity required - _UNMANAGED_OS_ is resolved through general policy criteria	An _UNMANAGED_OS_ was resolved for the device and administrative action is required.
Admin activity required - Corrupted OS, unusable, shutdown	The client's OS is corrupt and there is no enough information or the permission to overwrite the broken installation.
%1\$s %2\$s has been selected	%1 = "OS" or "Hardware Configuration" %2 = The name of the OS or LDS Indicates what has been selected based on policy.
%1\$s %2\$s already installed	%1 = "OS" %2 = "OS name" The OS referenced has previously been installed.
%1\$s %2\$s was installed	%1 = "OS" %2 = "OS name" The OS referenced was installed successfully.
No to install	A valid OS exists on the device and the user responded No to the prompt to perform an OS installation.
No was entered to Install acknowledgement	The user declined to reinstall an OS that policy dictated should be reinstalled.
Installing [%1\$s] on [%2\$s], OS type: [%3\$s]	%1 = "OS name" %2 = "partition or disk ID" %3 = "OS type"
Partitioning Hard Disk...	The deployment system is in the process of partitioning the hard disk that the OS will be installed to.
Please check the RPS configuration	Sentient failed to find the pertinent files on the core or satellite server. The OS management process will continue with a warning, but the deployment may fail because the files are missing.
Admin activity required - _UNMANAGED_OS_ is selected where an OS is to be installed	_UNMANAGED_OS_ was resolved for the device because it has no OS or because the device is managed but the OS must be repaired (_INCONSISTENT_OS). The device is unusable and Sentient does not know how to proceed.

Message	Description
	Therefore, the device has been turned off until the administrator changes policy and sends a WOL to the device.
Admin activity required - No OS has been selected	No OS was selected for this device and administrative action is required. This can occur when multiple OSs resolve and the behaviors are configured for CENTRAL selection. The administrator must arbitrate the OS.
OSSTATE has been set to _DESIRED_	The OS has been installed according to policy.
OSSTATE set to _DESIRED_	Sentient determined that it was not necessary to install an OS and set the system to desired state. OR Sentient determined that a selected OS needed to be installed; it installed successfully and the system was set to desired state.
Rebuilt ROMBL.CFG, OSSTATE was _INCONSISTENT_, now _DESIRED_	Sentient detected that the OSSTATE was INCONSISTENT. But, Sentient then determined that the system's install is OK and set the system to desired state.
Machine under OS management missing machine instance in Client Automation Portal	A managed device does not have a device object; one is created.
A machine previously having been in _DESIRED_ state came up with corrupted MBR/boot partition. Admin has to either manually repair this situation or explicitly invalidate it to force re-install according to policy.	A machine has been determined to be in a disaster recovery situation. Some part of the current install was detected to be broken, corrupt or is in another failure state. You will have to wait until the Admin forces a re-install or if the local user is allowed to force a re-install.

User Messages

The following messages may be displayed to the user of the target device. Messages remain on screen for 30 seconds and then depending on the situation, the machine will be powered off, rebooted, or the failed action will be attempted again.

User Messages

Message	User Action
This machine is installed with a factory pre-imaged OS that is managed by the Client Automation OS Manager. The Client Automation OS Manager System Agent is unable to connect to the Client Automation OS Manager	N/A

Message	User Action
infrastructure to configure this machine. The machine cannot be used. The system will retry later.	
The local machine does not contain a usable OS. Networking problems prevented the Client Automation OS Manager System Agent from connecting to the Client Automation OS Manager infrastructure to install this machine. The machine cannot be used. The system will retry later.	N/A
The local machine contains a usable OS. Networking problems prevented the Client Automation OS Manager System Agent from connecting to the Client Automation OS Manager infrastructure to determine policy for this machine. The machine will be booted to the local Operating System.	N/A
This machine has an OS installed but is not currently managed by the OS Manager. It contains a local partition but no management marker and no machine object. Select install to install an operating system according to policy or use to keep the existing operating system for now. Please select install or use .	Select install to install the resolved OS, or select use to continue to use the existing OS.
This machine is new to the OS Manager. The attempt to register this machine in the device information repository failed and it is not allowed be used. The system will retry later.	N/A
Please select one of the following roles which will be used, along with other policy criteria, to determine the correct configuration for this machine.	Select a role.
This machine has no local OS or the OS is invalid. An OS must be reinstalled. Policy indicates that there are no eligible OSs assigned to this machine. The administrator should verify that at least one of the OSs selected for this machine have the following characteristics: ACPI: \$::acpi APIC: \$::apic Minimum CPU speed: \$::cpuspeed Minimum RAM size: \$::mem Boot Hard Drive Type: \$::boottype Minimum Hard Drive Size: \$::hdsiz The machine cannot be used and will shut down until an administrator specifies policy and performs a Wake On LAN.	N/A
The current state of this machine is unusable. Policy returned multiple OSs for this machine. The machine will shut down until an administrator selects an eligible OS and performs a Wake On LAN.	N/A
The current state of this machine is unusable. Policy returned multiple Hardware Configurations for this machine. The machine will shut down until an administrator selects an eligible Hardware Configuration and performs a Wake On LAN.	N/A
Policy requires that the OS must be reinstalled on this machine. Select an OS from the following list:	Select an OS.
Policy requires that the Hardware Configuration must be reinstalled on this machine. Select a Hardware Configuration from the following list:	Select a Hardware

Message	User Action
	Configuration.
This machine has no local OS or the OS is invalid. It must be reinstalled. However, no eligible OSs have been returned for this machine. The machine cannot be used and will shut down until an administrator changes policy and performs a Wake On LAN.	N/A
This machine has no local OS or the OS is invalid. It must be reinstalled. However, the intended OS for this machine cannot be determined due to an error during resolution. The machine cannot be used and will shut down until an administrator changes policy and performs a Wake On LAN.	N/A
Policy requires that the OS for this machine must be reinstalled. Is it ok to install the new OS now?	Indicate whether it is okay to continue the installation.
Policy requires that the OS for this machine should be reinstalled. The selected OS is the same as the currently installed OS. Do you want to use the current installation or do you want to refresh the OS?	Specify whether to use the existing installation or to refresh the current OS.
This machine is in the process of having its Hardware Configuration modified. However, a critical element of the configuration has failed. The machine will shut down until an administrator fixes the problem and performs a Wake On LAN. Please contact your administrator.	N/A
This machine is in the process of having an OS installed. However, a critical aspect of the installation has failed. The machine will shut down until an administrator fixes the problem and performs a Wake On LAN. Please contact your administrator.	N/A
This machine is in the process of having its Hardware Configuration modified. However, a critical Hardware Configuration Element has failed due to incorrect or corrupt instructions. The machine will shut down until an administrator fixes the problem and performs a Wake On LAN. Please contact your administrator.	N/A

About the Boot Server

Sentient Enterprise is supported with two types of boot servers:

- PXE
- iPXE

PXE (Pre-execution Environment): It is Windows-based boot Server and Trivial File Transfer Protocol (TFTP) server for the Sentient environment. Note that the TFTP daemon runs secure mode.

Note: PXE uses DHCP broadcast, multicast, or UDP protocols and receives broadcasts. This means that if broadcast traffic is restricted between subnets, you must place PXE servers in each subnet, enable broadcasts (which may not be an option), or use a DHCP helper function to pass DHCP broadcast traffic. This situation is similar to that of standard DHCP servers and is probably well understood by your network administrator.

The PXE server is a low volume server. The TFTP server volume is slightly higher, but should only be transferring the OS management Boot Loader (less than 64 KB) on every target device boot and the Service OS *only* when a state change is required (such as, initial discovery, installation, or change of OS). This transfer will *not* occur for devices in desired state. Therefore, a few strategically placed PXE/TFTP servers should be able to support many clients. They should be accessible, however, on a relatively high-speed connection.

iPXE(Pre-execution Environment): It is an open-source network-boot firmware, based on gPXE(an open-source PXE client firmware and bootloader derived from Etherboot). It provides a full stack network protocols such as TFTP, SAN, and HTTP. It provides a full PXE implementation enhanced with additional advanced features.

Advantage of using iPXE:

- With PXE, bootable files get downloaded using TFTP protocol. But with the latest iPXE support, files get downloaded using HTTP protocol; thus, the download speed becomes faster.
- You can chain load into iPXE in a hassle-free way to get the functionality of iPXE without the inconvenience of reflashing, or you can use iPXE to replace the current PXE ROM on your network card.

Refer to the **Post installation Configuration Task** topic under the **Installing Satellite Server** section in the 11.0 Installation and Configuration guide. To know more about iPXE refer to the [link](#).

Prerequisites

- Do *not* configure your DHCP server to preclude the use of the Boot Server.
- PXE Client version 2.2 or higher.
- Do not install the Boot Server on a machine that has cygwin installed, because this is not supported.
- If you have more than one PXE server in your environment, each must be on a separate segment, and the PXE packets should not pass between the segments. You can use the Discover Boot Server utility to determine if there are PXE servers in your environment.

- A static IP address for the Boot Server.

Note: If the Sentient IP address or port is ever changed, you must update the Boot Server ISVR value and the ISVRPORT value in the Boot Server default file. The default file is typically located in *SystemDrive*:\Hewlett-Packard\CM\BootServer\X86PC\UNDI\boot\linux.cfg.

Do not use editors that automatically convert to Windows format, such as Notepad. Use Nano or WordPad to modify the Boot Server's configuration files.

- Remember that target devices must contain a PXE-compliant NIC card and be set to boot from the network. To determine whether a device contains a PXE-compliant NIC card; see the card's specifications.

Note: To enable PXE in your network environment:

In some network environments (such as those containing Cisco), the client may fail to PXE boot and you may need to modify the network port configuration.

For a Cisco switch, use the following:

```
set port channel off  
set spantree port fast enable
```

For all other vendors, consult their documentation.

Setting Default Service OS

When Sentient is installed, it is configured to use the Linux Service OS by default and only switches over to WinPE if required to by a particular management operation. Under certain circumstances, you may prefer to run an environment using WinPE as the default Service OS, switching over to Linux only if necessary. The following steps describe how to convert an environment to use WinPE as the default Service OS.

Caution: Changing the default Service OS will affect newly discovered target devices in HPCA 7.50 and higher only. Existing target devices will continue to operate using the Linux Service OS as the default.

To convert the default Service OS to WinPE:

1. Opening the Boot Server's default file. This is typically located in *<InstallDir>*\BootServer\X86PC\UNDI\boot\linux.cfg.

Caution: Do not use a text editor that automatically converts to Windows format, such as Notepad. Use Nano or WordPad to modify the Boot Server's configuration files.

2. Modify the settings for PXE:
 - a. In the OS Manager section, change the DFTLSVOS to `_SVC_PEX86_`.
 - b. Save and close the file.
3. Modify the setting for LSB by opening the Client Automation Administrator CSDB Editor and going to PRIMARY, OS, Operating Systems (ZSERVICE), Local Service Boot. In the right

pane, scroll to the Service OS List (ELGBLSOS) attribute.

- a. Double-click the attribute and change the setting to `_SVC_PEX86_`.
 - b. Save, and close the CSDB Editor.
4. Modify your deployment CD-ROM according to the instructions in “Building a Custom Windows PE Service OS” in the *Sentient Enterprise Console Online Help*.

Chapter 6

Usage Management

Usage Data Collection Processing

When collection is requested for a database, the agent compares the current archive data against the reported data for that database. All file inventory data is compared. Usage data is only compared for files that pass the inclusion filter. Any differences in inventory or usage are added to the collection file.

Once the collection file has been created, it is sent to the destination defined in the PRIMARY.USAGE.UMDESTPT class, COLLEST instance attribute.

To minimize bandwidth requirements, the collection files contain only what has not yet been sent to the specific database requesting the collection. These files also contain data aging and deletion information for removing old data from the database.

The history file contains all usage data, regardless if it has been requested for upload into a database for the machine and all of its users. The history file contains up to one year's data and as new days are added, data older than one year is deleted. The number of months is determined by the History Retention Months setting. This feature enables a database collection process to modify policy collection filters to capture any data that is up to a year old. If a new database is added as a destination point, then it can request any of this data from any machine that the Application Usage Manager has monitored—either presently or in the past.

Collection Point Destinations

A collection point is a directory destination to which a collection file is copied, and then automatically imported into a SQL database by the KB Server.

Collection Destination Point Unavailable

If a collection request fails to successfully copy the current usage data, then the collection file is placed in the *<InstallDir>\AUM Agent\Usage Manager\Collect* directory with the file name *DatabaseName.USDBase*, where Database Name is the file name set in the DBNAME instance of the PRIMARY.USAGE.UMDBASE class. This file can then be manually collected by copying it from the machine, or it is recreated with the latest application usage data and then collected when the agent receives the next collection request.

Agent Processing of Data Collection Request

The monitoring process collects data for every executable that has been run on the machine. This data is saved in the active monitoring file. Since administrators need reports on what exists on the machine, but has not been used, an executable inventory is also run to augment the active usage data collected.

Executable Inventory Scan and Options

Since the inventory capture process may take several minutes to complete, the inventory process can be configured during installation to be collected at a predefined time, daily or weekly at, for example, 1:00 A.M. Sunday. The request for data collection can aggregate the collected inventory data with the current usage monitoring data at the time of the collection request. Also, the inventory collection can be configured to run at the time the collection request is issued to provide a more up-to-date inventory. Since the executable file inventory scan can take several minutes, the request for collection is performed synchronously once the inventory scan has finished.

Initiating Collection of Agent Data

Collection of monitoring data is initiated by running an executable with the appropriate command line parameters. Each collection request is SQL database-specific and the technique of launching an executable enables one or more collection requests to the same agent computer for either the same or differently filtered data destined for different SQL databases.

Note: The usage time for appropriate applications is collected based on parameters that you select. This usage time should not be confused with focus time (the time that an application was in focus, that is, the active window in the forefront on a user's desktop).

Database Performance and Maintenance

An `Optional_Features` folder within the `<InstallDir>\Media\usage` folders include scripts and views that can be applied to a database to enhance the database performance.

Note: The script names may abbreviate Materialized to Mat, as in: `StepX_Define Filter Mat Tables and Indexes.sql`.

Using Standard Materialized Views and Filtered Materialized Views

The Materialized Views are used to enhance reporting view. Either one can be optionally applied to a database to improve its performance.

- **Standard Materialized Views (SMV)**
A feature where all the views are converted into tables and indexes are added to enhance the query speed.
- **Filtered Materialized Views (FMV)**
A feature similar to SMV, but differs in that it requires filters to be applied at the time the views are converted into tables. The filters are stored in a separate table. As an example, if a filter for `Notepad.exe` is selected, the FMV table is populated with only notepad details for all the devices. In this way, the customer can choose to see only those applications which are important to them.

Applying the Scripts for SMV or FMV

1. Stop the service for the Radia Knowledge Base Server. The service can be stopped and started through the Administrative Tools options of Windows Control Panel.
2. Use normal procedures to execute the database scripts, in the given order, provided in the following locations:
 - For SQL Server:
`<InstallDir>\Media\usage\Optional_Features\SQL Server\Filter Materialized Views`

Or

`<InstallDir>\Media\usage\Optional_Features\SQL Server\Std Materialized Views`
 - For Oracle:
`<InstallDir>\Media\usage\Optional_Features\Oracle\Filter Materialized Views`

Or

`<InstallDir>\Media\usage\Optional_Features\Oracle\Std Materialized Views`

Each of the above locations also includes a script to remove the view from your database. For example, for SQL Server and Filtered Materialized Views, the script name is:

`SQLServer-Remove Filter Mat Tables and Indexes.sql`.

Optional Utility Scripts

As a database administrator, use the following scripts to enhance the reporting view performance:

- `Purge_Computer_Data`: Deletes all data associated with the computer name. The computer name should be provided at the appropriate place in the script. The default value is MYCOMPUTER.
- `Purge_User_Data`: Deletes all data associated with the computer name and the user name. The computer name and the user name should be provided at the appropriate place in the script. The default values are MYCOMPUTER and BOB.
- `Delete All Windows OS Files from Database`: Deletes all Windows Operating System (OS)-related files from the Usage database.
- `sp_PurgeNonCurrentComputers`:
Deletes all the data associated with the device, which is not active based on the parameters passed to it and the last collection date of each device.

For example, `sp_PurgeNonCurrentComputers (90)` procedure deletes all non-current devices and data associated with it, if the last collection date is more than 90 days of time.

Miscellaneous Scripts for Oracle

Miscellaneous scripts are additional scripts that can be applied along with the utility scripts to enhance the reporting view performance.

- `Optional_Create_Public_Synonyms`: Creates public synonyms. The script may have to be edited for the Usage Manager's user names.
- `Optional_Drop_Public_Synonyms`: Drops the public synonyms created by using the `Optional_Create_Public_Synonyms` script.
- `Step99a_DropAll.sql`: Drops all the tables present in the Application Usage Manager database.

Enhancing Reporting Performance

Sentient (Usage Manager) provides several scripts and materialized views that can be applied to the Microsoft SQL Server and Oracle databases to enhance the reporting performance.

The scripts and views are available at:

- `Media\Usage\Optional Features\SQL Server` for Microsoft SQL Server database
- `Media\Usage\Optional Features\Oracle` for Oracle database

Using Views

There are two types of views, Standard Materialized Views and Filter Materialized Views. Both views enhance reporting performance. Either one can be optionally applied to a database. See the comments in the scripts for additional information about the functions of each view.

Note: The script names may abbreviate "Materialized" to "Mat", as in: `StepX_Define Filter Mat Tables and Indexes.sql`

Standard Materialized Views (SMV) - Converts all the views accessed by reports into tables. This view includes the index to enhance the query execution time. A feature where all the views (which is what the reports access) are converted into tables, and indexes are added to enhance the query speed.

Filtered Materialized Views (FMV) - Converts all the views accessed by reports into tables and requires filters to be applied before the views are converted into tables. The filters are stored in a separate table. For example, if a user selects `notepad.exe` as a filter, the FMV table is populated with the notepad details for all the devices. It is similar to SMV, but differs in that it requires filters to be applied at the time the views are converted into tables. The filters are stored in a separate table. As an example, if a filter for `Notepad.exe` is selected, the FMV table will be populated with only notepad details for all the devices.

Applying the scripts for SMV or FMV

1. Stop the service for the RCA **Knowledge Base Server**. The service may be stopped and started through the `Administrative Tools\Services` options of Windows Control

Panel.

2. Use normal procedures to execute the database scripts, in the given order, provided in the following locations:

- For SQL Server:

`\SQL Server\Optional Features\Filter Materialized Views`

Or

`\SQL Server\Optional Features\Standard Materialized Views`

- For Oracle:

`\Oracle\Optional Features\Filtered Materialized Views`

Or

`\Oracle\Optional Features\Standard Materialized Views`

Each of the above locations also includes a corresponding script to remove the view from your database. For example, the script name for the Microsoft SQL Server and Filtered Materialized Views is:

`SQLServer - Remove All Filter Mat Tables and Indexes.sql`

Utility Scripts

You as a database administrator can use the following scripts to enhance the reporting view performance:

- `Purge_Computer_Data.sql`: Deletes all data associated with the computer name. The computer name should be provided at the appropriate place in the script. The default value is MYCOMPUTER.
- `Purge_User_Data.sql`: Deletes all data associated with the computer name and the user name. The computer name and the user name should be provided at the appropriate place in the script. The default values are MYCOMPUTER and BOB.
- `Delete All Windows OS Files from Database.sql`: Deletes all Windows Operating System (OS) related files from the Usage Manager database.

Miscellaneous Scripts for Oracle

Miscellaneous scripts are additional scripts that can be applied along with the utility scripts to enhance the reporting view performance.

- `Optional_Create_Public_Synonyms.sql`: Creates public synonyms. The script may have to be edited for the Usage Manager's user names.
- `Optional_Drop_Public_Synonyms.sql`: Drops the public synonyms created by using the `Optional_Create_Public_Synonyms` script.
- `Step99a_DropAll.sql`: Drops all the tables present in the Usage Manager database.

Concept of Current Computer

The term Current Computer refers to the device that is currently in use or is active in the network. A Non-Current Computer refers to a device which no longer exists in the network for various reasons, such as, the device has been removed from the network or it has been renamed.

The Application Usage Manager database contains information on all current and non-current devices and makes it available for reporting whenever required. By default, the Reporting Server provides reports containing information on all the devices. If required, you can generate the reports with data only for the current devices using the concept of Current Computer. However, you can customize the content of the reports by categorizing the devices as Current Computers and Non-Current Computers.

The following command enables you to categorize the current and non-current devices in the database:

For Microsoft SQL Server and Oracle,

```
UPDATE rcaWindowsComputerUsers set CurrentComputer= 1 where  
ComputerName = 'MYComputerName'
```

The above command sets the **CurrentComputer** value to 1 for the device **MyComputerName** in the **rcaWindowsComputerUsers** table and signifies that the device is a current device. The value 0 in the **CurrentComputer** signifies that the device is a non-current device.

Note: The default value in the **CurrentComputer** column is 1. Therefore, all the devices in the network are considered current devices.

Instead of manually changing the **CurrentComputer** values in the table using the **UPDATE** command, you can use the **sp_FlagCurrentComputers** stored procedure to flag the device to 0 or 1, that is, current or non-current device based on the parameter passed to it.

For example, **sp_FlagCurrentComputers** (90, 30) procedure checks the last collection date of all the devices and marks the devices as current if the last collection date is within 30 days of time. If the last collection date is more than 90 days, it marks the device as non-current or not active device. The devices falling in between these two conditions remain unchanged.

Note: The **sp_FlagCurrentComputers** () procedure only flags the device to 0 or 1. It does not delete or purge the data associated with the device.

For more information on how to purge the data, see ["Optional Utility Scripts "](#) on page 315.

Aggregation of Usage Data Files with Tier-Level Store and Forward

The Aggregation of Application Usage Manager data files (USDBase files) with Tier-Level Store and Forward support is based on the methodology that the USDBase files pass through a set of existing upstream servers within a centralized network (Store and Forward Messaging Servers), which reduces the need to install multiple Usage Databases and KB Servers. The Store and Forward Messaging Servers accept the incoming USDBase files and forward them to another

upstream server. Upon receipt and prior to being forwarded, the USDBase files undergo an Aggregation process.

Making use of tiered store and forward servers with aggregation tools drastically improves scalability and reduces large-scale issues that can arise when dealing with large numbers of Application Usage Manager agents (Usage agents) across an enterprise. When using store and forward aggregation, a single KB Server can support much larger numbers of Usage agents, as well as show improved performance when supporting the same number of Usage agents.

Aggregation

Aggregation is a process of merging two or more USDBase files into a single USDBase by eliminating all the duplicate data among the USDBase files.

When the KB Server imports the usage data, it must query the database for each import data record to check for its existence, and then updates the database accordingly. If there are N number of files with common data to import, the KB Server has to query the database N number of times. If the data in these common files can be aggregated into a single file, the KB Server only needs to query the database and update it once. Thus, aggregating the data for common files prior to import improves the KB Server performance enormously.

An aggregated file contains only a single reference of a data for each file involved in aggregation. This reduces the size of an aggregated file and the overhead of multiple queries to the database when importing.

Implementation

The Messaging Server's Store and Forward capability is used to implement the Usage Data Files Store and Forward. The Messaging Server is responsible of storing all the USDBase files, trigger the aggregation process once the trigger constraint is satisfied, and forward the aggregated file.

In the non-Store and Forward scenario, all the Usage agents send their USDBase collection files to a predefined collection point that is monitored by KB Server which in turn extracts the usage data from the files collected from each agent computer and loads this data into your SQL-enabled database, making the data available for reporting purposes.

With Store and Forward, all the Usage agents send their USDBase collection files to a Messaging Server. You can have multiple such Messaging Servers in an enterprise to form a tier level architecture. The tier-level architecture should be implemented according to the number of Usage agents available in that enterprise to get better performance.

In an n-tier level architecture, the level 0 is the tier with only the Usage agents, level n is the final tier having the Portal installed with KB Server configured for importing USDBase files. All middle level tiers have Messaging Server with Aggregation enabled and may have Usage agents. Each middle level can have one or more Messaging Servers configured to forward the aggregated USDBase file to the next level Messaging Server.

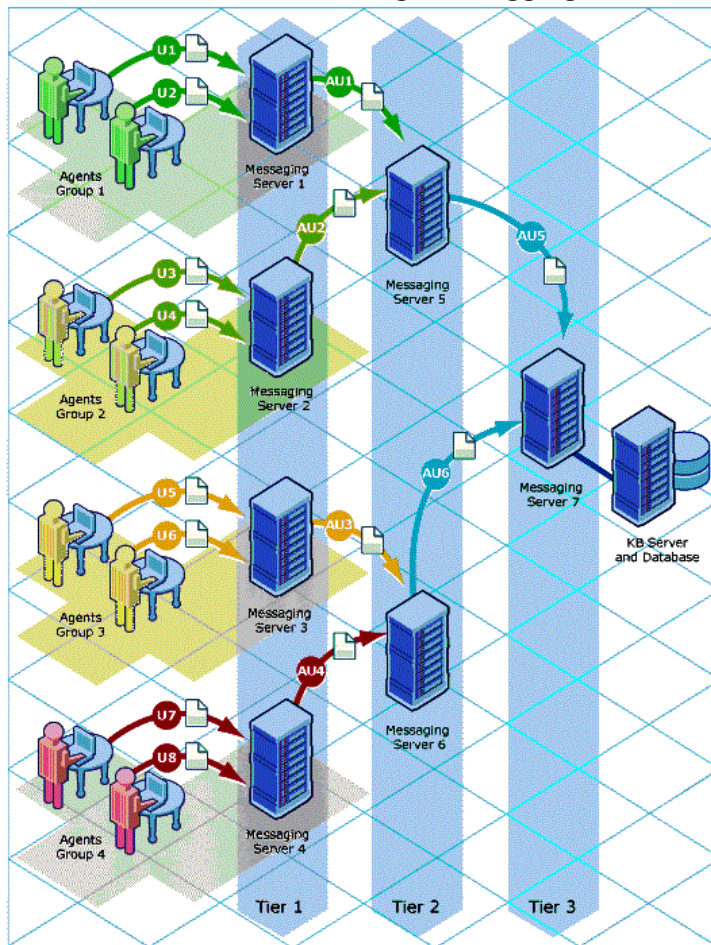
All the Usage agents must be configured to send their USDBase collection files to its immediate next level tier Messaging Server. This is the only change required in the Usage agent needed to implement Store and Forward Aggregation.

Once the Messaging Server in a middle tier receives the USDBase files, depending on the configurations set it triggers the Aggregation Process by executing the `USDBAggr.exe` which is

the tool for performing Aggregation. After the Aggregation Tool is successfully executed, the Messaging Server transfers the aggregated USDBase file to the next tier server. If the next tier is one of the middle level tiers, then the aggregation process continues till it reaches the final tier. If the next tier is the final tier, then the aggregated USDBase file gets imported into the database by the KB Server.

The following figure illustrates a multi-tiered Messaging Server architecture used to provide aggregated USDBase files for import into the usage database.

Tier-Level Architecture for Usage File Aggregation



Configuring Usage Data Files Store and Forward

You can configure your environment to store and forward usage data files to enhance the performance.

This section describes various sections and parameters in the `usage.dda.cfg` file and how to configure aggregation in your environment.

Configuring Aggregation

1. Configure the entries in the `usage.dda.cfg` file, as explained in the section "[About the Sections in the USAGE.DDA.CFG File](#)" below.
2. Following configuration, restart the **RCA Messaging Server** service.
You can have multiple tiers of Messaging Servers performing aggregation. The URL parameter in the **msg::register usage-forward** section of the `usage.dda.cfg` determines whether the aggregated usage data is forwarded to the next-tier Messaging Server, or to a KB Server import location.

About the Sections in the USAGE.DDA.CFG File

The `usage.dda.cfg` file in the Messaging Server defines the configurations available for Aggregation. The file has the following main sections after the header and the required lines.

```
# DO NOT REMOVE FOLLOWING LINE  
  
package require nvd.httpd
```

msg::register usagehttpd

This is the HTTPD Section Type for the `usage.dda` configuration. If the Messaging Server is receiving USDBase files from one or more Usage agents or another Messaging Server, define the URLHANDLER location on which to look for the files. This location is the source folder for the Aggregation Tool.

The parameters in the URLHANDLER are summarized below:

USE defines the name of AGGREGATE Type that receives the incoming file.

DIR defines the directory for the HTTPD Type.

URL defines the prefix that that is accepted by the Messaging Server. When messages are received with the designated URL they are deposited in the associated queue.

HTTPMETH defines the method used by the Usage agent to transfer the file. Set the HTTPMETH type to PUT.

msg::register usage-aggregate-queue-in

This is the AGGREGATE Type section for the `usage.dda` configuration. This defines the Aggregation Queue in which the USDBase files are queued up for processing by the aggregation tool.

DIR defines the source directory for the Aggregation Tool. This should be same as that of the HTTPD Type.

USE defines the name of the HTTP Type that forwards the aggregated file.

COUNT defines the minimum number of USDBase files expected in the DIR directory to initiate the Aggregation Process. The default is **400**. The Administrator can configure the COUNT.

POLL defines the interval, in seconds, used to poll the source DIR directory for USDBase files.

TIMEOUT defines the maximum interval to wait, in minutes, for the number of USDBase files to reach the COUNT.

USER_DEFINED is a flag to enable a user-defined procedure to execute whose return value controls the triggering of the Aggregation. Set to `Y` if a user-defined procedure is to be used. When set to `Y`, the COUNT and TIMEOUT parameter are ignored, and the UD_TCL_FILE and UD_PROC parameters must be used to define the location and procedure name to execute.

UD_TCL_FILE defines the fully-qualified path to the TCL file containing the user-defined procedure, when USER_DEFINED is set to `Y`.

UD_PROC defines the user-defined procedure name to execute, when USER-DEFINED is set to `Y`.

USDBAGGRPATH defines the full path of the Aggregation Tool.

TARGET defines the fully-qualified path to the directory where the aggregated USDBase file is placed.

ERRORPATH defines the fully-qualified path to the directory where the erroneous USDBase file is moved from the DIR directory.

ARCHIVE Set to `Y` to archive the USDBase file after aggregation; set to `N` to delete the USDBase file after aggregation.

ARCHIVEPATH defines the fully-qualified path of the directory where the USDBase file is placed when ARCHIVE is set to `Y`.

MAXSIZE defines the maximum size, in bytes, for the generated aggregated file.

Note: If the Input USDBase file size is larger than the MAXSIZE, the file is not aggregated and is sent without any modification.

LOGPATH defines the fully-qualified path of the directory where the log files are written by the aggregation tool.

msg::register usage-forward

This is the HTTP Type for the `usage.dda` configuration. Type is HTTPS if the target servers are SSL-enabled. This defines the forwarding server to which the aggregated file is sent.

PRI denotes the priority in which to send the files to the companion URL. The default Priority is **10**. This setting only matters if multiple ADDRESS entries are configured.

URL specifies the URL to use to send the aggregated file. This URL entry uses the same syntax as the COLLDEST for the Usage agents, discussed on "Configuring the Application Usage Manager Agent for Distribution" in the *Sentient Enterprise Console Online Help*.

- To send the aggregated USDBase file to a Portal that has the KB Server installed on it, code the URL as:
URL `http://Portal_IP:3471/KB_Mgr1_Usage/`
- To forward the aggregated USDBase file to the next tier Messaging Server enabled for the aggregation of usage data, code the URL as:
URL `http://messaging_svr_IP:3461/proc/usage`

Note: If the target servers are SSL-secured, adjust the URL syntax to include https and the secure port number.

Performance

You must fine-tune the configuration parameters in **usage.dda.cfg** to obtain the best aggregation performance. The main parameters that need tuning include:

MAXSIZE – This parameter directly impacts the compression ratio of an aggregated file, because MAXSIZE determines the number of files that can be placed in an aggregated file.

COUNT – This affects the MAXSIZE. Make sure the number of files in COUNT is sufficient enough to reach the aggregated file size set in MAXSIZE.

POLL – This parameter determines how often the files are collected from the source folder.

Application Usage Manager Agent

This section describes how to configure Application Usage Manager Agent for distribution. This chapter also describes various filters to be used to collect usage data, how to apply these filters, and how to initiate inventory and usage collection.

The USAGE Domain Defined

The Application Usage Manager utilizes the USAGE Domain within your Configuration Server Database, enabling management of the Application Usage Manager in the enterprise. The USAGE Domain is comprised of classes that you use to create Application Usage Manager services to distribute to your agent computers. These services install the Usage agent, which collects Application Usage data based on your specifications. The next few sections describe the USAGE Domain classes.

Use the CSDB Editor to configure each class in the USAGE Domain.

USAGE Domain Classes

Class	Description
Application (ZSERVICE)	Contains the Application Usage Manager services.
Application Packages (PACKAGE)	Contains the Application Usage Manager packages.
Client Methods (CMETHOD)	Client methods used to process each instance.
Collection (UMCOLLECT)	Controls the Application Usage Manager collection options that represent the unique collection criteria with which the USDBase collection files on each computer are associated. A single computer may have multiple collections targeted, each associated with a unique data store. The

Class	Description
	collection file contents are sent to different collection points and ultimately to different SQL databases.
Configuration (UMCONFIG)	Controls the Application Usage Manager installation options.
Database (UMDBASE)	Unique usage database name, which correlates to a backend SQL database.
Destination Point (UMDESTPT)	Location where Application Usage data is stored.
File Resources (FILE)	File instances included within a package.
File Root (FILEROOT)	Defines the base location of a file without any extended path information. The path can be a drive, like C:\, or a well known folder location, like ProgramFiles folder. File Roots are only used for filtering the collection file content collected from the agent in the CSDB Editor.
Filter Criteria (UMFLTCRI)	Defines usage filtering criteria.
Filter Rule (UMFLTRUL)	Connects to usage filtering criteria. The filter type determines whether it is an inclusion or exclusion Filter Rule, and its priority determines its importance when compared with other Filter Rules defined in a Filter Set. For more information, see "Filters" on the next page .
Filter Set (UMFLTSET)	Connects to one or more Filter Rules. Filter Sets are in turn connected to collections in the UMCOLLCT Class. Each UMCOLLCT Class may then have specific filtering associated with its data collection.
Inventory (UMINVENT)	Defines default configuration criteria for the Application Usage Manager inventory scan.
Path (PATH)	A unique path to one or more components.

Significance of Collection Instances

Collection class instances, and their related filter class instances, establish the content that is uploaded to a specific SQL database from the device which the Application Usage data is collected. Collection class instances define the database-specific agent collection properties; they also define the destination location once the usage and inventory data has been aggregated on the agent machine and is ready to be passed through the network for import into a specific SQL database.

Filters

By default, the Application Usage Manager collects usage information for every executable installed on the agent computer. To collect only specific executable information, filters can be defined and attached to the collection instance. Because all executable information is collected by default, the **Exclude All** filter rule must be included with any filters you use.

The USAGE Domain contains classes associated with creating filters for your usage data. This enables you to collect specific information based on parameters you define. The filter related classes are:

- Filter Criteria
- Filter Rule
- Filter Set

For more information about rule, criteria and set, see *Sentient Enterprise Console Online Help*.

Filter Criteria Class (UMFLTCRI)

The Filter Criteria Class contains various instances that reflect vendor-specific applications. Use these to attach a filter criterion using one of the included applications. The instances include filter criteria for Compaq, McAfee, Microsoft, and InstallShield, as well as many others.

Filter Rule Class (UMFLTRUL)

The Filter Rule Class contains pre-configured rules using common vendor-specific parameters. For example, Include Microsoft Office Products includes all Microsoft Office applications, and Exclude Microsoft Windows excludes all Microsoft Windows files.

This Filter Rule Class connects to the usage filtering criteria. The filter type determines whether it is an inclusion or exclusion Filter Rule and its priority determines its importance when compared with other Filter Rules in a Filter Set (Zero is the lowest priority). If the event priorities are equal within a Filter Set, inclusion filters take precedence over exclusion filters.

If no criteria are specified for an inclusion filter, then all usage information is captured in the collection file. If no criteria are specified for an exclusion filter, then all usage is excluded.

Filter Set Class (UMFLTSET)

A Filter Set is comprised of one, or a collection of, Filter Rules. The Filter Set class contains pre-configured Filter Sets you can use to collect or exclude vendor-specific information.

After a Filter Set has been created, or you decide which existing Filter Set to use, connect it to a Collection class (UMCOLLECT) instance to enable the filter. Only Filter Sets may be connected to a collection class. All criteria associated with that Filter Set are then processed. By connecting the Filter Set to a Collection class instance, each data collection can then contain specific filtering.

Default filter set parameters

Attribute	Default Value	Description
NAME	Default Filter Set Parameters	Filter Set Friendly Name
INCLUSAG	Yes	Determines whether or not to include usage data as well as inventory data in the collection.
CONCURR	No	Defines whether to collect concurrent usage data. For more information, see "Using Concurrency" in the <i>Sentient Enterprise Console Online Help</i> .
CMETHOD	CMETHOD.UMFLTSET	Client method used to process Filter Set class instances
FLTRUL##		Filter Rules connections

Using Filters

Filters can be applied to Collection class instances. Use the existing Filter Set instances, or create your own using the filter classes provided. The following exercises explain how to apply a filter as well as how to create a new filter based on parameters you define.

Filter Set Class Instances

The Filter Set Class (UMFLTSET) contains a few default Filter Sets that can be used for generic data collections. Each class instance collects a specific type or set of data. Following table describes each class instance in detail.

Filter set class (UMFLTSET) instances

Instance Name	Description
All Apps – Except Base OS Apps	Collects data for all installed applications with the exception of base operating system applications.
CCM Filter Set	Collects data for the Microsoft and Internet Explorer applications defined in the Filter Criteria connections.
Concurrency – Novadigm Apps Only	Collects concurrency data. This default setting collects data for all Sentient applications. Sentient applications were formerly known as Novadigm applications.
Default Filter Set Parameters	Contains the default Filter Set instance values and is used to create custom Filter Sets.
Inventory Only – Incl All Files	Collects inventory data for installed applications - no usage data is collected.
Microsoft Licensed	Collects data for Microsoft applications as defined in each Filter Criteria

Instance Name	Description
Applications	connection.
Usage – Include All Files	Collects all usage data.

Applying Filters

Filters can be applied to Collection class instances by simply dragging the Filter Set instance onto the appropriate Collection class instance.

Applying a Filter to a Collection Class

1. Start the Sentient Administrator CSDB Editor and double-click to expand the PRIMARY.USAGE class.
2. Connect the appropriate Filter Set to the connection class you would like to apply the filter to.
3. Click **Copy** and accept the changes.

Creating Filters

Filters can be created by following three basic steps:

1. Create or select a Filter Criteria instance.
2. Create or select a Filter Rule instance and attach the Filter Criteria instance.
3. Create or select a Filter Set instance and attach the Filter Rule instance.

The following example demonstrates how to create a filter to collect usage information for Adobe Acrobat.

Creating a Filter Criteria Instance

1. Start the Sentient Administrator CSDB Editor and double-click to expand the PRIMARY.USAGE class.
2. Double-click to expand **Filter Criteria (UMFLTCRI)**.
3. To create a new instance, right-click **Default Filter Criteria Params** and select **Copy Instance**, from the shortcut menu. The Copy Instance dialog box opens.
4. Rename the new instance **Adobe Acrobat**, and then click **OK**.
5. In the tree view of the CSDB Editor, double-click the newly created instance. The Editing Instance dialog box opens.

Use the attributes defined in the following table to define your filter criteria.

Filter Criteria Attributes

Attribute	Description
NAME	Friendly Name
ROOTCONN	Compares for applications that reside in any of the predefined shell folder names, such as ProgramFilesFolder, WindowsFolder, or SystemsFolder. These can be used to filter based on a well-known folder, such as TempFolder.
DESCRIPT	Description of filter criteria.
ENABLED	Whether this filter is enabled or not. [Y/N] Default is <code>Y</code> .
FILEROOT	Defines the file root path, or the base location of a file without any extended path information. Can be a drive, like <code>C:\</code> , or a well known folder location, like ProgramFiles folder.
FILEPATH	Defines the suffix of the path name that is appended to the file root path. For example, this would contain the characters <code>\Microsoft Office</code> regardless if the installation for the Microsoft Office application was to the <code>ProgramFilesFolder\Microsoft Office</code> path or to the <code>TempFilesFolder\Microsoft Office</code> path.
FILENAME	The application file name executable, for example <code>winword.exe</code> .
COMPNAME	The vendor name defined in the executable's header.
PRODNAME	The product name defined in the executable's header.
PRODVER	The product version defined in the executable's header.
FILEDESC	The file description defined in the executable's header.
FILEVER	The internal file version defined in the executable's header.
ORIGNAME	The original file name defined in the executable's header. This string does not change if the file is renamed.
MD5HASH	The MD5 hash file signature that uniquely identifies the contents of the file. Any change to a file assigns a unique MD5 hash signature to the file.
CMETHOD	The client method used to process Filter Criteria instances.

6. Double-click the **COMPNAME** attribute from the attributes list. The Editing Instance dialog box opens.
7. Type **Adobe** in the Company Name text box.
8. Click the **PRODNAME** attribute and type **Acrobat**.
9. Click **OK** and save the changes.

You have successfully created a new Filter Criteria instance.

Creating the Filter Rule Instance

1. Start the Sentient Administrator CSDB Editor and double-click to expand PRIMARY.USAGE class.
2. Double-click to expand the **Filter Rule (UMFLTRUL)** class.
3. Right-click **Default Filter Rule Parameters** and select **Copy Instance**, from the shortcut menu. The Copy Instance dialog box opens.
4. Rename the new Filter Rule instance **Adobe Apps**, and click **OK**.
5. In the CSDB Editor window, connect the Filter Criteria instance you created, Adobe Acrobat, to the newly created Adobe Apps Filter Rule instance.
6. Click **Copy**.
7. Click **Yes** to accept the changes.
8. Click **OK**. The Filter Criteria instance is now connected to a Filter Rule instance.
9. In the list view of the CSDB Editor, double-click the PRIORITY instance attribute and set the value to 1 to ensure this rule is applied before any default rules are used. (Zero is the lowest possible priority as well as the default value.)

The Filter Rule instance is now complete.

Creating the Filter Set Instance

1. Start the Sentient Administrator CSDB Editor and double-click to expand PRIMARY.USAGE class.
2. Double-click to expand the **Filter Set (UMFLTSET)**.
3. Right-click **Default Filter Set Parameters** and select **Copy Instance**, from the shortcut menu. The Copy Instance dialog box opens.
4. Rename the new Filter Set instance `Adobe Licensed Applications`.
5. Connect the Filter Rule instance you created earlier, Adobe Apps, to the newly created Adobe Licensed Applications Filter Set instance.
6. Also connect the existing Filter Rule instance, Exclude All, to the newly created Filter Set instance. This ensures that only usage data for the specific executable defined in your criteria, Adobe Acrobat, is collected.
By default the priority setting for this instance is 0, the lowest possible. By setting the priority for the Adobe Apps rule to 1 earlier, you have given priority to that Filter Rule instance, ensuring it is applied before any default rules.

The Filter Set instance is complete. You are now ready to apply the filter to a collection instance. For more information, see the section "[Applying Filters](#)" on page 327.

Handling Renamed Device

In Application Usage Manager, the Reporting Server provides all usage and inventory data report based on the name of a device. Whenever a device is renamed, the data about the device is stored twice in the database, with the old name and the new name. The Reporting Server treats the renamed device as a new device and gives information about both the devices.

For example, device A is renamed to device B, the Reporting Server treats device B as a new device and shows data associated with device A as well as with device B, which are same devices.

The Application Usage Manager uses the concept of Current Computer to avoid duplication of data. In the concept of Current Computer, the renamed device is treated as current or active device, and the data associated only with the current device is reported.

For more information, see ["Concept of Current Computer" on page 318](#).

During the first collection from the client device performed after renaming the device, the Usage Agent sends usage and inventory information with the both old and new device names. This occurs because the process of starting the operating system and renaming the device results in executables being launched under the old and the new names of the device.

After the first collection, you must update CurrentComputer column to 0 for the old device and then for subsequent collections, the Usage Agent sends the usage and inventory information of the renamed device only. The devices in the CurrentComputer column can be flagged to 0 or 1 using the Concept of Current Computer.

Note: On the client device, the `History.USDBase` file has both old and new device information. However, the history purge deletes the data whenever it qualifies for a history purge as scheduled.

Internet Explorer Usage Counts for Windows Vista

When Internet Explorer is running on Windows Vista® (x64) with the Enable Protected Mode selected, (click **Tools > Internet Options > Security menu** to select Enable Protected Mode) each launch of the application results in usage count of two.

This is because when the user launches Internet Explorer, it first runs `IEUser.exe`, which runs at the default medium-integrity level. `IEUser.exe` then spawns a low-integrity level executable, `IEExplorer.exe`, which is the program the user sees and interacts with. Therefore, the usage count will always be two for each launch of Internet Explorer.

Setting Search Criteria using AUM Administrator

Use the Application Usage Manager (AUM) Administrator to create specific search criteria to be implemented when you are generating your usage monitoring reports. Creating these criteria enable you to supplement the existing search options and create better reports based on your individual organization's requirements. Note that Sentient Application Usage Manager Administrator is available as a separate installer and is not installed with Sentient Application Usage Manager. For more information, see *Sentient Enterprise Installation and Upgrade Guide*.

Accessing the Application Usage Manager Administrator

1. From the **Start** menu, go to **Sentient Enterprise Application Usage Manager > Application Usage Manager Admin.**
2. Select the name of the DSN that you will use and type your User Name and Password in the text boxes provided.
3. Click **OK**.

The Application Usage Manager Admin consists of four tabs you can use to define criteria, rules, rule sets, and rule set groups.

Application Usage Manager Admin Search Function

After selecting criteria, rule, rule set, or rule set group, click the **Search** button to preview your query results. Query results are displayed in a table at the bottom of the Application Usage Manager Admin window.

The search function can be used at any stage of the rule creation process.

Creating Criteria, Rules, Rule Sets, and Rule Set Groups

Use the Application Usage Manager Admin to create criteria, rules, rule sets, and rule set groups that are used to generate usage reports.

Operators AND versus OR

There are two types of criteria, rules, rule sets, and rule set groups you can create: AND and OR. Before creating the rule, decide which type you would like to create and select the appropriate operator from the **OPERATOR** list.

Creating a criterion using the AND operator specifies that in order for a record to match that criterion, all of the properties specified must be true. For example, a criterion designed with

Vendor Property	= Equals Microsoft
Application property	= Like WinWord
Operator	= AND

will return only Microsoft Word records.

If the OR operator is selected in the above example, all applications with Vendor Microsoft are returned.

The AND operator is most effective when creating criteria only. The OR operator is more appropriate for creating rules, rule sets, and rule set groups.

Criteria Tab

Use the Criteria tab to define specific application criteria you use to display collected information when generating usage monitoring reports.

Before you select application criteria, use the Criteria list to check if any of the existing applications can be used.

After you select an application from this list, select a **Property** check box to add that to the Criteria.

Double-click any row in the **Condition** or **Value** columns to add a condition and value to the criteria. As an example, the following steps describe how to create rule criteria where the Vendor property is equal to Microsoft.

Creating a Criterion

1. To the right of the Criteria box, click the **Add** button to create new criteria. The Add Criteria dialog box opens.
2. Type a name for the criteria in the Add Criteria dialog box.

Note: Names are sorted in the report pages in ascending sequence. Therefore, frequently used names should be prefixed with a character that places them at the top of the sort sequence.

3. Click **OK** to close the Add Criteria dialog box.
4. Double-click the Vendor attribute row. The Criteria:Vendor dialog box opens.
5. Select **Equals** in the **Condition** list.
6. In the **Value** box, type **Microsoft**. Note that this value is used in a SQL command and must conform to SQL syntax rules.

Note: The typed text must conform to SQL Server query rules. For example, you can select the **LIKE** clause and type text such as `%Microsoft%` to define a criterion for any application whose Vendor definition contains the character string Microsoft.

Note: The use of LIKE clauses with preceding % may cause lengthy search times during reporting. It is recommended to use the EQUALS clause.

7. Test the criteria by clicking **Search** to retrieve all entries in the Application Usage Manager Knowledge Base that match the criteria you defined. Entries are displayed in the table at the bottom of the window.

The new criterion is complete and ready to be used in any rules you generate.

Rules Tab

Rules are a combination of Criteria. Use the Rules tab to define rules based on the criteria you selected in the Criteria tab.

Use the Rules list to select any default rules. After a rule is selected, the criteria that are part of that rule are displayed.

Creating a New Rule

1. Click **Add** to define a new rule. The Add Rule dialog box opens.
2. Type a name for the rule in the **Add Rule** dialog box.
3. Click **OK**.
4. Select any criteria you would like to include in the new rule by clicking the check box to the left of the criterion name.
5. Test the rule by clicking **Search**. All matching records are displayed in the table at the bottom of the window.

The new rule is complete and ready for inclusion in any Rule Sets you generate.

Rule Sets Tab

A Rule Set is a grouping of Rules. Use the Rule Sets tab to define which rules you would like to combine to form a Rule Set instance.

Use the Rule Sets list to select any existing Rule Sets. After a Rule Set is selected, the rules that make up that Rule Set are displayed.

Creating a New Rule Set

1. Click **Add** to define a new Rule Set. The Add Rule Set dialog box opens.
2. Type a name for the Rule Set in the **Add Rule Set** dialog box.
3. Click **OK**.
4. Select any rules you would like to include in the new Rule Set by clicking the check box to the left of the rule name.
5. Test the Rule Set by clicking **Search**. All matching records are displayed in the table at the bottom of the Application Usage Manager Admin window.

The new Rule Set is complete and ready for inclusion in any Rule Set Groups you may generate.

Rule Set Groups Tab

If you want to combine multiple Rule Sets, create a Rule Set Group instance.

Use the Rule Set Groups list to select any existing Rule Set Groups. After a Rule Set Group is selected, all of the Rules Sets that make up that Rule Set Group are displayed.

Creating a New Rule Set Group

1. Click **Add** to define a new Rule Set Group. The Add Rule Set dialog box opens.
2. Type a name for the Rule Set Group in the **Add Rule Set** dialog box.
3. Click **OK**.
4. Select any Rule Sets you would like to include in the new Rule Set Group by clicking the check box to the left of the Rule Set name.
5. Test the Rule Set Group by clicking **Search**. All matching records are displayed in the table at the bottom of the Application Usage Manager Admin window.

The new Rule Set Group is complete.

Now that you have finished creating criteria, rules, rule sets, and rule set groups, you can generate reports based on your own specifications using the report generator options Include Rule and Exclude Rule. For more information, see *Viewing Usage Reports in the Sentient Enterprise Console Online Help*.

Chapter 7

Security and Compliance Management

The Security and Compliance Management features in Sentient enable you to monitor and manage security vulnerabilities, configuration compliance, and security tool performance across your environment.

Vulnerability Management

Vulnerability management is the process of identifying, locating, and rectifying software security and vulnerability issues in the enterprise. There are three main steps in this process:

1. Obtain updated vulnerability definitions and scanner.
2. Scan the managed devices in the enterprise for the presence of vulnerabilities.
3. Report the vulnerability assessment of the scanned devices, including summary information for the enterprise as a whole.

The following terms are used throughout the Sentient vulnerability management solution:

Vulnerability Management Terms

Term	Definition
vulnerability	A weakness in a system, its configuration, or its software that allows an individual to compromise the system's integrity to gain unauthorized access to its resources.
exposure	Exposure can refer to a measurement of the various vulnerabilities in an environment. It can also be used to refer to a piece of software that provides information or capabilities that a hacker might use to attack or exploit a system.
CVE	Common Vulnerabilities and Exposures The CVE is a dictionary of common names (CVE Identifiers) for publicly known information security vulnerabilities and exposures. The CVE was started in 1999. It is currently sponsored by the United States Department of Homeland Security and managed by the MITRE Corporation. For more information, see http://cve.mitre.org
NVD	National Vulnerability Database The NVD is the U.S. government repository of standards based vulnerability management data. This data enables automation of vulnerability management, security measurement, and compliance. For more information, see http://nvd.nist.gov
CVSS	Common Vulnerability Scoring System The CVSS is a standard severity scoring system for information security vulnerabilities. CVSS includes three groups of metrics: • Base

Term	Definition
	• Temporal • Environmental For more information, see http://www.first.org/cvss
OVAl	Open Vulnerability and Assessment Language OVAl is the standard used to encode and transmit security information and system details. It is based on three XML schemas that represent the three security vulnerability assessment process steps: representing system configuration, expressing a specific machine state, and reporting the results of the assessment. The purpose of the CVE is to catalog all known vulnerabilities. The purpose of OVAl is to describe how to identify specific vulnerabilities. Most OVAl definitions are based on a CVE, but some are not. Accelerite Content Network transmits information in OVAl and CVE format to Sentient. For more information, see http://oval.mitre.org/

Compliance Management

Compliance management is the process of identifying, locating, and rectifying software configuration problems on managed client devices in the enterprise. There are three main steps in this process:

- Obtain updated compliance benchmarks and scanner.
- Scan the managed client devices in the enterprise to determine whether their configuration is in or out of compliance with the pertinent policy or regulatory standard defined by the compliance benchmarks.
- Report the results of the compliance scans, including summary information for the enterprise as a whole.

At this point, the administrator can take steps to resolve any configuration issues identified.

The following terms are used throughout the Sentient compliance management solution:

Compliance Management Terms

Term	Definition
CCE	Common Configuration Enumeration The CCE is a dictionary of names for software security configuration issues (for example, access control settings and password policy settings). By providing unique identifiers for system configuration issues, the CCE facilitates fast and accurate correlation of configuration data across multiple information sources and tools. The CCE is currently managed by the MITRE Corporation. For more information, see http://cce.mitre.org
CCI	The CCI provides a standard identifier and description for each of the

Term	Definition
(Control Correlation Identifier)	singular, actionable statements that comprise an IA control or IA best practice. CCI bridges the gap between high-level policy expressions and low-level technical implementations. It allows a security requirement that is expressed in a high-level policy framework to be decomposed and explicitly associated with the low-level security setting(s) that must be assessed to determine compliance with the objectives of that specific security control. The ability to track security standards from their roots to their low-level implementation (for example, laws, IA frameworks) helps organisations to effectively demonstrate compliance with various IA compliance frameworks. CCI also offers a way of objectively rolling out and evaluating the outcomes of the associated enforcement evaluation through various technologies. For more information, see https://public.cyber.mil/stigs/cci/cci.mil/stigs/cci/cci/
FDCC (Federal Desktop Core Configuration)	The FDCC is a security configuration mandated by the Office of Management and Budget (OMB) for all U.S. government agencies. For more information, see https://www.nist.gov/programs-projects/federal-desktop-core-configuration-fdcc
USGCB (The United States Government Configuration Baseline)	The purpose of the USGCB initiative is to create security configuration baselines for Information Technology products widely deployed across the federal agencies. The USGCB baseline has evolved from the FDCC mandate. The USGCB is a Federal government-wide initiative that provides guidance to agencies on what should be done to improve and maintain an effective configuration settings focusing primarily on security. The USGCB currently exists for Microsoft Windows XP, Windows Vista, and Windows 7 operating system software. These recommendations were developed at the National Institute of Standards and Technology (NIST), which collaborated with DoD and Microsoft to produce the Windows 7, Windows 7 Firewall, Internet Explorer 8 USGCB. There are also USGCB benchmarks for Windows 7 Firewall, and Internet Explorer 8. For more information, see http://usgcb.nist.gov/index.html.
SCAP	Security Content Automation Protocol (pronounced ess-kap) SCAP is a framework of interoperable and automatable security standards established by the National Institute of Standards and Technology (NIST). SCAP enables organizations to automate security monitoring, vulnerability management, and security policy compliance evaluation. SCAP incorporates the following specifications: • CVE (see "Vulnerability Management" on page 335) • CCE (see "Compliance Management" on the previous page) • Common Platform Enumeration (CPE), a naming convention for

Term	Definition
	hardware, operating system (OS), and application products - Extensible Configuration Checklist Description Format (XCCDF), an XML specification for structured collections of security configuration rules used by OS and application platforms - OVAL (see "Vulnerability Management" on page 335) - CVSS (see "Vulnerability Management" on page 335) Because SCAP uses XML-based standards, SCAP content is both human and machine-readable. NIST provides SCAP content, such as vulnerability and product enumeration identifiers, through a repository supplied by the National Vulnerability Database (NVD). For more information, see http://nvd.nist.gov/scap.cfm
CIS (Center for Internet Security)	The CIS developed a set of compliance standards before the time that NIST created SCAP. As of the publication of this documentation, the CIS had not released any additional benchmarks for newer operating systems. The Accelerite Content Network team provides CIS benchmarks in SCAP format to Accelerite Content Network content subscribers. For more information, see http://cisecurity.org.
DISA (Defence Information Systems Agency)	DISA is a U.S combat support agency that connects the U.S military and government through IT and communications support. It is originally known as the defense communications industry (DCA). It was created in 1960, partially in response to communication issues during WWII. Currently, the DISA serves U.S soldiers globally, allowing them to access crucial information worldwide through services like computer hosting, information assurance, multinational information sharing, and data services. DISA's information systems are used across the full spectrum of military force operations including offensive tactics, defensive tactics, counter terrorism, and humanitarian efforts. The main purpose of DISA is achieving information dominance by providing an effective and efficient enterprise infrastructure that can provide the user an advantage in combat. For more information, see https://www.disa.mil/

A group of related compliance requirements is known as a **Benchmark**. Benchmarks can be revised. A benchmark is given a new version name whenever it is revised (for example, MS-Windows-10-V1R18-STIG-SCAP-1-2-Benchmark).

Benchmarks contain **rules**. Each rule includes one or more automated tests that are used to determine if a client device meets the requirements specified by that rule.

A benchmark consists of one or more **profiles**, which are used to define different levels of compliance within that benchmark. A profile specifies the following:

- A set of rules in the benchmark (possibly all of them)
- For each rule, the value that determines compliance against that rule

Compliance with a rule is determined by the profile. When Sentient runs a compliance scan on a managed client device, it evaluates the requirements for the applicable benchmark profile.

Each UGSCB benchmark contains a single profile. The DISA benchmark contains separate profiles for different types of systems.

Each rule is assigned a **weight** based on the potential effect and exposure to the enterprise if client devices do not comply with that rule. When a compliance scan is performed on a managed client device, a **score** is determined that reflects how many compliance rules passed and failed. This score represents a device's compliance with respect to a particular benchmark profile (SCAP checklist).

Note: In certain compliance reports and dashboards, compliance results for a particular benchmark are aggregated across all profiles that pertain to each managed client device. See "Compliance Management Reports" and the "Compliance Management Dashboard" in the *Sentient Enterprise Console Online Help* for more information.

The benchmark, profiles, and rules are all delivered as a bundle of files known as a SCAP **datastream**. These files are read by SCAP-capable tools, such as the Sentient compliance scanner.

Security Tools Management

Sentient scans the managed client devices in your enterprise to determine what security tools are present and to collect pertinent information about the products detected. The following types of security tools are supported:

- Anti-virus tools
- Anti-spyware tools
- Software firewalls

Sentient Security Tools Management (STM) scanner contains embedded knowledge about various security products that it can detect. The Sentient STM scanner enables Sentient to determine which specific security products are installed, which are enabled, when the most recent anti-virus and anti-spyware scan was performed on each client device, and when virus and spyware definitions were most recently updated on the client devices. The collected information is then displayed in the Security Tools Management dashboard and related reports.

To view the list of products currently supported by the Sentient STM scanner, see the Security Tools Management update for the current release on Accelerite Content Network. For more information, contact Persistent Systems Support.

It is recommended to subscribe to the Accelerite Content Network email notifications to get the latest updates.

Note: The STM scanner detects and provides detailed information for the products listed on the above URL. For other products, the STM scanner detects and provides the following

basic information: protection status, product name, product version, and vendor name.

How Accelerite Content Network Content is Updated

Accelerite Content Network provides two types of security and compliance management content:

- Data – vulnerability definitions and SCAP data
- Scanners – a vulnerability scanner, a compliance scanner, and a security tools management scanner

In order to access the Accelerite Content Network content, see "Accelerite Content Network " in the *Sentient Enterprise Console Online Help*.

When you update your Sentient security and compliance management content – either from Accelerite Content Network or from the file system – the following three things happen:

1. Both the updated scanners and data are copied into a temporary directory.
2. The data is pushed from the temporary directory to the Core database. This drives the detailed definition reports and primes the database for processing the collected scan results.
3. Both the data and scanners are loaded into the CSDB.

When a client device with a configured security policy subsequently makes a connection to the SECURITY Domain in the CSDB, the data and scanners are deployed to that client device. At this point, the client device will be scanned. The results of the scans are then sent to the Core database.

The diagram illustrates the Accelerite Content Network architecture. It shows the flow of security and compliance content from independent vendors and the National Vulnerability Database (NVD) through a central network to end-user devices. The process is numbered 1 through 3:

- 1**: Content is received from Independent Vendors and the National Vulnerability Database (NVD) into the Accelerite Content Network.
- 2**: Content is processed and stored in the Core Database and Configuration Server Database.
- 3**: Content is distributed to the Core Console, which then delivers it to end-user devices (represented by computer icons).

- The security and compliance content that is loaded into the CSDB includes both “service” definitions and “master” definitions. The service definitions are related to the scanning services and are deployed to the platform-specific agents for performing the scans. The master definitions are used when you move content from a test environment to a production environment (see ["Move Accelerite Content Network Content from a Test Environment to a Production Environment"](#) on page 366).

For compliance scanning, the master definitions include the compliance benchmarks in SCAP format.

Radia Client Automation Reference Guide (11.00)

anti-virus and anti-spyware tools, the scanner also determines when each tool last updated its definitions and when it last performed a full system scan.

When you install Sentient, some security services are available by default. Following are the security services that become available after you perform an Accelerite Content Network update.

- **Discover and Manage All Security Tools:**

- ZSERVICE.HP_SECTOOLS_MGMT_ALL_T- This template can be used to scan security tools management (STM) tools
- ZSERVICE.HP_SECTOOLS_MGMT_ALL_V001- This template can be used to create customised profile for scan and remediate STM tools

These security tools enable you to create security profiles that can be applied on all or on some of the selected security tools (AV, AS, FW). They also provide options to scan information on managed devices. The default profile scans all the security tools on the managed devices. You can either enable or disable tools based on the Product Name, Vendor, or Version. You can also initiate an update or scan.

- **Anti-Spyware Security Tools Management:**

- ZSERVICE.HP_SECTOOLS_MGMT_AS_T- This template can be used to scan STM tools
- ZSERVICE.HP_SECTOOLS_MGMT_AS_V001- This template can be used to create customised profile for scan and remediate STM tools

You can create security profiles from these templates that can be applied only to the anti-spyware security tools. They provide options to scan the information to enable or disable products based on the Product Name, Vendor, or Version. For example, Enable all McAfee Tools and Disable Symantec Security Tool. Options are also available to initiate an update (of definition file) or scan. The default profile scans for the anti-spyware tools on the client machine.

- **Anti-Virus Security Tools Management:**

- ZSERVICE.HP_SECTOOLS_MGMT_AV_T- This template can be used to scan STM tools
- ZSERVICE.HP_SECTOOLS_MGMT_AV_V001- This template can be used to create customised profile for scan and remediate STM tools

You can create security profiles from these templates that can be applied only to the anti-virus security tools. They provide options to scan the information to enable or disable products based on the Product Name, Vendor, or Version. For example, Enable all McAfee Tools and Disable Symantec Security Tool. Options are also available to initiate an update (of definition file) or scan. The default profile scans the anti-virus tools on the client machine.

- **Client Firewall Security Tools Management:**

- ZSERVICE.HP_SECTOOLS_MGMT_FW_T- This template can be used to scan STM tools
- ZSERVICE.HP_SECTOOLS_MGMT_FW_V001- This template can be used to create customised profile for scan and remediate STM tools

You can create security profiles from these templates that can be applied only to the firewall tools. They provide options to scan the information to enable or disable products based on the Product Name, Vendor, or Version. For example, Enable all McAfee Tools and Disable Symantec Security Tool. Options are also available to initiate an update (of definition file) or scan. The default profile scans the firewall tools on the client machine.

Similarly for vulnerability management, following are the tools:

- ZSERVICE.HP_VULN_MGMT_ALL_T (<Vulnerability Management>)- This is a template-based ZSERVICE.
- ZSERVICE.HP_VULN_MGMT_ALL_V001 (<Sample Discover Vulnerabilities>)

These services scan the agent devices for vulnerabilities and provide the results of the scan to RCS (for eventual routing to RRS). It finds vulnerabilities using definitions from the National Vulnerability Database (NVD).

Note: NVD is updated on a yearly basis.

Adding new Products for Security Management

STM scanner uses the STMLibrary, an XML file that contains an entry for each security product supported by STM scanner. Each entry defines the product detection and data collection logic for the corresponding product. For the STM scanner to support new product, you need to add an entry for the corresponding security product into the STMLibrary.xml under the appropriate section.

The STM Library is updated periodically or on demand whenever a new security product support is required and uploaded to the Accelerite Content Network. The STM Library is updated automatically in your Sentient environment when you connect to Accelerite Content Network. You can use the information given in this section and customize the STMLibrary.xml file if you have an urgent need to add new security products.

Customize the STMLibrary.xml File

Perform the following steps to customize the STMLibrary.xml file:

1. Open the Sentient Administrator CSDB Editor window. The default user name and password to open the CSDB Editor window are admin and secret respectively.
2. Expand **Primary -> Security -> Application Packages**.
3. Expand **Security Tools Scanner for Windows -> Security Tools Scanner for Windows: <all>**.
4. Right-click the **Security Tools Scanner for Windows\security\sectools\scanner\STMLibrary.xml** option and select the **Edit this Component** option from the popup menu.
5. Select the notepad.exe and browse to any text editor to open the STMLibrary.xml file in a text editor. Click **OK**.
6. Edit the STMLibrary.xml file as per your requirement and save it to a temp location. Refer to the ["STM Library Schema" on the next page](#) and ["Sample STMLibrary.xml" on page 354](#) sections to know more about STM Library components.

Replace the STMLibrary.xml File

After you make the required changes in the STMLibrary.xml file, you must replace the existing STMLibrary.xml file in the CSDB:

Perform the following steps to replace the STMLibrary.xml file:

1. Open the Sentient Administrator CSDB Editor window. The default user name and password to open the CSDB Editor window are admin and secret respectively.
2. Expand **Primary -> Security -> Application Packages**.
3. Expand **Security Tools Scanner for Windows -> Security Tools Scanner for Windows: <all>**.
4. Right-click the **Security Tools Scanner for Windows\security\sectools\scanner\STMLibrary.xml** option and select the **Replace Component Data** option from the popup menu.
5. Browse to your customized STMLibrary.xml file saved at temp location.
6. Click **Yes** to confirm the file replacement.

STM Library Schema

STM Library is an XML file, STMLibrary.xml, that contains an entry for each security product supported by STM scanner. Each entry defines the product detection and data collection logic for the corresponding product. For the STM scanner to support new products, you need to add an entry for the corresponding security product into the STMLibrary under the appropriate section.

The top level tag in the STMLibrary is <SecurityApplication>. The <SecurityApplication> tag contains three security types tags as follows each one for the supported security tools:

- <AntiViruses>
- <AntiSpywares>
- <Firewalls>

The security type tags listed above contain a respective product tag as listed below.

- <AntiVirus>
- <AntiSpyware>
- <Firewall>

The security type tags can contain multiple product tags. Each product tag represents a security product. The table below list various attributes of the product tags.

Attribute	Description
Name	Specifies the name of the product
Verison	Specifies the version of the product
WMIName	Specifies the name displayed in the WMI explorer for the product

Product Tags Definitions

The product tags contain the following definitions:

- **<IsInstalled>**: Detects whether a product is installed or not on a machine.
- **<Detection>**: Contains sub tags for the every piece of information that is gathered from machine if the product is installed on the machine.
- **<Remediation>**: Contains sub tags to remediate the product if found vulnerable.

The sub tags list for anti-virus and anti-spyware products are same. However, the sub tags for firewall products are different.

Detection sub tags for anti-virus and anti-spyware

The table below lists the sub tags used in the **<Detection>** definition for the **<AntiVirus>** and **<AntiSpyware>** product tags.

Sub tag	Description
<ProductName>	Name of the product
<ProductVersion>	Version of the product
<VendorName>	Vendor name
<DataFileDirectory>	Location where the definition files are stored
<DefinitionDate>	The latest definition date
<DefinitionVersion>	The latest definition version
<DefinitionVersion>	Version of the engine
<InstallDirectory>	Location where the product is installed on the machine
<IsFileSystemProtectionStatusEnforced>	Is the security product currently turned on
<IsFullSystemScanInProgress >	Is the scan currently running
<IsUpdateInProgress>	Is the update currently running
<LastFullSystemScan>	The last time of the full system scan

Remediation sub tags for anti-virus and anti-spyware

The table below lists the sub tags used in the **<Remediation>** definition for the **<AntiVirus>** and **<AntiSpyware>** product tags.

Sub tag	Description
<SystemProtectionTurnOn >	Turn on the product , if turned off

<SystemProtectionTurnOff>	Turn off the product, if turned on
<RunFullSystemScan>	Run the full system scan
<RunDefinitionUpdate>	Run the security product update

Detection sub tags for firewall

The table below lists the sub tags used in the <Detection> definition for the <Firewall> product tag.

Child tag	Description
<ProductName>	Name of the product
<ProductVersion>	Version of the product
<VendorName>	Vendor name
<Policies>	Rules defined for the firewall
<InstallDirectory>	Location where the product is installed on the machine
<IsEnabled>	Is firewall turned on
<IsProductAuthentic/>	Is the product authentic

Remediation sub tags for firewall

The table below lists the sub tags used in the <Remediation> definition for the <Firewall> product tags.

Child tag	Description
<FirewallProtectionTurnOn>	Turn on the firewall
<FirewallProtectionTurnOff>	Turn off the firewall

Operation Tags

The product tag definitions make use of the operation tags and logical tags to perform data collection or remediation activities. For example, the Registry operation tag retrieves and captures the name for a security product from registry.

Note: Whenever the Name attribute appears in an operation tag, the value defined for the Name attribute is considered as substitution variable. This substitution variable is scoped inside the product tag, which means a substitution variable for a particular product cannot be used for another product. To reference the substitution variable, the value defined for the corresponding Name attribute should be placed within the %% symbols. In addition to the substitution variable, the %% can be used to reference the value of system environment variables and Detection sub tags.

The STMLibrary.xml makes use of the following operation tags.

<Registry>

The Registry tag is used to perform read and write operations on Windows registry. The table below lists the attributes of the Registry tag and the possible values.

Attribute	Possible Value	Description
Name	Customer defined value	A variable to store the value data retrieved from registry. The Name attribute is used in conjunction with the Evaluate attribute. If the Evaluate attribute is used, the value retrieved from the registry is stored in the variable you defined in the Name attribute. Else, the value is not stored in the variable you defined in the Name attribute.
Verb	Exists	Used to check the presence of the particular registry key under the hive. If attribute is not specified, returns the value of the registry key. This is an optional attribute.
Key	Any key defined by the vendor	Represents a value name from registry
Reg	Complete path of registry hive	Complete path of registry hive
Arch	32 64 32 64 64 32	Read 32-bit registry Read 64-bit registry Read 32 bit registry and if value not found , read 64 bit Read 64 bit registry and if value not found , read 32 bit
FormatType	getDirName	Return the absolute path till the parent directory, if the value from registry is an absolute path.
Evaluate	False	Used along with the Name attribute. When used, the value retrieved from the registry is stored in the variable you specified in the Name attribute.
Type	REG_DWORD REG_SZ REG_BINARY	Read or write the DWORD value from/to registry Read or write the string value from/to registry Read or write the binary value from/to registry
Value	Any REG_DWORD/REG_SZ/REG_BINARY value	If used, signifies the write operation.

Example:

```
<Registry Name="EngineVersionMajor" Reg="HKEY_LOCAL_
MACHINE\SOFTWARE\McAfee\AVEngine" Key="EngineVersionMajor" Type="REG_
DWORD" Evaluate="false" />
```

<Process>

The Process tag is used to perform process related activities, such as creation of process and check whether the process is running. The table below lists the attributes of the Process tag and the possible values.

Attribute	Possible Values	Description
Path	Path of the executable	Specifies the path where the executable of the process resides.
Verb	Check	Checks whether the process is running
	Create	Creates the process
ProcessName	Process Name	Name of the process
CommandLine	Process Command Line	Command line of the process
WindowTitle	Window title	Window Title of the process, if it has a window

Example:

```
<Process Verb="Check" ProcessName="ccSvcHst.exe" WindowTitle="LiveUpdate
Status"/>
```

<RunCommand>

The RunCommand tag is used to execute commands that are executed by the vendor. The table below lists the attributes of the RunCommand tag and the possible values.

Attribute	Possible Values	Description
Name	Customer defined value	A variable to store the value returned upon command execution. The Name attribute is used in conjunction with the Evaluate attribute. If the Evaluate attribute is used, the output of the command execution is stored in the variable you defined in the Name attribute. Else, the output is not stored in the variable you defined in the Name attribute.
Path	Absolute path to the command	Specifies the absolute path to the command location

	executable	
Command	Command name	Specifies the command to be executed
Args	Command arguments	Arguments to be passed along with the command, if any
Evaluate	False	Used along with the Name attribute. When used, the output of the command execution is stored in the variable you specified in the Name attribute.

Example:

```
<RunCommand Command="netsh" Args = "advfirewall set allprofiles state on"
Path = "%InstallDirectory%" />
```

<FileProperties>

The FileProperties tag is use to read the file property values. The table below lists the attributes of the FileProperties tag and the possible values.

Attribute	Possible Values	Description
Path	Path of the file	Absolute path of file to which property value is to be retrieved

Value	Comments	Returns the comments specified in the file property
	InternalName	Returns the Internal name of the file specified in the file property
	ProductName	Returns the Product name specified in the file property
	CompanyName	Returns the Company name specified in the file property
	LegalCopyright	Returns the copy right information specified in the file property
	ProductVersion	Returns the version of product specified in the file property
	FileDescription	Returns the file description specified in the file property
	LegalTrademarks	Returns the Legal Trademarks specified in the file property
	PrivateBuild	Returns the Information about a private version of the file
	FileVersion	Returns the Version number of the file specified in the file property
	OriginalFilename	Returns the Original name of the file, not including a path
	CreationDate	Returns the Creation time of the file
	ModifiedDate	Returns the last modification time of the file
	AccessTime	Returns the last access time of the file.

Example:

```
<FileProperties Path="%DataFileDirectory%\NAVENG32.DLL"
Value="FileVersion" />
```

<FileRead>

The FileRead tag is used to read the file contents according to the attribute values. The table below lists the attributes of the FileRead tag and the possible values.

Attribute	Possible Values	Description
Path	Path of the file	The absolute path where the file resides
Line	Line number	Line number , If the line number is given as 3 , the 3rd line from the file is returned
PivotString	Any string value	Searches for first occurrence of pivot string, if found will return the string following it till an end of line is encountered
Evaluate	false	If Evaluate is set to false, the content read is not returned instead

		the content are stored for further reference.
Name	Customer defined value	A variable to store the content read. The Name attribute is used in conjunction with the Evaluate attribute. If the Evaluate attribute is used, the content read is stored in the variable you defined in the Name attribute. Else, the content is not stored in the variable you defined in the Name attribute.

Example:

```
<FileRead Name="Temp" Path="C:\temp\<anyfile.txt>" Line="3"
PivotString="xyz" Evaluate="false" />
```

<DirRead>

The DirRead tag is used to read directory and file names within a directory. The table below lists the attributes of the DirRead tag and the possible values.

Attribute	Possible Values	Description
Name	Customer defined value	A variable to store the output. The Name attribute is used in conjunction with the Evaluate attribute. If the Evaluate attribute is used, the output is stored in the variable you defined in the Name attribute. Else, the output is not stored in the variable you defined in the Name attribute.
Path	Absolute path of the directory	Absolute path of the directory to read
Verb	File	Returns the file names within the path specified, delimited by " "
	Dir	Return the directories names within the path specified, delimited by " "
Evaluate	False	If Evaluate is set to false, the output is stored in the variable you defined in the Name attribute for further reference.

Example:

```
<DirRead Name="DIR" Evaluate="false" Path="%DataPath%" Verb="dir" />
```

<XMLRead>

The XMLRead tag is used to read the value or attribute value of XML Node. The table below lists the attributes of the XMLRead tag and the possible values.

Attribute	Possible	Description

Values		
Name	Customer defined variable	A variable to store the output. The Name attribute is used in conjunction with the Evaluate attribute. If the Evaluate attribute is used, the output is stored in the variable you defined in the Name attribute. Else, the output is not stored in the variable you defined in the Name attribute.
Evaluate	False	If Evaluate is set to false, the output is stored in the variable you defined in the Name attribute for further reference.
Node	Tag name	Specifies the XML tag where data is to be read
Attribute	Attribute name	This is an optional attribute. Retrieves value from the attribute name of the tag specified in the Node attribute.
Path	XML file location	Absolute path of the XML file

Example:

```
<XMLRead Name="Protect" Evaluate ="false"
Path="%InstallDirectory%settings\xyz.xml" Node="root/xyz/profile"
Attribute = "profile" />
```

<String>

The String tag is used to perform string operations. The table below lists the attributes of the String tag and the possible values.

Attribute	Possible Values	Description
Verb	Return	Returns the value in the Value attribute.
	Compare	Compares the value in the Value attribute with the Expected Value attribute and returns true if both values match, else returns false.
	NCompare	Compares the value in the Value attribute with the Expected Value attribute and returns false if both values match, else returns true.
value	Any string value	Stores the string value for operation specified in the Verb attribute, usage of substitution variable is allowed. It is also allowed to have value as a composition of substitution variables and normal strings. For example, %Major%.%Minor%
Evaluate	True	Used when the Value attribute has any substitution variable and needs substitution.

ExpectedValue	Any string value	Contains any string value. Used when the Verb is Compare or Ncompare.

Example:

```
<String Verb="Compare" value="%xyz%" ExpectedValue="abc" Evaluate="true"/>
```

<Integer>

The Integer tag is used to perform integer operations. The table below lists the attributes of the String tag and the possible values.

Attribute	Possible Values	Description
Verb	Return	Returns the value in the Value attribute
	Compare	Compares the value in the Value attribute with the Expected Value attribute and returns true if both values match, else returns false.
Value	Any integer value	Contains returned value or the value used for comparisons
ExpectedValue	Any integer value	Used when the verb is Compare or Ncompare

Example:

```
<Integer Verb="Compare" value="%DisableRealtimeMonitoring%"  
ExpectedValue="0"/>
```

Logical Connector Tags

The logical connector tags are used to combine operation tags to accomplish a task. When logical connector tags are used, the value returned by the last tag in the logical connector tags is the return value. Nesting of logical connector tags is allowed. The logical connector tags work the same way logical operators work.

The logical connector tags available are

<And>

<Or>

<Not>

The <And> tag processes all the nodes placed within it, while the <Or> tag processes the nodes sequentially till it encounters a successful operation, after which it terminates.

Example snippet:

```
- <SystemProtectionTurnOff>  
- <And>
```

```

<Registry Reg="HKEY_LOCAL_MACHINE\SOFTWARE\McAfee\SystemCore\VSCore\On
Access Scanner\BehaviourBlocking" Key="APEnabled" Type="REG_DWORD"
value="0" />

<Registry Reg="HKEY_LOCAL_MACHINE\SOFTWARE\McAfee\SystemCore\VSCore\On
Access Scanner\BehaviourBlocking" Key="BOPEnabled" Type="REG_DWORD"
value="0" />

</And>

</SystemProtectionTurnOff>

- <RunFullSystemScan>

- <Or>

<Process Verb="Create" Arch="64" ProcessName="scan64.exe"
Path="%InstallDirectory%\x64\scan64.exe" />

<Process Verb="Create" Arch="32" ProcessName="scan32.exe"
Path="%InstallDirectory%\scan32.exe" />

</Or>

</RunFullSystemScan>

```

Sample STMLibrary.xml

This section explain the STMLibrary.xml schema of one product for each security type—anti-virus, anti-sypware, and software firewall.

Anti-virus McAfee VirusScan Enterprise 8.8

McAfee VirusScan Enterprise 8.8 Schema

```

<?xml version="1.0" encoding="UTF-8"?>

<SecurityApplication>

<AntiVirus Name="McAfee VirusScan Enterprise 8.8" Version="8.8"
WMIName="McAfee VirusScan Enterprise">

- <IsInstalled>

<Registry Verb="Exists" Reg="HKEY_LOCAL_MACHINE\SOFTWARE\McAfee\ePolicy
Orchestrator\Application Plugins\VIRUSCAN8800" Key="Install Path"
Type="REG_SZ" />

</IsInstalled>

- <Detection>

- <ProductName>

<Registry Reg="HKEY_LOCAL_MACHINE\SOFTWARE\McAfee\ePolicy
Orchestrator\Application Plugins\VIRUSCAN8800" Key="Product Name"
Type="REG_SZ" />

</ProductName>

```

```
- <ProductVersion>

<Registry Reg="HKEY_LOCAL_MACHINE\SOFTWARE\McAfee\ePolicy
Orchestrator\Application Plugins\VIRUSCAN8800" Key="Version" Type="REG_SZ"
/>

</ProductVersion>

- <VendorName>

<String Verb="Return" value="McAfee, Inc." />

</VendorName>

- <DataFileDirectory>

<Registry Reg="HKEY_LOCAL_MACHINE\SOFTWARE\McAfee\AVEngine" Key="DAT"
Type="REG_SZ" />

</DataFileDirectory>

- <DefinitionDate>

<Registry Reg="HKEY_LOCAL_MACHINE\SOFTWARE\McAfee\AVEngine"
Key="AVDatDate" Type="REG_SZ" />

</DefinitionDate>

- <DefinitionVersion>

<Registry Name="AVDatVersion" Reg="HKEY_LOCAL_
MACHINE\SOFTWARE\McAfee\AVEngine" Key="AVDatVersion" Type="REG_DWORD"
Evaluate="false" />

<Registry Name="AVDatVersionMinor" Reg="HKEY_LOCAL_
MACHINE\SOFTWARE\McAfee\AVEngine" Key="AVDatVersionMinor" Type="REG_DWORD"
Evaluate="false" />

<String Verb="Concat" value="%AVDatVersion%.%AVDatVersionMinor%"
Evaluate="true" />

</DefinitionVersion>

- <EngineVersion>

<Registry Name="EngineVersionMajor" Reg="HKEY_LOCAL_
MACHINE\SOFTWARE\McAfee\AVEngine" Key="EngineVersionMajor" Type="REG_
DWORD" Evaluate="false" />

<Registry Name="EngineVersionMinor" Reg="HKEY_LOCAL_
MACHINE\SOFTWARE\McAfee\AVEngine" Key="EngineVersionMinor" Type="REG_
DWORD" Evaluate="false" />

<String Verb="Concat" value="%EngineVersionMajor%.%EngineVersionMinor%"
Evaluate="true" />

</EngineVersion>

- <InstallDirectory>
```

```
<Registry Reg="HKEY_LOCAL_MACHINE\SOFTWARE\McAfee\ePolicy
Orchestrator\Application Plugins\VIRUSCAN8800" Key="Install Path"
Type="REG_SZ" />

</InstallDirectory>

- <IsFileSystemProtectionStatusEnforced>

- <Or>

<Registry Reg="HKEY_LOCAL_MACHINE\SOFTWARE\McAfee\SystemCore\VSCore\On
Access Scanner\BehaviourBlocking" Key="APEnabled" Type="REG_DWORD" />

<Registry Reg="HKEY_LOCAL_MACHINE\SOFTWARE\McAfee\SystemCore\VSCore\On
Access Scanner\BehaviourBlocking" Key="BOPPEntabled" Type="REG_DWORD" />

</Or>

</IsFileSystemProtectionStatusEnforced>

- <IsFullSystemScanInProgress>

- <Or>

<Process Verb="Check" ProcessName="scan32.exe"
Path="%InstallDirectory%scan32.exe" />

<Process Verb="Check" ProcessName="scan64.exe"
Path="%InstallDirectory%x64\scan64.exe" />

</Or>

</IsFullSystemScanInProgress>

- <IsUpdateInProgress>

<Process Verb="Check" ProcessName="MCUPDATE.EXE"
Path="%InstallDirectory%MCUPDATE.EXE" />

</IsUpdateInProgress>

<LastFullSystemScan />

</Detection>

- <Remediation>

- <SystemProtectionTurnOn>

- <And>

<Registry Reg="HKEY_LOCAL_MACHINE\SOFTWARE\McAfee\SystemCore\VSCore\On
Access Scanner\BehaviourBlocking" Key="APEnabled" Type="REG_DWORD"
value="1" />

<Registry Reg="HKEY_LOCAL_MACHINE\SOFTWARE\McAfee\SystemCore\VSCore\On
Access Scanner\BehaviourBlocking" Key="BOPPEntabled" Type="REG_DWORD"
value="1" />

</And>

</SystemProtectionTurnOn>
```



```

- <SystemProtectionTurnOff>

- <And>

<Registry Reg="HKEY_LOCAL_MACHINE\SOFTWARE\McAfee\SystemCore\VSCore\On
Access Scanner\BehaviourBlocking" Key="APEnabled" Type="REG_DWORD"
value="0" />

<Registry Reg="HKEY_LOCAL_MACHINE\SOFTWARE\McAfee\SystemCore\VSCore\On
Access Scanner\BehaviourBlocking" Key="BOPEnabled" Type="REG_DWORD"
value="0" />

</And>

</SystemProtectionTurnOff>

- <RunFullSystemScan>

- <Or>

<Process Verb="Create" Arch="64" ProcessName="scan64.exe"
Path="%InstallDirectory%x64\scan64.exe" />

<Process Verb="Create" Arch="32" ProcessName="scan32.exe"
Path="%InstallDirectory%scan32.exe" />

</Or>

</RunFullSystemScan>

- <RunDefinitionUpdate>

<Process Verb="Create" ProcessName="MCUPDATE.EXE"
Path="%InstallDirectory%MCUPDATE.EXE" WaitToComplete="true" />

</RunDefinitionUpdate>

</Remediation>

</AntiVirus>

</SecurityApplication>

```

Anti-spyware McAfee VirusScan Enterprise 8.8

McAfee VirusScan Enterprise 8.8 Schema

```

<?xml version="1.0" encoding="UTF-8"?>

<SecurityApplication>

<AntiSpywares>

<AntiSpyware Name="McAfee VirusScan Enterprise 8.8" Version="8.8">

<IsInstalled>

<Registry Verb="Exists" Reg="HKEY_LOCAL_MACHINE\SOFTWARE\McAfee\ePolicy
Orchestrator\Application Plugins\VIRUSCAN8800" Key="Install Path"
Type="REG_SZ" />

```

```
</IsInstalled>

<Detection>

<ProductName>

<Registry Reg="HKEY_LOCAL_MACHINE\SOFTWARE\McAfee\ePolicy
Orchestrator\Application Plugins\VIRUSCAN8800" Key="Product Name"
Type="REG_SZ" />

</ProductName>

<ProductVersion>

<Registry Reg="HKEY_LOCAL_MACHINE\SOFTWARE\McAfee\ePolicy
Orchestrator\Application Plugins\VIRUSCAN8800" Key="Version" Type="REG_SZ"
/>

</ProductVersion>

<VendorName>

<String Verb="Return" value="McAfee, Inc." />

</VendorName>

<DataFileDirectory>

<Registry Reg="HKEY_LOCAL_MACHINE\SOFTWARE\McAfee\AVEngine" Key="DAT"
Type="REG_SZ" />

</DataFileDirectory>

<DefinitionDate>

<Registry Reg="HKEY_LOCAL_MACHINE\SOFTWARE\McAfee\AVEngine"
Key="AVDatDate" Type="REG_SZ" />

</DefinitionDate>

<DefinitionVersion>

<Registry Name="AVDatVersion" Reg="HKEY_LOCAL_
MACHINE\SOFTWARE\McAfee\AVEngine" Key="AVDatVersion" Type="REG_DWORD"
Evaluate="false" />

<Registry Name="AVDatVersionMinor" Reg="HKEY_LOCAL_
MACHINE\SOFTWARE\McAfee\AVEngine" Key="AVDatVersionMinor" Type="REG_DWORD"
Evaluate="false" />

<String Verb="Concat" value="%AVDatVersion%.%AVDatVersionMinor%"
Evaluate="true"/>

</DefinitionVersion>

<EngineVersion>

<Registry Name="EngineVersionMajor" Reg="HKEY_LOCAL_
MACHINE\SOFTWARE\McAfee\AVEngine" Key="EngineVersionMajor" Type="REG_
DWORD" Evaluate="false" />
```

```
<Registry Name="EngineVersionMinor" Reg="HKEY_LOCAL_
MACHINE\SOFTWARE\McAfee\AVEngine" Key="EngineVersionMinor" Type="REG_
DWORD" Evaluate="false"/>

<String Verb="Concat" value="%EngineVersionMajor%.%EngineVersionMinor%"
Evaluate="true"/>

</EngineVersion>

<InstallDirectory>

<Registry Reg="HKEY_LOCAL_MACHINE\SOFTWARE\McAfee\ePolicy
Orchestrator\Application Plugins\VIRUSCAN8800" Key="Install Path"
Type="REG_SZ" />

</InstallDirectory>

<IsFileSystemProtectionStatusEnforced>

<Or>

<Registry Reg="HKEY_LOCAL_MACHINE\SOFTWARE\McAfee\SystemCore\VSCore\On
Access Scanner\BehaviourBlocking" Key="APEnabled" Type="REG_DWORD" />

<Registry Reg="HKEY_LOCAL_MACHINE\SOFTWARE\McAfee\SystemCore\VSCore\On
Access Scanner\BehaviourBlocking" Key="BOPPEntabled" Type="REG_DWORD" />

</Or>

</IsFileSystemProtectionStatusEnforced>

<IsFullSystemScanInProgress>

<Or>

<Process Verb="Check" ProcessName="scan32.exe"
Path="%InstallDirectory%scan32.exe" />

<Process Verb="Check" ProcessName="scan64.exe"
Path="%InstallDirectory%x64\scan64.exe" />

</Or>

</IsFullSystemScanInProgress>

<IsUpdateInProgress>

<Process Verb="Check" ProcessName="MCUPDATE.EXE"
Path="%InstallDirectory%MCUPDATE.EXE" />

</IsUpdateInProgress>

<LastFullSystemScan />

</Detection>

<Remediation>

<SystemProtectionTurnOn>

<And>
```

```
<Registry Reg="HKEY_LOCAL_MACHINE\SOFTWARE\McAfee\SystemCore\VSCore\On
Access Scanner\BehaviourBlocking" Key="APEnabled" Type="REG_DWORD"
value="1" />

<Registry Reg="HKEY_LOCAL_MACHINE\SOFTWARE\McAfee\SystemCore\VSCore\On
Access Scanner\BehaviourBlocking" Key="BOPEnabled" Type="REG_DWORD"
value="1" />

</And>

</SystemProtectionTurnOn>

<SystemProtectionTurnOff>

<And>

<Registry Reg="HKEY_LOCAL_MACHINE\SOFTWARE\McAfee\SystemCore\VSCore\On
Access Scanner\BehaviourBlocking" Key="APEnabled" Type="REG_DWORD"
value="0" />

<Registry Reg="HKEY_LOCAL_MACHINE\SOFTWARE\McAfee\SystemCore\VSCore\On
Access Scanner\BehaviourBlocking" Key="BOPEnabled" Type="REG_DWORD"
value="0" />

</And>

</SystemProtectionTurnOff>

<RunFullSystemScan>

<Or>

<Process Verb="Create" Arch="64" ProcessName="scan64.exe"
Path="%InstallDirectory%\x64\scan64.exe" />

<Process Verb="Create" Arch="32" ProcessName="scan32.exe"
Path="%InstallDirectory%\scan32.exe" />

</Or>

</RunFullSystemScan>

<RunDefinitionUpdate>

<Process Verb="Create" ProcessName="MCUPDATE.EXE"
Path="%InstallDirectory%\MCUPDATE.EXE" WaitToComplete="true" />

</RunDefinitionUpdate>

</Remediation>

</AntiSpyware>

</AntiSpywares>

</SecurityApplication>
```

Microsoft Windows Firewall 7

Microsoft Windows Firewall 7 Schema

```
<?xml version="1.0" encoding="UTF-8"?>

<SecurityApplication>

  <Firewalls>

    <Firewall Name="Microsoft Windows Firewall" Version="7">

      <IsInstalled>

        <Registry Verb="Exists" Reg="HKEY_LOCAL_
MACHINE\SYSTEM\CurrentControlSet\services\SharedAccess" Key="ImagePath"
Type="REG_SZ" />

      </IsInstalled>

      <Detection>

        <ProductName>

          <String Verb="Return" value="Microsoft Windows Firewall" />

        </ProductName>

        <ProductVersion>

          <Registry Name="ServiceDll" Reg="HKEY_LOCAL_
MACHINE\SYSTEM\CurrentControlSet\services\SharedAccess\Parameters"
Key="ServiceDll" Type="REG_SZ" Evaluate="false"/>

          <FileProperties Path="%ServiceDll%" Value="ProductVersion" />

        </ProductVersion>

        <VendorName>

          <String Verb="Return" value="Microsoft Corp." />

        </VendorName>

        <InstallDirectory>

          <Registry Reg="HKEY_LOCAL_
MACHINE\SYSTEM\CurrentControlSet\services\SharedAccess\Parameters"
Key="ServiceDll" Type="REG_SZ" FormatType="getDirName"/>

        </InstallDirectory>

        <Policies>

          <RunCommand Name="WINFW" Command="netsh" Args="advfirewall firewall show
rule all" Path = "%InstallDirectory%" Evaluate="False"
Parse="FirewallRule"/>

          <String Verb="Return" value="%WINFW%" Evaluate="true"/>

        </Policies>

        <IsEnabled>

          <Registry Name="StandardProfile" Reg="HKEY_LOCAL_
```

```
MACHINE\SYSTEM\CurrentControlSet\services\SharedAccess\Parameters\Firewall
Policy\StandardProfile" Key="EnableFirewall" Type="REG_DWORD"
Evaluate="false" />

<Registry Name="DomainProfile" Reg="HKEY_LOCAL_
MACHINE\SYSTEM\CurrentControlSet\services\SharedAccess\Parameters\Firewall
Policy\DomainProfile" Key="EnableFirewall" Type="REG_DWORD"
Evaluate="false" />

<Registry Name="PublicProfile" Reg="HKEY_LOCAL_
MACHINE\SYSTEM\CurrentControlSet\services\SharedAccess\Parameters\Firewall
Policy\PublicProfile" Key="EnableFirewall" Type="REG_DWORD"
Evaluate="false" />

<Or>

<Integer Verb="Return" value="%StandardProfile%"/>

<Integer Verb="Return" value="%DomainProfile%"/>

<Integer Verb="Return" value="%PublicProfile%"/>

</Or>

</IsEnabled>

<IsProductAuthentic>

</IsProductAuthentic>

</Detection>

<Remediation>

<FirewallProtectionTurnOn>

<RunCommand Command="netsh" Args = "advfirewall set allprofiles state on"
Path = "%InstallDirectory%" />

</FirewallProtectionTurnOn>

<FirewallProtectionTurnOff>

<RunCommand Command="netsh" Args = "advfirewall set allprofiles state off"
Path = "%InstallDirectory%" />

</FirewallProtectionTurnOff>

</Remediation>

</Firewall>

</Firewalls>

</SecurityApplication>
```

Updating Accelerite Content Network Content Through Command Line

As an alternative to using the Accelerite Content Network page (under Infrastructure Management on the Operations tab) to schedule or trigger a Accelerite Content Network content update, you can use the `content-update.cmd` command-line utility located in the following directory:

Core and Satellite: `<InstallDir>\VulnerabilityServer\bin`

Note that this directory is not automatically placed in your PATH when Sentient is installed.

This utility has the following syntax:

```
content-update.cmd [-settingName <settingValue>]...
```

This command has both ["Required Settings" below](#) and ["Optional Settings" on the next page](#). Note that you must always specify a value for the `content_source` setting.

Any values that you specify on the command line override the stored configuration settings specified elsewhere (see ["Stored Settings" on page 365](#)). If you do not specify a value for a particular setting, the stored configuration setting is used.

Note: The `content-update` command writes status and error messages to the `vms-commandline.log` file.

See ["Examples" on page 366](#) for typical uses of the `content-update.cmd` command.

Required Settings

The following table lists the required settings for the `content-update.cmd` command.

Note: Any values that you specify on the command line override the stored configuration settings that were specified elsewhere (see ["Stored Settings" on page 365](#)). If you do not specify a value for a particular setting, the stored configuration setting is used.

Required Settings for content-update.cmd

Setting	Description
<code>content_source</code>	This setting is required. It specifies the source for the updated content. This must be one of the following: FILESYSTEM – Acquire the content from a location in the file system. The content must have previously been downloaded from Accelerite Content Network to this file system location. The <code>content_path</code> setting must also be specified, either on the command line or on the Accelerite Content Network page under Infrastructure Management on the Operations tab. CSDB_MASTER – Acquire the content from master content previously published to the configuration server database (CSDB). This data will be used to load the Reporting database. Service deployment content will NOT be republished. This is intended for use when a test configuration server

Setting	Description
	content deck has been imported into a production configuration server.
content_path	The fully qualified path to the file system location containing the content that you manually obtained from Accelerite Content Network. This setting is only required if you specified FILESYSTEM as the content_source. This path can specify either a directory or a ZIP archive file. The directory structure (or ZIP file structure) must exactly match the structure of directories and files created when an automatic Accelerite Content Network update is performed:  bundles  definitions You must also replicate the sub-directories under these folders to match the automatic update structure. In some cases, Accelerite Content Network updates only a subset of the content. In this case, some of these directories may not be delivered during a Accelerite Content Network update. In any case, when you update from the File System, your directory structure must match that delivered by Accelerite Content Network.

Optional Settings

The following settings for the `content-update.cmd` command are optional.

Note: Any values that you specify on the command line override the stored configuration settings that were specified elsewhere (see ["Stored Settings" on the next page](#)). If you do not specify a value for a particular setting, the stored configuration setting is used.

Optional Settings for content-update.cmd

Setting	Description
csdb_host	Configuration Server network addressable system name. This can be a fully qualified host name, localhost, or an IP address.
livenetwork_username	User name for the Accelerite Content Network subscription.
livenetwork_password	Password for the Accelerite Content Network subscription.
livenetwork_proxy_http_server	HTTP proxy server used to connect to the Accelerite Content Network download site. This option must have the following form: <http https>://<host>:<port>
livenetwork_proxy_http_username	User name for the HTTP proxy server, if any, used to connect to the Accelerite Content Network download site.
livenetwork_proxy_http_password	Password for the HTTP proxy server, if any, used to connect to the Accelerite Content Network download site.

Setting	Description
password	
reporting_db_databasename	Name of the database instance that you created before installing Sentient, which is discussed in the “Create the Sentient Database” section of the <i>Sentient Enterprise Installation and Upgrade Guide</i> .
reporting_db_drivename	Name of the database driver to use (either <code>oracle</code> or <code>sqlserver</code>). This must map to a supported driver.
reporting_db_server	Network addressable server name where the Reporting database is located.
reporting_db_port	Reporting database port number. This must be empty if the port is dynamic. If the port is static, it must be a value between 1 and 65536.
reporting_db_username	User name for the Reporting database.
reporting_db_password	Password for the Reporting database.

Stored Settings

If you do not specify a value for one of the `content-update` settings, the values specified on the following Accelerite Content Network configuration pages are used by default:

Stored Settings for `content-update.cmd`

Option	Where Specified
csdb_host csdb_port csdb_username csdb_password	Sentient First-Time Setup wizard
livenetwork_username livenetwork_password livenetwork_proxy_http_server livenetwork_proxy_http_username livenetwork_proxy_http_password	Accelerite Content Network page and Proxy Settings page
reporting_db_databasename reporting_db_drivename reporting_db_server reporting_db_port reporting_db_username reporting_db_password	Automatically configured when Sentient is installed

Examples

Example 1— Perform a content update from a local directory

```
content-update.cmd -content_source FILESYSTEM -content_path  
c:\mycontent
```

Example 2 – Perform a content update from a local ZIP file

```
content-update.cmd -content_source FILESYSTEM -content_path  
c:\mycontent\content.zip
```

To view full usage information for `content-update.cmd`, type the following command from the `<InstallDir>\bin` directory:

```
content-update.cmd -?
```

Move Accelerite Content Network Content from a Test Environment to a Production Environment

You may find it useful to test your Accelerite Content Network content in a small controlled environment before performing a large scale rollout. To do this, you will first create a test Sentient environment with its own “test” Configuration Server Database (CSDB) and “test” Reporting database. After completing your testing, you will export the “test” relevant Domain and then import that CSDB content into your production Sentient environment.

Note: The files used to export and import CSDB content are known as a “deck.”

Before following these procedures, be sure to review ["How Accelerite Content Network Content is Updated" on page 340](#).

Testing your Accelerite Content Network content in a controlled test environment

1. In the test environment, perform an Accelerite Content Network content update—either automatically from the Accelerite Content Network subscription site, or manually from the file system.
2. Test the updates by running scans and reviewing the pertinent reports and dashboard panes.

Moving your Accelerite Content Network content from a controlled test environment to a production environment

Caution: In the `raddbutil` commands shown here, there are no spaces after the commas. If you cut and paste these commands from this guide or the online help, be sure to remove any spaces introduced by the paste operation.

1. Connect to the test CSDB and use the `raddbutil` tool to export the relevant deck:
 - a. Go to the Configuration Server `bin` directory on the system where you want to export the data (the test environment).

- b. If the `admin` user has a password, use the following command:

```
raddbutil EXPORT DATA=TRUE,WALK=TRUE,OUTPUT=  
<tempDir>,USERID=admin,PASSWORD=<password> PRIMARY.<DOMAIN>
```

If the `admin` user does not have a password, use the following command:

```
raddbutil EXPORT DATA=TRUE,WALK=TRUE,OUTPUT=  
<tempDir>,USERID=admin PRIMARY.<DOMAIN>
```

In both cases, `<tempDir>` is the directory where the exported files will be placed on the test CSDB system.

For more information, see ["Configuration Server Database Utility \(RadDBUtil\)" on page 667](#).

2. Transport the relevant deck files to the production CSDB system using the file transfer mechanism of your choice.
3. On the production CSDB system, use the `raddbutil` tool to import the relevant deck:
 - a. Go to the Configuration Server directory on the system where you want to import the data (the production environment).

- b. If the `admin` user has a password, use the following command:

```
raddbutil IMPORT  
INPUT=<tempDir>,COMMIT=TRUE,ACCEPT=A+D+U,USERID=admin,PASSWORD=  
<password>
```

If the `admin` user does not have a password, use the following command:

```
raddbutil IMPORT  
INPUT=<tempDir>,COMMIT=TRUE,ACCEPT=A+D+U,USERID=admin
```

In this case, `<tempDir>` is the directory on the production CSDB system where the files are placed.

4. In the production environment, load the production Reporting database using the “master” content in the relevant deck that you just imported.

There are two ways to do this:

- **Method 1:** Use the Sentient Console
 - i. Click Infrastructure Management under the **Operations** tab.
 - ii. In the left navigation menu, select **Live Network**.
 - iii. Click the **Update Now** tab.
 - iv. Select the **From the Configuration Server Database** update option.
 - v. Click the **Update Now** button.

- **Method 2:** Use the `content-update` command-line utility
content-update.cmd -content_source CSDB_MASTER

See ["Updating Accelerite Content Network Content Through Command Line" on page 363](#) for more information about the `content-update` command.

In either case, using the CSDB_MASTER content source forces the update tool to only update the Reporting database content and bypass performing any updates to the packages linked to the relevant content. This ensures that the service content you deployed in your test environment will exactly match the content that you will be deploying in your production environment.

More Information about Security and Compliance Management

The following sections contain information about configuring and viewing security and compliance management information in the Sentient Console:

- "Using the Dashboards" in the *Sentient Enterprise Console Online Help*
- "Using Reports" in the *Sentient Enterprise Console Online Help*
- "Accelerite Content Network" in the *Sentient Enterprise Console Online Help*

Visit the following web sites to learn more about security and compliance management:

- <http://cve.mitre.org>
- <http://nvd.nist.gov>
- <http://nvd.nist.gov/scap.cfm>
- <http://oval.mitre.org>
- <http://www.us-cert.gov>

Chapter 8

OOB Devices Management

OOBM Features

You can perform Discover, Heal, and Protect operations (based on device type) on the provisioned devices in your network through the Out of Band Management (OOBM) options in Sentient console.

The following sections describe in detail the OOBM features that enable you to ["Discover" below](#), ["Heal" on the next page](#), and ["Protect" on page 372](#) the OOB devices on your network.

Note: Protect feature is relevant to vPro devices only.

Discover

You can discover the ["Hardware Assets" below](#) and ["Software Assets" below](#) on all of the provisioned OOB devices on your network.

Hardware Assets

Intel vPro devices store hardware asset information in flash memory. DASH devices store this information in the network controller's NVRAM. Both can be read anytime even if the device is powered off. The only conditions are that the devices are physically connected to the network and that they are plugged into a power source. The OOB devices do not rely on software agents to prevent accidental data loss. You can use the Sentient Console to access this information and use it to:

- Determine the exact specifications for any hardware component on the device that may need to be replaced.
- Identify compatibility issues.
- Inspect the configuration of a device before provisioning a new operating system.
- Retrieve ad-hoc inventory information even when the machine is powered off.

Software Assets

Intel vPro enables you to view a list of applications that have registered themselves with the third party data storage (3PDS) on the vPro device. For HP applications that have registered with the 3PDS, you can also view the data that the applications have written to the scratch pad area of the vPro device. DASH devices store software asset information in the network controller's NVRAM. You can use the Sentient Console to access this information and use it to:

- Determine if there are applications installed on the device that are taking advantage of vPro or DASH. The exact use of data storage is specific to the application.
- Retrieve out of band information for vPro and DASH-aware HP applications.
- Confirm that vPro and DASH-aware applications are correctly registered. This can help with troubleshooting certain applications.
- Confirm that vPro and DASH-aware HP applications are working correctly.
- View the software list of applications that you want the OOBM agent running on the vPro device to monitor.
- View the system message that the OOBM agent will display to the console of the vPro device if the Agent Presence policy is activated.

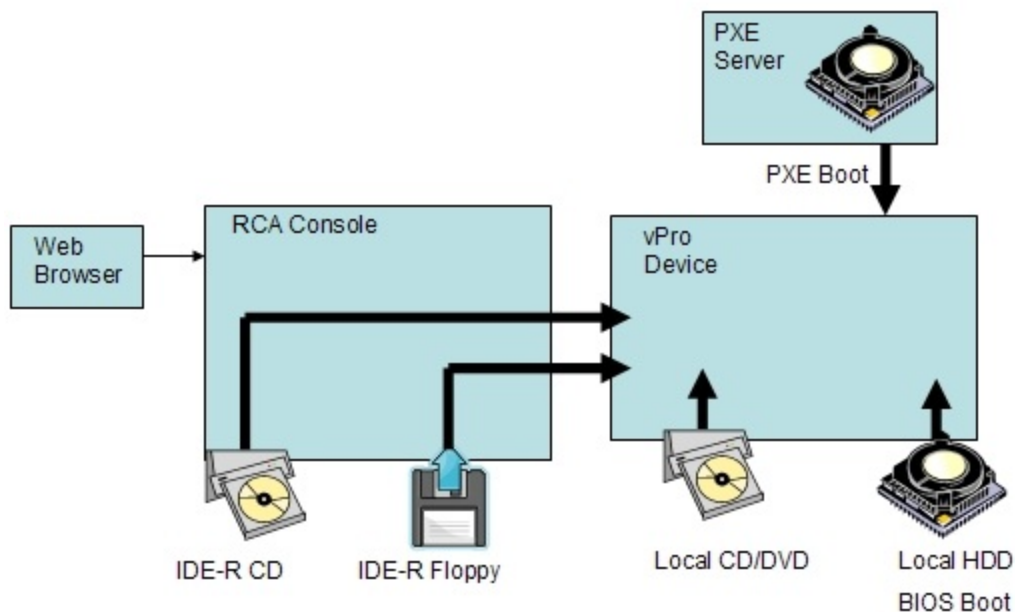
Heal

Heal operations include ["Remote Operations"](#) below and ["Event Management"](#) on the next page.

Note: Event Management is relevant to vPro devices only.

The following figure illustrates the various remote management operations that can be performed through the OOBM options in the Sentient Console (IDE-R pertains to vPro only).

Overview of Remote Management Operations



Remote Operations

Intel vPro and DASH devices provide out of band access to remotely diagnose and repair devices after software, operating system, and hardware failures. Through the Sentient Console, you can

remotely view the power state of a device, reboot from its hard disk, reboot from an image on its local CD/DVD, reboot from a remote CD or floppy drive, reboot from a PXE server, and reboot to BIOS setup. The power state of the system is changed when any remote operation is performed or the system is in an idle state.

You can use these capabilities to perform remote power management operations, which include:

- Powering devices up and down
- Restarting OOB devices using console text redirection and IDE-R

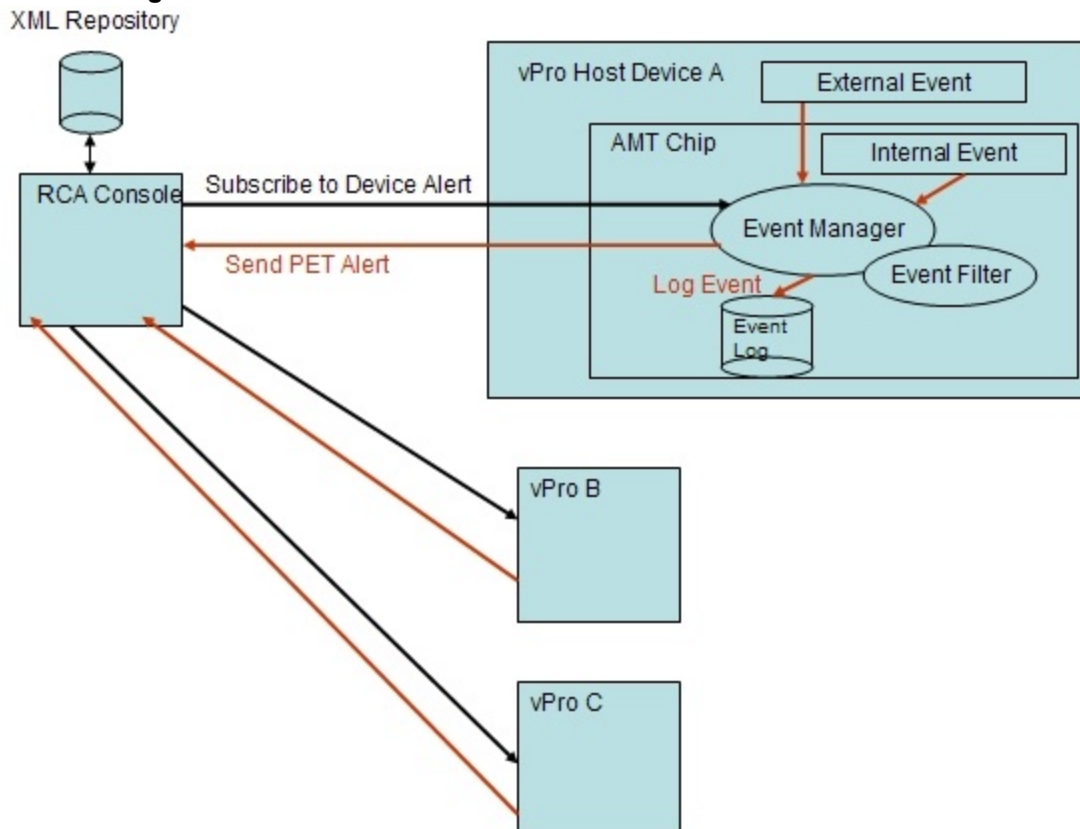
Note: Integrated drive electronic redirect (IDE-R) is available on vPro devices only.

Power management operations enable you to restore problematic OOB devices to a sane state.

Event Management

The following figure illustrates the dynamics involved in event management on vPro systems.

Event Management Overview



Intel vPro provides for alerting and event logging to assist you in diagnosing problems quickly to reduce end-user downtime. Events are created from external sources such as the System Management (SM) bus and hardware sensors. There are also internal, vPro-self created events. There are several sources for these internal events. They include events triggered by System Defense filters, by Agent Presence failure, firmware updates, and several other scenarios. So an event can be a physical occurrence, such as a fan failure or a filter-detected occurrence (due to a

change in traffic pattern), such as a virus attack. If an event occurs on the system, the Event Manager residing on the vPro chip raises an event and looks in the Event Filter to determine what actions to take. The Event Filter defines a set of criteria that is applied to each incoming platform event. If the incoming event matches the criteria, the Event Filter specifies the actions to take. These actions can include sending an alert to the Sentient Console, logging the event in the vPro log, or both.

You must subscribe to a vPro device for event alerts created by that device to be sent to the Sentient Console. The subscription provides a destination in the Event Filter for an event alert. You can subscribe or cancel event alerting to determine what device alerts you want displayed to the Sentient Console.

Intel vPro devices come with a set of default event filters built in to their firmware chip. The Sentient Console also enables you to view the events in the event log for a specific vPro device to determine the type, severity, date, and description of each logged event.

You can use this capability to control event alerting and logging. The alerts sent to the console and the events written to the event log enable you to determine if heal or protect actions are required for a specific device.

Protect

You can protect the vPro devices on your network from malicious software attacks and worm proliferation. Intel vPro provides this capability through packet filtering and by monitoring the presence of critical OOBM agents running on the devices in your network. It also enables you to quarantine worm-infected devices by providing a mechanism that continuously observes outgoing traffic to detect and impede the proliferation of worms.

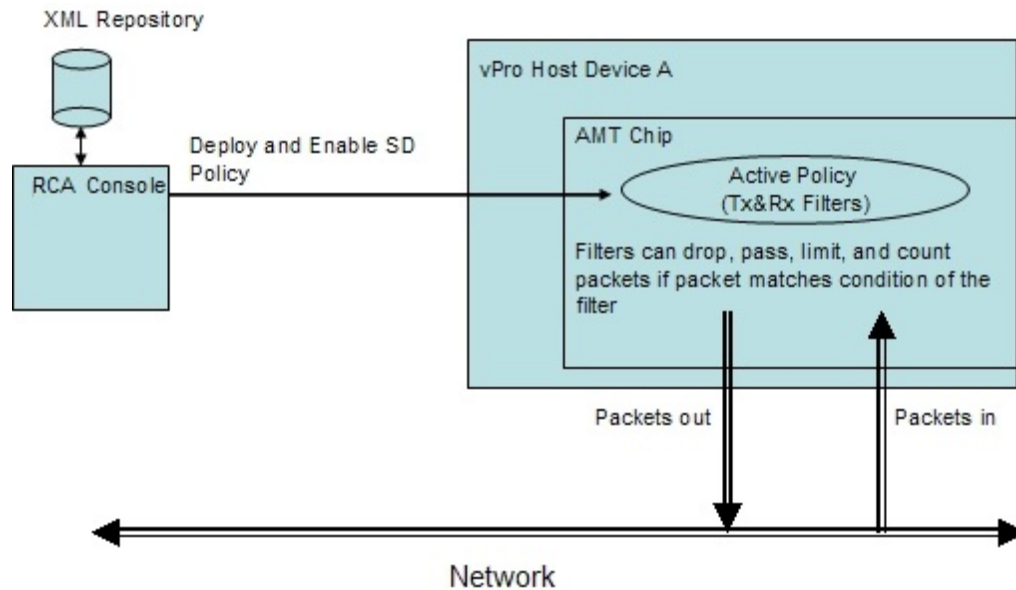
These topics are discussed in the following sections:

- ["System Defense" below](#)
- ["Agent Presence" on page 375](#)
- ["Network Outbreak Containment Heuristics" on page 376](#)

System Defense

The following figure illustrates an overview of how System Defense works by using policies and filters that monitor packets that are transmitted or received by vPro devices.

System Defense Overview



Policies

The System Defense vPro capability enables the Sentient Console to define and enforce network security policies. A System Defense policy contains a set of filters that are applied to incoming and outgoing network packets combined with actions to take when a packet matches (or does not match) the conditions in the filter. System Defense enables for selective network isolation of Ethernet and IP protocol flows based on the filters associated with a policy. These filters enable the Management Console to pass, limit, or block specific IP-based network flows and to keep traffic counts or log the occurrence of these flows.

The Sentient Console stores these filters and policies in its XML repository. The console can then deploy the System Defense policies to multiple vPro devices where they reside in the firmware of the devices.

After you have deployed policies to a vPro device, you can use the Sentient Console to enable a policy on that device to become the default System Defense policy. You can also set a policy as the Agent Presence policy that can be enabled by an watchdog action. This is discussed in greater detail in ["Agent Presence" on page 375](#). The enabled policy with the higher priority becomes the active policy for the device. After a policy is activated, the vPro device inspects each incoming and outgoing packet and performs the necessary actions specified by the filters associated with the policy. If a vPro device has more than one network interface card (NIC), that is, wired and wireless, you can enable a System Defense policy and set an Agent Presence policy for each NIC.

Filters

Filters can perform the following actions if the conditions associated with the filter are matched:

- Pass packets
- Discard packets
- Limit packets
- Count packets to collect statistical data

In addition to performing the packet-related actions, a filter can also cause an event to be raised.

There are two filter modes. They are the following:

- **Transmit:** Filters in this mode are applied to packets transmitted from the vPro device to the network. Such filters can be used to block all traffic from a device suspected of being infected preventing it from infecting other devices on the network. The default filter in transmit mode catches all the transmit packets that do not match any of the other policy transmit filters. Filters in this mode can be of the pass, limit, statistical, or drop type.
- **Receive:** Filters in this mode are applied to packets received from the network to the vPro device. Such filters can be used to block all packets received by the device after boot until a OOBM agent starts such as an antivirus agent. The default filter in receive mode catches all the receive packets that do not match any of the other policy receive filters. Filters in this mode can be either of the pass or drop type.

There are several filter types. They are the following:

- **Default Else:** This is the default Else filter for both the Receive and Transmit directional modes. It is used for catching all packets that do not match the condition of any of the policy filters. If the Else filter is matched (that is the packets do not match the conditions of any other filter), a filter action can be created.
- **Drop:** This is the drop filter for both the Receive and Transmit directional modes. It discards all packets matching the conditions of the filter.
- **Pass:** This is the pass filter for both the Receive and Transmit directional modes. It passes all packets matching the conditions of the filter.
- **Statistical Drop/Pass:** This is the statistical filter for both the Receive and Transmit directional modes. They count the number of packets that match the conditions of the filter. They are used for collecting statistical data. They can either pass or discard packets based on whether they are statistical pass or statistical drop filters.
- **Rate Limit:** This is the rate limit filter for both the Receive and Transmit directional modes. They limit the number of specific types of packets per second that are received or transmitted matching the conditions of the filter. This filter has a threshold and when the threshold is reached, it cuts off any additional traffic.

In addition to filter types, Transmit filters can have anti spoofing enabled. When this property is enabled, all outgoing packets are checked and the source IP is compared to the network interface IP address. If the IP addresses do not match, the packets are dropped. If this filter is enabled, it prevents a host from falsifying its identity by sending IP packets with a source IP address that is different from its assigned IP address.

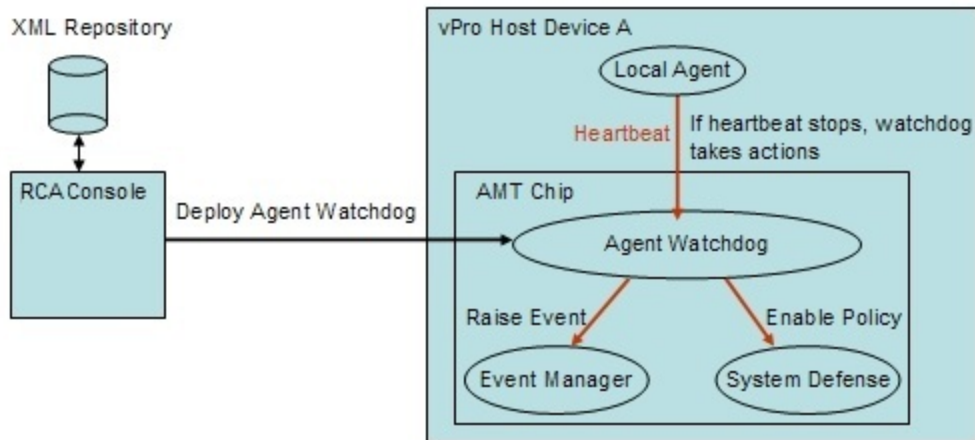
Intel vPro supports 32 filters in Receive (Rx) mode and 32 filters in Transmit (Tx) mode. One each of the filters in 32 Tx and Rx modes is used as the Else (non-matching) filter. If anti spoofing is enabled, it utilizes one of the filters in Tx mode. This reduces the available filters for a vPro device to 31 in-bound and 30 out-bound.

Note: If this limit is reached, you cannot deploy additional policies containing filters to the vPro device until you delete some of the existing filters on that device.

Agent Presence

The following figure illustrates the components involved when monitoring for the presence of a OOBM agent on a vPro host device.

Agent Presence Overview



The Agent Presence capability enables the Sentient Console to create an watchdog to monitor the state of a OOBM agent running on the host CPU of the vPro device. The OOBM agent is typically software that secures the vPro device by monitoring security applications related to anti-virus or firewall protection. After the OOBM agent starts, it sends heartbeats to the watchdog at regular intervals. If the heartbeat stops, the watchdog will take actions to protect the device.

OOBM Agents

As indicated, the OOBM agent secures the vPro device by monitoring the state of any critical applications running on the device. The list of applications that the OOBM agent monitors is user-defined. If a monitored application stops running, the OOBM agent stops sending a heartbeat to the watchdog. If the watchdog enables the Agent Presence policy and it becomes the active policy, a system message is displayed to the console on the vPro device. The system message is also user-defined although a default message is provided. The application list and system message are stored in the 3PDS of the vPro device.

When the OOBM agent is installed (and the software list of applications has been created and deployed to the device), it is automatically started as an NT service. After the OOBM agent starts, it does the following:

1. Registers with the watchdog.
2. Gets the heartbeat interval from the vPro chip and starts sending heartbeats to the watchdog.

3. Reads the 3PDS to get the list of applications to monitor and starts monitoring the applications. The same heartbeat interval is used for monitoring the applications list.
4. Stops sending the heartbeat, if an application on the application list stops running.
5. Shuts down the watchdog and displays the user-defined system message that it reads from the 3PDS.

Watchdogs

The actions an watchdog can perform are as follows:

- Enable the Agent Presence policy if it has been set. If it has a higher priority than the default System Defense policy, it will become the active policy, and the filters associated with this policy will be activated to protect the network.
- Raise an event. Depending on what the Event Manager finds in the Event Filter, the event will be written to the event log on the vPro chip, and/or an event alert will be sent to the Sentient Console if it has been subscribed to.

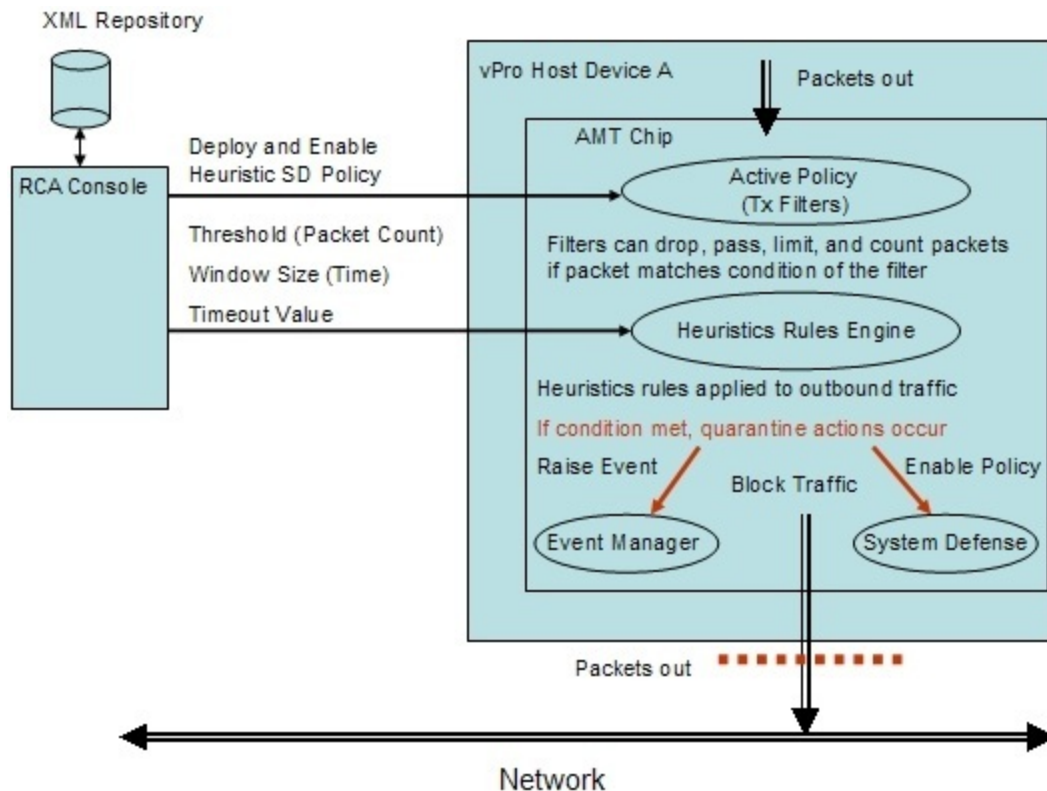
You can use the Sentient Console to do the following:

- Create a watchdog.
- Specify timers to detect when the OOBM agent initializes and periodically transmits “heartbeat” signals to its watchdog.
- Specify transition states for the OOBM agent that will trigger the watchdog actions. Valid states include the following:
 - not started
 - stopped
 - running
 - expired
 - suspended
- Set the actions for the watchdog to take if the transition criteria are met (namely, enable Agent Presence policy and/or enable event logging).
- Deploy (and undeploy) the watchdog to multiple vPro devices.
- Create the list of applications to be monitored by the OOBM agent.
- Create the message that will be displayed on activation of the Agent Presence policy.

Network Outbreak Containment Heuristics

The following figure illustrates the architectural overview of a worm-containment system.

Worm-containment Architecture



The worm-containment heuristics mechanism provides additional value to a network even when there are firewalls and intrusion detection systems in place on that network. Firewalls and intrusion detection systems can be used effectively only against known worms, but they are not effective against zero-day worm outbreaks.

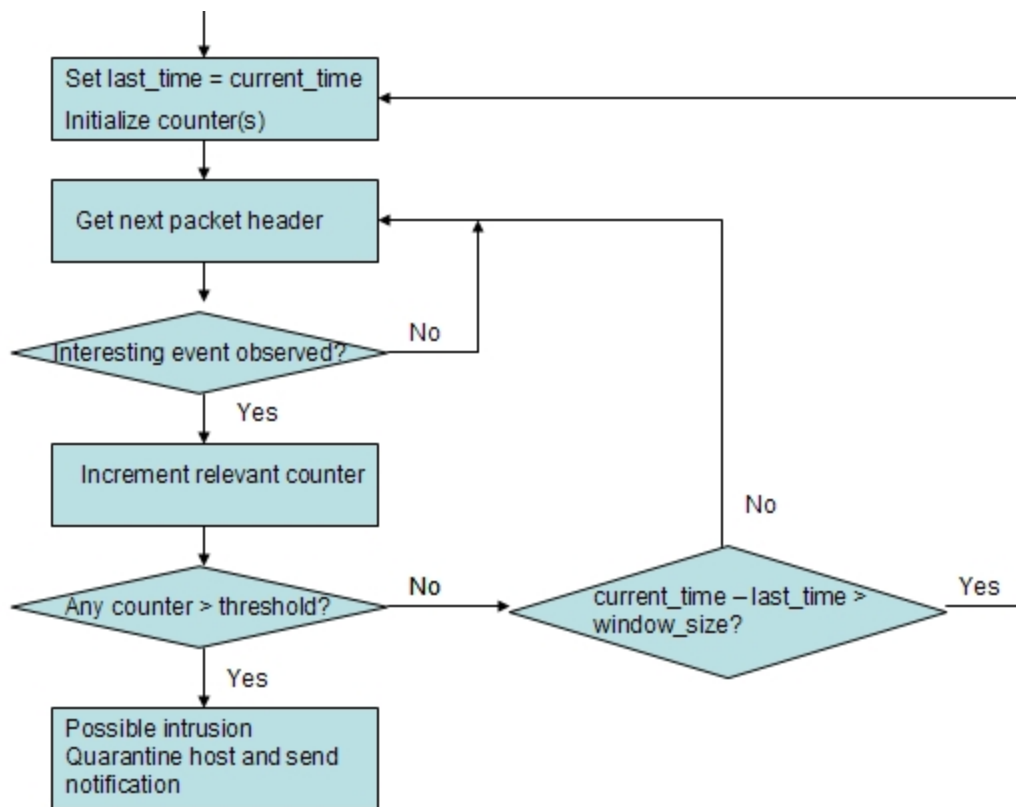
The vPro worm-containment system works by applying heuristic rules to the outbound traffic from the host vPro device. If the Heuristic Rules Engine detects an anomaly, the worm-containment system quarantines the host from the network. The Active Policy filters operate on the IP and TCP/UDP protocol header fields of the host traffic. As a result of this filtering, the vPro chip may take specific actions, such as, dropping the packets.

The Heuristics Rules Engine analyzes the traffic. If the Heuristics Rules Engine detects evidence of anomalous traffic, it can perform any of the following actions:

- Raise an event alert.
- Raise an event alert and block all outbound packets from the offending port.
- Raise an event alert and block all outbound packets from all ports.
- Raise an event and enable a vPro System Defense policy.

The worm-containment system relies on heuristic rules to detect traffic anomalies that could indicate scanning activity by a worm. The heuristic rules are based on the fundamental property of all self-propagating worms, namely, a worm has to contact new hosts to spread into the network. As a result, all heuristic rules identify events that suggest contact with a new host and monitor such events for anomalous patterns.

The following figure illustrates the basic mechanism employed by all heuristics.

Basic Operation of Heuristics

For additional information, see the Intel study on a heuristics-based worm-containment system.

Window Size and Threshold Values

All heuristics examine packet headers and count the number of “interesting events” in a given time interval. As a result, all heuristics expose two configurable parameters, namely, a window size and a threshold. The window size (time count) indicates the period at which the heuristic resets its counters. The threshold (packet count) represents a limit value, which when exceeded by the counter, indicates an anomalous event.

The Sentient Console enables you to configure these two parameters. When specifying these parameters, you must take into consideration two types of worms, which require different heuristic values. They are the fast spreading worm and the slow spreading worm. Different window sizes and threshold values should be used for the fast and slow spreading worms to successfully detect worm infections. Using a combined heuristic (different window sizes with appropriate threshold values) is more effective across a wider range of worms. The heuristic with the smaller window size is more effective for the fast worm, where the heuristic with the larger window size is effective for slower worms.

The configurable time window size ranges for fast and slow worms are the following:

- Fast: 10 milliseconds to one second (1000 milliseconds)
- Slow: one second (1000 milliseconds) to 50 seconds (50000 milliseconds)

The configurable threshold range for both heuristics is 8 to 64 packet count.

The following configurations are recommended for window sizes and threshold combinations:

- Fast: 8 packets in 10 milliseconds
- Slow: 64 packets in 50 seconds

Containment Actions and Timeout Values

The Sentient Console also enables you to specify the autonomous actions the system should take if the thresholds are exceeded. As indicated [here](#), you can choose to raise an event only or raise an event in combination with blocking outbound host traffic or enabling a heuristics System Defense policy. The Heuristics Rules Engine is disabled after the action is taken.

You can specify how long the containment actions should be applied to the vPro device through the Sentient Console. If you indicate a non-zero timeout value (greater than or equal to 20 seconds), the Heuristics Rules Engine stops scanning packets and applies the specified actions for that period of time. When the period of time has elapsed, the actions will automatically be removed and the chip will start scanning packets again. If you specify a zero timeout value, the Heuristics Rules Engine stops scanning packets and the actions are applied permanently. To remove the containment actions and start scanning packets again, you must manually trigger this to occur through the Sentient Console.

Setting Up vPro and DASH Devices

Intel vPro devices are enabled by Intel Active Management Technology (AMT). AMT is just one part of vPro technology. The Intel chipset and the Intel Network Interface Card (NIC) are also part of the vPro technology solution. Intel vPro devices can be discovered and managed even when powered off because the Intel AMT firmware stores information about them in non-volatile memory and provides a set of management operations that can be invoked by a remote management console.

Similarly, DASH-enabled devices can take advantage of out of band management. DASH is designed to provide the next generation of standards for secure out of band and remote management of desktop and mobile systems. DASH is one of several Distributed Management Task Force (DMTF) Management Initiatives, providing a comprehensive framework for syntax and semantics necessary to manage computer systems, independent of machine state, operating platform, or vendor.

Both technologies provide remote diagnostics and repair capabilities, which include hardware-based remote-boot and text console-redirection. On vPro devices, remote boot is provided through integrated drive electronics redirect (IDE-R), and text console redirection is available through serial over LAN (SOL) technology.

Intel vPro technology also enables you to automatically provision remote vPro devices. In addition, it provides System Defense and Agent Presence capabilities, which serve to protect vPro devices from malware attacks and the removal of OOBM agents that secure the system. In addition, it provides Network Outbreak Containment (NOC) heuristics, which is a mechanism for measuring, analyzing, and reacting to traffic to detect and impede the proliferation of worms.

All vPro OOBM capabilities can be secured through Transport Layer Security (TLS) and Digest authentication. Currently, for Dash-enabled device, the only supported mechanism is through Digest authentication.

Note: The Sentient server does not support Kerberos authentication.

Note: The only conditions for discovery and management of the OOB devices are that they are physically connected to the network and that they are plugged into a power source.

Setting up vPro Device

To make a vPro device operational, the SCS server (also referred to as the Setup and Configuration Server) performs all the necessary steps. The SCS server is the system that runs the Intel Setup and Configuration Service (SCS) and is relevant to vPro devices only.

For the SCS server and the vPro device to communicate with each other, several setup and configuration steps must be performed on both sides as described in the following sections.

Note: Be sure to use the version of the SCS software that is bundled with the Sentient Core distribution media located in the `Media\OOBM\win32\AMT Config Server` directory. Also, if you are migrating from an earlier release, ensure that you migrate the SCS software as well to the one included on the distribution media for the current release. Otherwise, you may experience erroneous behavior. This option does not appear on the Operations tab if you have opted to manage only DASH devices because it is not relevant for these devices.

SCS Provisioning of vPro Devices

The SCS server and the vPro device communicate securely. The SCS server creates and sends the vPro device initial parameter values and security-related configuration information. Set up and configuration include setting parameters that are either included in profiles or created automatically.

For more information on vPro security-related configuration information, see the *Introduction to Intel SCS* section in the *Intel Setup and Configuration Service Console User Guide Version 5.4* located in the `Media\oobm\win32\AMT Config Server` directory on the Sentient Core distribution media.

For more information on the Setup and configuration parameters, see the *Setup and Configuration Operational Overview* section in the *Intel Setup and Configuration Service Console User Guide Version 5.4*.

Sentient and SCS on Different Systems

You may want to install the SCS software and Sentient software on separate systems since the supported operating system platforms for SCS and Sentient may be different.

Setting Up TLS Mode

If you do not require TLS, skip to "Setting up SCS Server" on page 386.

Security is important for many vPro features, especially for redirection. The usage model of serial over LAN (SOL) and drive electronics redirect (IDE-R) includes remote troubleshooting that enables remote diagnostics, boot, and OS installation. These procedures usually involve authentication steps, which require usernames and passwords to be sent over the LAN as part of the redirection session. If the vPro device supports TLS, the Sentient server will establish a TLS session with it before opening SOL or IDE-R sessions, thus ensuring that all relevant network communications are secure.

You must use Mutual Authentication for vPro devices and Server Authentication for SCS server.

Note: If you set up TLS authentication on the Sentient server, it manages only those vPro devices that are configured to use TLS. Conversely, if you have not set up TLS authentication on the Sentient server, it manages only those vPro devices that are not configured to use TLS.

Perform the following tasks to set up TLS mode:

Task 1: Installing Microsoft Certification Authority

The Microsoft Certification Authority enables you to create the client certificate and to export the client certificate and the trusted root certificate so that they can be used for TLS authentication.

To install the Microsoft CA on Windows Server 2003, follow these steps:

Note: The steps to install Microsoft CA may vary based on the platform.

1. Click the Windows **Start** button and select the **Control Panel**.
2. Double-click **Add or Remove Programs**.
3. From the left panel, click **Add/Remove Windows Components**.
4. Select the **Certificate Services** checkbox. A warning message is displayed indicating that the machine name or the domain membership of the machine cannot be changed while it acts as a certificate server. Click **Yes** to close the message window.
5. Click **Details**.
6. Select both the **Certificate Services CA** checkbox and the **Certificate Services Web Enrollment Support** checkbox to specify the subcomponents of Certificate Services. Click **OK**.
7. On the Windows Components window, click **Next**. The CA Type window opens.
8. Select **Enterprise root CA** as the type of certificate you want to setup. Click **Next**. The CA Identifying Information dialog opens.
9. In the CA Identifying Information dialog, specify the following:
 - **Common Name for this CA:** The Common Name for the Microsoft CA must be the same as the Computer Name on which you are installing the Microsoft CA.
 - **Distinguished name suffix:** An example is `DC=AMT, DC=HP, DC=COM`
10. Complete the remaining steps in the Windows Components Settings Wizard. Click Finish when done.

This installs the Microsoft CA software. The Microsoft CA enables you to create the client certificate as described in the next sections, as well as export trusted root certificates that are used for signing in the authentication process. The export procedure is described in "[Task 7: Exporting the Root Certificate](#)" on page 385.

Task 2: Creating Internet Information Services Server Certificate on SCS Server

A server certificate is required to provide secure communication between the SCS server and the SCS console.

For detailed steps, see the *Securing the Connection to IIS Using SSL* section under the *Installing Microsoft's Certification Authority* chapter in the *Intel Setup and Configuration Service Installation Guide Version 5.4* located in the `Media\oobm\win32\AMT Config Server` directory on the Sentient Core distribution media.

Task 3: Creating Client Certificate Template

You must create a client certificate for TLS mutual authentication that will be installed on the Sentient server.

To create the client certificate template, follow these steps:

1. Click the Windows **Start** button and select **Run**.
2. Enter **MMC** and click **OK**. The Microsoft Management Console opens.
3. From the File menu, select **Add/Remove Snap-in**.
4. Click **Add**. The Add Standalone Snap-in dialog box opens.
5. Select **Certificate Templates** and click **Add** and then click **Close**.
6. Click **OK** in the Add/Remove Snap-in window.
7. Click **Certificates Templates** in the left pane. All existing templates display to the right pane of the console.
8. Right-click the Web Server Template and select **duplicate template**.
9. On the **General** tab, specify the following:
 - **Template display name**: The name you want to appear when the template is displayed. For example, it can be `ClientAuthTpl`.
 - **Template name**: The name of the template. It can be the same as the display name.
 - Select the **Publish certificate in Active Directory** checkbox.
10. On the **Request Handling** tab, select the **Allows private keys to be exported** checkbox.
11. On the **Extensions** tab, select **Application Policies** and click **Edit**. The Edit Application Policies Extension window opens. The Server Authentication policy is displayed by default.
12. Select **Server Authentication policy** and click **Remove**. Now the Application policies list is empty.
13. Click **Add**. The Add Application Policy window opens.
14. In the Add Application Policy window, select **Client Authentication Policy** and click **OK**. The Add Application Policy window closes and the Edit Application Policies Extension window opens. The Client Authentication Policy is displayed in the list.

15. In the Edit Application Policies Extension window, click **Add**. The Add Application window opens.
16. Click **New**. The New Application Policy window opens.
17. Enter the Name, for example, `AMTRemote`, and Object identifier, `2.16.840.1.113741.1.2.1` for the policy. This policy enables the private key for the server certificate to be exported.
18. Click **OK** three times. After adding the application policies, the Client Authentication policy and the `AMTRemote` policy are displayed in the Description of Application Policies list.
19. Edit Issuance Policies and click **Add**. Select the **All Issuance Policies** option and click **OK** twice. Now the all issuance policies option is displayed in the Description of Issuance Policies list.
20. On the **Security** Tab, select **Domain Admins** and set Read, Write, Enroll and Autoenroll permission. Select **Enterprise Admins** and set Read, Write, Enroll and Autoenroll permission. Select **Authenticated users** and set Read permission.
21. The other tabs (Issuance Requirements, Superseded Templates, and Subject Name) do not require any changes.
22. Click **Apply** and then **OK**. The new Template for Client Authentication is displayed in the right pane of Certificate Template.

Task 4: Issuing Client Certificate Template

Before you install the certificate, you must issue the certificate template that enables the template to become a certificate.

To issue the client certificate template, follow these steps:

1. Click the Windows **Start** button and select **Administrative Tools > Certificate Authority**.
2. Expand the installed Microsoft CA. Right-click Certificate Templates and select **New > Certificate Templates to Issue**. The Enable Certificate Templates window opens.
3. Select the client authentication template you created in "[Task 3: Creating Client Certificate Template](#)" on the previous page. In our example, it is the **ClientAuthTmpl** template.
4. Click **OK**. The issued certificate template is displayed in the right pane of the Certificate Templates window.

Task 5: Installing Client Certificate

1. You must install the client certificate on the SCS server as well as on the system where you installed the Sentient server .
 - To install client certificate on SCS server or Sentient server, use the following URL:
`http://<FQDN_CAServer>/certsrv`
where `<FQDN_CAServer>` is the Fully Qualified Domain Name of the Certification Authority Server.

You can find the fully qualified domain name (FQDN) of a machine from the Windows desktop on that machine.

- a. Right-click **My Computer**, select **Properties**. The System Properties window opens.
- b. Select the **Computer Name** tab. The FQDN of the machine is displayed in this window.

Make sure that this URL is added to the browser's trusted sites list. To add this site, do the following:

- a. In your browser, go to **Tools > Internet Options > Security** and select **Trusted Sites**.
 - b. Click **Sites**. The Trusted sites window opens.
 - c. Enter the URL in the **Add this Web site to the zone** field.
 - d. Clear the **Require Server Verification (https :) for all sites in this zone** check box.
 - e. Click **Add**.
 - f. Click **Close**.
2. Click **Request a certificate > Advanced certificate request > Create and submit a request to this CA**.
 3. Select the client certificate template from the **Certificate Template** pull down list. In our example, it is `ClientAuthTmpl`.
 4. In the **Identifying Information For Offline Template**, the **Name:** field must be the fully qualified name of the machine where client certificate will be installed.
 5. Select the **Mark keys as exportable** check box.
 6. Click **Submit**. Select **Yes** in the subsequent window and install the certificate.

Task 6: Exporting the Client Certificate

You *must* export the client certificate for both, Sentient server and SCS server.

To export the client certificate, follow these steps:

1. On the server, click the Windows **Start** button and select **Run**.
2. Enter **MMC** and click **OK**. The Microsoft Management Console opens.
3. From the File menu, select **Add/Remove Snap-in**.
4. Click **Add**.
5. Select **Certificates** and click **Add**.
6. Select **My user account** and click **Finish**.
7. Click **Close** and then **OK**.
8. From the left panel of the Microsoft Management Console, expand the Certificates-Current User branch.
9. Expand the Personal branch.
10. Select **Certificates**.
11. In the right panel, right click the client certificate. A popup menu opens. You can find the client certificate on the **Intended Purposes** tab.
12. Select **Open**. The Certificate Information Window opens.
13. Select the **Details** tab.
14. Click **Copy to File**. The Welcome window of the Certificate Export Wizard opens.

15. Click **Next**. The Export Private Key window opens.
16. Select **Yes > export the private key > Next**. The Export File Format window opens. Click **Next**.
17. Enter and confirm the password that protects the private key. You will need this password when you import the certificate. Click **Next**.
18. Enter the complete path and name for the file to Export. The suffix for the file indicating its file type (`.pfx`) is automatically created. Make note of the location since you will have to access it in subsequent procedures.
19. Click **Next** and then **Finish**.
20. Click **OK** to close the Certificate Information window.

Task 7: Exporting the Root Certificate

The root certificate is required on both the vPro device and the Sentient server to verify the digital signature on the server and client certificates.

- On the vPro device, the SCS server provisions the vPro device with the root certificate when it configures the profile for the device. The root certificate is used on the vPro device to authenticate the identity of the Sentient server client when the Sentient server sends the vPro device its client certificate.
- On the Sentient server, the root certificate is used to authenticate the identity of the vPro device when the device sends its certificate to the Sentient server. The root certificate is used for signing with vPro device to authenticate hardware and software queries and remote control capabilities.

To export the root certificate to the Microsoft CA server, follow these steps:

1. Click the Windows **Start** button and select **Administrative Tools > Certificate Authority**. The Certification Authority Management Console opens.
2. In the left window, right-click the installed Microsoft CA. From the shortcut menu, click **Properties**. The CA Properties window opens.
3. Select the **General** tab.
4. Select the certificate and click **View Certificate**.
5. Select the **Details** tab.
6. Click **Copy to file**. The Certificate Export Wizard opens.
7. Click **Next**.
8. Select the Export File Format as **DER encoded binary X.509 (.CER)** (default selection).
9. Click **Next**.
10. Enter the complete path and name for the file to Export. The suffix for the file indicating its file type (`.cer`) is automatically created. Make note of the location since you will have to access it in subsequent procedures.
11. Click **Next** and review the settings. To proceed, click **Finish**. A message displays indicating that the export was successful.
12. Click **OK**. You are returned to the **Details** tab on the Certificate window.

13. Click **OK** to close the Certificate window.
14. Click **OK** to return to the Certification Authority Management Console. Close the console.
15. Copy the certificate file to a location on the SCS server and Sentient server.

Task 8: Converting and Importing Certificates

The root certificate on the Sentient server is converted to PEM format so that it can be used to secure IDE-R/SOL operations when TLS mutual authentication is turned on. This conversion is described in ["Converting Certificates to PEM Format" on page 409](#).

The client certificate is installed on the Sentient server where it is converted to PEM format so that it can be used to secure IDE-R/SOL operations when TLS mutual authentication is turned on. The conversion is described in ["Converting Certificates to PEM Format" on page 409](#).

On the Sentient server, the root certificate is added to the Java key store so that it can be used to authenticate the identity of the vPro device server when the device sends the Sentient server its server certificate. This conversion is described in ["To Import the Root Certificate into the Java Key Store" on page 408](#).

Task 9: Configuring Security Parameters and Restarting RCA Application Server Service

You must specify the password for the PEM and PFX client certificate and set the configuration parameters for TLS authentication as described in ["Configuring Security Parameters " on page 398](#).

Finally, restart the **RCA Application Server** service.

Setting up SCS Server

The Setup and Configuration Service must be configured so that all communications with the vPro device are secure. This software is installed on the SCS server.

Install Setup and Configuration Service

To install the components of the Setup and Configuration Service, see the *Installation* chapter in the *Intel Setup and Configuration Service Installation Guide Version 5.4*.

Logging in to the SCS Console

To log in to the SCS console:

1. On the SCS server, click the Windows **Start** button and select **Intel AMT Configuration > Intel AMT SCS Console**. The log in window opens.
2. In the Service Name field, enter the URL path, including the virtual directory, for the SCS Web Services. The format is `<http|https>://<FQDN of Provision Server>/<SCS Web Services Virtual_Directory>`.
The SCS Console opens.

Configuring SCS Service Settings

You must select **None** for Active Directory Integration, in the SCS Service Settings windows.

For more information on configuring SCS Service settings, see the *Viewing and Configuring SCS Services* chapter in the *Intel Setup and Configuration Service Console User Guide Version 5.4*.

Creating and Configuring the Profile

This profile is associated with a vPro device as described in "[Task 3: Adding the new vPro system](#)" on page 389. The profile provides initial values to the vPro device during the provisioning process.

To configure the profile:

In the Optional Settings window, enable the following options:

- **ACL**
- **Use TLS secure communication for operations on the platform** (Required only if you plan to use TLS communication.)
- **Allow connection to WiFi network** (Required only if you want to manage using wireless NIC.)

In the ACL Details window, specify the following:

- User Type: Select **Digest User**.
- User/Group name: Enter the user name for this type of account.
- Password: Enter the password for the account and enable **Mask**.
- Access Type: Select **Both** from the list.
- Realms: Select the required realms in the list. Realms determine the type of operations the user account can perform when managing a vPro device with the Sentient Console. It is recommended that you select all realms except the Security Audit Log Realm for the first account that you create.

In the TLS window, specify the following (Required only if you plan to use TLS communication):

- Under Basic TLS Configuration: For TLS Server Certification Authority, select a **Certificate Authority** from the list. For Server Certificate Template, select **WebServer** from the list.
- Under Advanced TLS Configuration: Select **Use mutual authentication** check box. Select the desired TLS trusted root certificates in the list.
- Click **Add**. The Add Trusted Root Certificate window opens. In this window, select **From File** radio button and browse to the root certificate exported. You can view the certificate. Click **OK** to complete the task.

If you plan to configure the WPA-TKIP profile through the Wireless Access Point of your router, you must make the following selections in the WiFi Profile window:

- Profile Name: Enter a unique name for the profile that is different from the general profile name, for example, PSGWirelessProfile.
- SSID: Enter the SSID value that you used when you set up the WPA-TKIP profile in the router Wireless Access Point. See the wireless router documentation.
- Key Management Protocol: Select **WiFi Protected Access (WPA)** from the list.

- Encryption Algorithm: Select **Temporal Key Integrity Protocol (TKIP)** from the list.
- Passphrase: Select the **Passphrase** radio button. Enter the pass phrase value that you used when you set up the WPA-TKIP profile in the router Wireless Access Point. See the wireless router documentation.

Note: If you are configuring a profile for a vPro device with a wireless interface, you must first configure the WPA-TKIP profile in the Wireless Access Point of your wireless router. You need to see the documentation that comes with your wireless router.

For more information on Creating and Configuring the Profile, see the *Creating and Changing Profiles* chapter in the *Intel Setup and Configuration Service Console User Guide Version 5.4*.

Provisioning vPro Devices

Intel vPro devices are by default delivered in an unconfigured state (Factory mode). Before a management application can access the vPro device, the device must be populated with configuration settings that include username, password, network parameters, TLS certificates, and PID-PPS key necessary for secure communication.

You can configure vPro devices by entering the Management Engine BIOS Extension (MEBx) of each device and entering the required information manually. Provisioning devices manually uses the Pre-shared Key (PSK) mode to secure communication between the SCS server and the vPro device. For more information, see ["Configuring the vPro Device Using MEBx" below](#).

Alternatively, you can configure the devices by taking advantage of the Intel Remote Configuration process, which provisions devices automatically. Provisioning devices automatically uses the Public Key Infrastructure (PKI) to secure communication. For more information, see ["Configuring the vPro Device Using Remote Configuration" on the next page](#).

Configuring the vPro Device Using MEBx

Task 1: Configuring security keys

For information on how to configuring security keys, see the *Configuring Pre-Setup and Configuration Security Keys* section under the *Using USB Drives for TLS-PSK Keys* chapter in the *Intel Setup and Configuration Service Console User Guide Version 5.4*.

Note: It is recommended that you select **Fixed Password** in the Create TLS-PSK Keys window and supply the new password.

Task 2: Configuring the vPro Device through the MEBx.

See the vendor documentation for the device for definitive steps. You must set up the following parameters to configure the vPro Device through MEBx:

- Enter the hostname of the vPro device.
- Enable TCP/IP.
- Enable DHCP.

- Select **Provisioning Server**. Enter the IP address of the machine you are using to run Intel SCS.
- Set the listening port to 9971.
- Set the PID and PPS to the values specified when configuring security keys.

Task 3: Adding the new vPro system

For information on Adding the new vPro system, see the *Adding a Platform Definition* section under the *Preparing and Managing Platforms* chapter in the *Intel Setup and Configuration Service Console User Guide Version 5.4*.

Configuring the vPro Device Using Remote Configuration

Remote Configuration is a vPro mechanism that eliminates the need to manually install a PID/PPS pair on each device to enable setup. To take advantage of Remote Configuration, you must do the following:

Task 1: Purchase a security certificate from a trusted Certification Authority (CA).

The CA vendor must match one of the vendors whose root certificate hashes are built into the vPro firmware. Go to the vendor's website of choice to purchase an SSL certificate. Each site documents the step required to request, enroll, install, and move an SSL certificate. You must install the certificate in the System Certificate Store on the system where the SCS server is installed. For more information, see the *Remote Configuration Flow* section in the Remote Configuration appendix in the *Intel Setup and Configuration Service Console User Guide Version 5.4*.

Task 2: Remote Configuration of the vPro Device

There are two ways a vPro device can be provisioned through Remote configuration. They are the following:

- Bare Metal Remote Configuration of the vPro Device as described in "[Bare Metal Remote Configuration of the vPro Device](#)" below.
- Delayed Remote Configuration Provisioning through OOBM Agent as described in "[Delayed Remote Configuration Provisioning of the vPro Device](#)" on the next page.

Bare Metal Remote Configuration of the vPro Device

Typically, immediate setup of the vPro device occurs (if you have performed the Remote Configuration Requirements correctly) when you connect the device to the network. This is referred to as *bare metal configuration*. The device starts sending out "Hello" messages to the SCS server indicating that it has transitioned to setup mode. If the device is successfully provisioned in the Limited Network Access time interval that the network interface opens for that device, no further provisioning tasks need to be performed.

Note: For HP desktops, Limited Network Access time interval is 255 hours. After the time interval has elapsed, the interface will close if the setup and configuration time was not extended by a network command from the SCS.

To perform a bare metal configuration of the vPro device:

1. Connect the vPro device to the network with the SCS server. The vPro device must be connected in the same domain where SCS has been installed.
2. Turn on the vPro device.
The vPro device automatically sends out “Hello” packets. After the vPro device receives a message from the SCS server, the provisioning process is started where the SCS server loads all of the settings and data needed to enable the vPro device.
3. After the vPro device is configured, install an operating system. You can install an IT-specified operating system from the network onto the vPro device providing a complete “no touch” configuration of the vPro system.

Bare Metal Remote Configuration Failure

Although the bare metal provisioning process starts as soon as the vPro device is connected to the network and AC power, the device may not be successfully provisioned within its limited network access window. Reasons for failure include the following:

- OTP has been enabled on the SCS server
- Excessive network traffic
- Certificate mismatch with root hashes on the vPro device

Delayed Remote Configuration Provisioning of the vPro Device

If bare metal configuration fails, you can provision the device by using Delayed Remote Configuration. To understand what occurs during the transition to Setup mode and the provisioning process in Delayed Remote Configuration, see ["Delayed Remote Configuration of the vPro Device" on page 428](#).

If the vPro device could not be successfully provisioned in the limited time window of bare metal configuration, the OOBM agent invokes delayed configuration as part of its installation process on unprovisioned devices.

Installing the OOBM Agent

There are two ways to install the OOBM agent on vPro devices. One way is to install the OOBM agent manually on each vPro device. Alternatively, you can install the OOBM agent automatically to multiple vPro devices through the Sentient Core Console.

Both methods are described in the following sections.

Configuration Client Role

To be able to provision vPro devices through the OOBM agent, you must do the following regardless of the installation method you choose:

- A user with the role of configuration client must be created and added in the SCS console. For example, the user created and added in the SCS console can be named **SCSUser**.
- This user must be created in domain **VLAN1**, for example, **SCSUser@vlan1.xx.com**.

The credentials for the user with the configuration client role need to be provided during OOBM agent installation. The user credentials of the SCS configuration client must be provided correctly, otherwise the OOBM agent will not start the provisioning of the device through delayed setup.

Note: When installing the OOBM agent, you must provide a “dummy” user name and password even if you do not intend to provision devices using delayed configuration. If you do not provide a user name and password, the installation fails with error code 1920.

Note: In some cases, you may see an error message in the event log with the error code 1063 as a result of OOBM agent installation or service restart. This message is harmless and can be ignored.

Manually Installing the OOBM Agent on Individual vPro Device

To manually install the OOBM agent on the vPro device:

1. Copy the `oobmclocalagent.msi` file located in the `\Media\client\default\win32\oobm\LocalAgent` directory on the Sentient Core distribution media to the vPro device. Double-click the `oobmclocalagent.msi` file. Alternatively, you can also copy the `setup.cmd` file located in the same directory on the distribution media to the vPro device. Double-click the `setup.cmd` file or type `setup.cmd` on the command line. The `setup.cmd` file calls the `oobmclocalagent.msi` file.
2. Click **Next** and accept the license agreement.
3. Click **Next**. The Remote Configuration Parameters window opens. In this window, specify the following:
 - SCS Configuration Client User Name: Enter the user name of the user with the role of configuration client. The format is `SCS_User_Name@Domain_Name`.
 - SCS Configuration Client Password: Enter the password of the user with the role of configuration client.
 - SCS Profile ID: Enter the profile ID for the vPro device. You can find this information in the Profiles area of the SCS Console.
 - SCS Remote Configuration URL: Enter the URL path including the virtual directory for the Intel Setup and Configuration Service (SCS) remote configuration service. An example is https://<provisionserver.yourenterprise.com>/amtscs_rcfg, where `provisionserver.yourenterprise.com` is the fully qualified domain name (FQDN) of the IIS host machine and `amtscs_rcfg` is the SCS remote configuration service virtual directory on the host machine.
4. Click **Next**. The User Information window opens, which permits you to enter the vPro digest user credentials for the user defined in the SCS Profile ID (referenced in the previous step). In this window, specify the following:
 - User Name: Enter the vPro username of the digest user defined in the SCS Profile.
 - Password: Enter the vPro password of the digest user defined in the SCS Profile.
 - Select the **TLS Mode** check box if the vPro device is provisioned in TLS mode.
5. Click **Next** and follow the remaining steps in the install wizard.

The RCA Out of Band Management service starts as soon as the OOBM agent is installed.

Automatic Install on Multiple vPro Devices through Client Automation

As indicated, you can automatically install the OOBM agent on multiple devices through the Sentient Core Console.

There are two parts to the automatic install procedure. They are the following:

- You must publish the OOBM agent software to the Sentient Enterprise Configuration Server database by using the Sentient Enterprise Administrator Publisher.
- You must deploy the OOBM agent to the vPro target devices by using the Sentient Console.

Publishing the OOBM agent to the Client Automation Database

1. Click **Start > Programs > Sentient Enterprise Administrator > Sentient Enterprise Administrator Publisher** to invoke the Sentient Enterprise Administrator Publisher. The Logon window opens.
2. Log in to the Sentient Enterprise Administrator Publisher using the Sentient Enterprise user name and password. By default, the user name is `admin` and the password is `secret`. The Publishing Options dialog window opens.
3. Select **Windows Installer** from the list.
4. Click **OK**. The Select wizard opens.
5. Select the OOBM agent installer file (`oobmclocalagent.msi`) in the left navigation menu as the Windows installer file that you want to publish.
6. Click **Next**. The Edit wizard opens.
7. Click the **Properties** link. The properties for the installer file are displayed to the right. Ensure that all the properties that start with **AMT** are set correctly. These properties include the following:
 - **AMTUSERNAME**: Enter the vPro username of the administrator (Digest username).
 - **AMTPASSWORD**: Enter the vPro password of the administrator (Digest password for user).
 - **AMTTLSMODE**: Enter **1** if the vPro device is provisioned in TLS mode, otherwise, enter **0**.
 - **AMTPROVSERVERADD**: Enter the URL path including the virtual directory for the Intel Setup and Configuration Service (SCS) remote configuration service.
 - **AMTPROFILEID**: Enter the profile ID for the vPro device. You can find this information in the Profiles area of the SCS Console.
8. Also on the properties page, ensure that the following properties are set correctly:
 - **SCSUSERNAME**: Enter the user name of the user with the role of configuration client.
 - **SCSUSERPASS**: Enter the password of the user with the role of configuration client.
9. Click **Next**. The Configure wizard opens.

10. In the Configure window, specify the following (fields not listed are optional):
 - **Service ID:** Enter a text string to identify the service, for example, `Radia_CA_OOB_LOCAL_AGENT`.
 - **Description:** Enter a description for the software, for example, `Radia CA OOB Local Agent`.
 - **Software catalog:** Select **User Application** from the list.
 - **Limit package to systems with:** If you do not select any of the operating systems, the software is deployed to all vPro devices regardless of the operating system.

Note: The fields may vary depending on the Sentient license.

11. Click **Next**. The Publish wizard opens. Summary and Progress information are displayed.
12. Click **Publish**. The OOBM agent application is published to the Sentient Enterprise Configuration Server database.
13. Click **Finish**.
14. Click **Yes** in the pop-up window to exit the Sentient Enterprise Administrator Publisher.

Automatically Deploying the OOBM Agent to Multiple vPro Devices

For detailed instructions on the following procedures, see the *Sentient Enterprise Console Online Help*.

1. Import the vPro devices into Sentient.
2. Deploy the OOBM agent to the target vPro devices.
3. Create a static group and add the target vPro devices to the group.
4. Deploy the OOBM agent to the vPro device group.

Checking Version of the OOBM Agent on the vPro Device

1. Go to the installation directory, `<InstallDir>\OOBM Agent` on the vPro device.
2. Right-click the `OOBMCLocalAgent.exe` file and select **Properties** from the context menu.
3. Select the **Version** tab. The version information is displayed in this window.

OOBM Agent on 64-bit Platforms

If you install the OOBM agent on a 64-bit platform, you must ensure that the Intel-specific drivers for both the 32-bit and the 64-bit platforms are installed on the vPro device. You can find these drivers at www.HP.com/support. For desktops, the drivers can be found under the Software – System Management section. The name of the driver is Intel Local Management Service (LMS) and Serial-over-LAN (SOL) Support. For notebooks, the drivers are typically present under the `SWSetup\AppInst` directory of the Windows install directory. If not, these drivers can also be downloaded from the above mentioned HP support site. After the drivers have been installed, you must reboot the system.

Viewing the vPro Device

After provisioning the vPro device, you can view it in the vPro SCS console.

Note: You may have to wait a short period of time before seeing the device in the console.

To View the vPro Device

1. Open the vPro SCS Console.
2. Expand the Platform Collections branch.
3. Double-click **All Platforms** to view the vPro devices. The vPro devices are displayed with their provisioned status.

Changing the Authentication Mode

Note: This section is relevant only if you have set up TLS authentication. If you have not set up TLS, skip to the ["OOBM Configuration" on page 396](#) chapter.

After provisioning the vPro device, it is possible to change the authentication mode of the device by using the vPro SCS console.

To Configure the vPro Device in TLS Server Authentication Mode

1. Open the vPro SCS Console.
2. Expand the Configuration Service Settings branch and select **Profiles**. It lists profiles which are created for different vPro Devices.
3. Select the required profile for the vPro Device and click **Edit**.
4. Select the **Network** tab and select the following options:
 - Use TLS
 - TLS Server Authentication for both the Local and Remote Interface
5. Click **Apply** and then **OK**. You are returned to the Main SCS Console.
6. Expand the Intel AMT Systems branch and select **Global Operations**. The Global Operations window opens.
7. In the Provisioning pane of this window, click **Re-Provision**. It assigns a Request ID to this re-provision request. Make a note of the ID because you can use it when searching for log information in the Action Status log.
8. Click **OK**.
9. Expand the Logs branch and select **Actions Status**. This log displays the status of all requests. You can check if your re-provision request succeeded or not by using the Request ID assigned in the previous step.

To Configure the vPro Device in TCP Mode

1. Open the vPro SCS Console.
2. Expand the Configuration Service Settings branch and select **Profiles**. It lists profiles which are created for different vPro Devices.
3. Select the required profile for the vPro Device and click **Edit**.
4. Select the **Network** tab and clear the **Use TLS** check box.
5. Click **Apply** and then **OK**. You are returned to the Main SCS Console.
6. Expand the Intel AMT Systems branch and select **Global Operations**. The Global Operations window opens.
7. In the Provisioning pane of this window, click **Re-Provision**. It assigns a Request ID to this re-provision request. Make a note of the ID because you can use it when searching for log information in the Action Status log.
8. Click **OK**.
9. Expand the Logs branch and select **Actions Status**. This log displays the status of all requests. You can check if your re-provision request succeeded or not by using the Request ID assigned in the previous step.

Setting Up DASH Device

DASH Configuration

Provision DASH-enabled devices according to the documentation accompanying the device.

For DASH-enabled devices from Hewlett-Packard, DASH configuration information is documented in the "Broadcom NetXtreme Gigabit Ethernet Plus NIC" whitepaper.

To Access this Documentation

1. Go to www.hp.com.
2. Select **Support and Drivers > Product Support and Troubleshooting**.
3. Enter a product that supports this NIC, for example, the dc7900.
4. Select one of the dc7900 models.
5. Select **Manuals (guides, supplements, addendums, etc)**.
6. Scroll to the **White papers** section and select **Broadcom NetXtreme Gigabit Ethernet Plus NIC** whitepaper.

DASH Configuration Utilities

The DASH Configuration Utility (BMCC application) is part of the Broadcom NetXtreme Gigabit Ethernet Plus NIC driver softpaq, which is found in the drivers section for each product that

supports this NIC.

To Access this Utility

1. Go to www.hp.com.
2. Select **Support and Drivers > Drivers and Software**.
3. Enter a product that supports this NIC, for example, the dc7900.
4. Select one of the dc7900 models.
5. Select an operating system.
6. Scroll to the Driver-Network section and select to download the **Broadcom NetXtreme Gigabit Ethernet Plus NIC Drivers**.

OOBM Configuration

This chapter explains how to configure Out of Band Management (OOBM) after it has been installed through the Sentient Installer. For system requirements and installation information, see the *Sentient Enterprise Console Online Help* and the *Sentient Enterprise 11.00 Support Matrix* available at the following URL: <https://support.accelerate.com/hc/en-us/categories/200146610?timestamp=1663293128>. For more details, contact your Persistent Systems sales representative.

The configuration tasks explained in this chapter are optional. You can perform these tasks if you want to change the default settings in OOBM.

Information about Configuration Parameters

Reconfiguring the SCS Path

If necessary, you can change the SCS path currently configured through the Sentient Console. For more information, see "[Setting Configuration Parameters](#)" on page 399.

Reconfiguring Client Automation Web Services

If necessary, you can change the gateway URL for the Sentient Console. An example URL is `http://CAhost:3466/ca`, where *CAhost* is the fully qualified name of the Client Automation server and *ca* is the Client Automation web services virtual directory on the Client Automation server. For more information, see "[Setting Configuration Parameters](#)" on page 399.

Configuring IDE-R Drives

The OOBM software is installed on the server with default settings for the CD and floppy drives when you use integrated drive electronics redirect (IDE-R). These settings are configurable. For more information, see "[Setting Configuration Parameters](#)" on page 399. If the default settings do not agree with the drive specifications on your server, you must change the default drive settings to agree with your server. The CD and floppy drive paths must point to real drives or images.

Note: You must have your CD/DVD configuration set correctly (that is, pointing at a real CD/DVD drive) even when you are using the Floppy Drive boot option. The same is true for the Floppy configuration when you are using the CD/DVD Drive boot option. If there is no Floppy drive connected to the Sentient server, you can point to any local bootable ISO image instead of the Floppy drive for IDE-R operations.

Note: If you modify the setting for the CD or floppy drive to use an ISO or IMG file, respectively, (as opposed to a physical CD or floppy), the drive path to the ISO or IMG file must be visible to the server running Sentient Application Server. Consequently, copy all necessary ISO and IMG files to the local drive of the server running Sentient Application Server.

Also, if your ISO or IMG file is a shared network resource, you must use the Universal Naming Convention (UNC) syntax to access the network file. UNC syntax is the following:

```
\\hostname\sharefolder\file
```

When using UNC syntax, you must use the actual hostname of the machine and not its IP address.

Configuring SOL Ports

The OOBM software is installed on the Sentient server with default settings for the serial over LAN (SOL) starting port and for the maximum number of SOL ports you can have open at one time to have multiple simultaneous SOL sessions on vPro devices. You can change these values. For more information, see ["Setting Configuration Parameters" on page 399](#).

Note: On vPro devices, SOL sessions launch Windows HyperTerminal, which is bundled with most Windows operating systems. Check to see if HyperTerminal is already installed on the web browser machine that you will use to access the Sentient server. If HyperTerminal is not already installed, see Microsoft documentation for installation instructions.

HyperTerminal is not bundled with the Windows Vista® operating system. In this case, SOL sessions launch Telnet.

Configuring the SNMP Port

This is the SNMP port used to get alert messages from vPro devices. This port is configurable. For more information, see ["Setting Configuration Parameters" on page 399](#).

Configuring the IDE-R and SOL Time-out Values

Remote operations performed on vPro devices with wireless NICs can fail because the time-out value for the IDE-R and SOL sessions are exceeded since wireless communication tends to be slower. IDE-R and SOL time-out and heartbeat interval values are configurable. For more information, see ["Setting Configuration Parameters" on page 399](#).

Configuring Web Service Time-out Value

The OOBM Service communicates with the vPro devices by making web services calls to the device. A time-out value is specified for this communication. You can reconfigure this value if it is not appropriate for the current network conditions. For more information, see ["Setting Configuration Parameters" on the next page](#).

Configuring the Cache Size for DASH Devices

You can configure the number of DASH devices that are cached in memory for a particular user session. For more information, see ["Setting Configuration Parameters" on the next page](#). Modifying this value affects performance. This value is dependent on the availability of memory resources.

Configuring Security Parameters

Note: This step is required only if you have TLS configured.

Note: If TLS AMT Authentication is enabled, run the Sentient Application Server under the domain user account to have proper permissions for accessing the Java key store.

There are a number of configuration parameters that must be set for TLS authentication. They enable OOBM to locate the trusted root and client certificates, to know the passwords associated with them, and the FQDN of the Certification Authority server.

You must configure the following:

- Full path to the root certificate in PEM format (`root_certificate`)
- Full path to the client certificate in PEM format (`client_certificate_pem`)
- Full path to the client certificate in PFX format (`client_certificate_pfx`)
- Client Certificate CN (`ca_server_commonname`)

For information on how to set these parameters, see ["Setting Configuration Parameters" on the next page](#).

For information about certificates in PEM format, see ["Converting Certificates to PEM Format" on page 409](#).

In addition, you must specify the password for the PEM and PFX client certificate. When executing the commands in the following procedures, specify the path without double quotes.

To Specify the PEM Client Certificate Password

1. From the `<InstallDir>/OOBM/bin` directory, run the `amtpem_chgpwd.bat` file.
2. When prompted, enter the Sentient Installation Directory.

3. When prompted, enter the password for PEM Client Certificate.

To Specify the PFX Client Certificate Password

1. From the `<InstallDir>/OOBM/bin` directory, run the `amtpfx_chgpwd.bat` file.
2. When prompted, enter the Sentient Installation Directory.
3. When prompted, enter the password for PFX Client Certificate.

Configuring Watchdog Settings

When you are creating a watchdog, two of the settings for the watchdog are the OOBM agent's heartbeat interval (time between heartbeats sent to the watchdog) and the startup time before the OOBM agent starts sending heartbeats to the watchdog. You can change the default values to reflect the needs of your network. For more information, see ["Setting Configuration Parameters" below](#).

Configuration Settings Used for Debugging

There are two configuration parameters that you can set to help debug performance-related problems. They are the `cache_update_thread_size` and the `blocking_timer_time` parameters.

The `cache_update_thread_size` parameter enables you to change the number of threads that are used to update the cache layer. This value need not be changed under normal conditions. However, this value can be changed in association with the `blocking_timer_time` parameter to resolve performance issues.

The `blocking_timer_time` setting enables you to change the time-out value in case there are any socket problems on the Sentient server when calling on vPro web services. If there are any problems related to the sockets, it is recommended that you increase the time-out value.

For more information, see ["Setting Configuration Parameters" below](#).

Setting Configuration Parameters

You can set configuration parameters by modifying the two properties files located in the `<<RCA_Install_DIR>\oobm\conf\` directory.

Note: If you change the configuration parameters in the properties files located in this directory, you must restart the RCA Application Server service.

The following parameters can be found in or added to the `config.properties` file. You can edit this file to reconfigure the value of any of the parameters listed by entering a new value for an existing **key=value** pair or adding a new **key=value** pair.

Note: When specifying path and fully qualified file names in `config.properties`, you must use `"\"` or `"/` as the separator between directories or the name will not be read

correctly. For example, `C:\\certs\\cc.pem` or `C:/certs/cc.pem` is correct while `C:\certs\cc.pem` is incorrect.

The following table lists the parameters contained in this file with their default settings and descriptions.

Configuration Parameters in the config.properties File

Parameter (key)	Default Value	Description
scsserver_url	No default value	URL for the SCS server. You can change the SCS path currently configured.
radia_gateway	No default value	URL of the Sentient Console.
default_cddrive_path	D:	Default IDE-R CD drive setting. The CD path must point to a real drive or image.
default_fddrive_path	A:	Default IDE-R floppy drive setting. The floppy drive path must point to a real drive or image.
sol_port_start	9999	Starting SOL port
sol_number_of_port	10	Maximum number of SOL ports
snmp_trapd_port	162	SNMP port
vPro_webservice_timeout	15000 ms	Web service time-out value
devices_cachequeuesize	50	Cache size for DASH devices. Modifying this value affects performance.
root_certificate	No default value	Full path to root certificate in PEM format
client_certificate_pem_format	No default value	Full path to client certificate in PEM
client_certificate_pfx	No default value	Full path to client certificate in PFX format
ca_server_commonname	No default value	Client certificate CN
apwatchdog_heartbeat_interval	60 seconds	Watchdog OOBM agent heartbeat interval
apwatchdog_startup_time	300 seconds	Watchdog OOBM agent startup time interval

Parameter (key)	Default Value	Description
device_synchronization_timeperiod	0	Time period to reload device list from the SCS repository. The synchronization time interval has a default value of zero, indicating that automatic synchronization will not occur. If you want synchronization to occur automatically, set value to a non-zero value. The unit for this value is minutes.
group_synchronization_timeperiod	0	Time period to reload group device list from the CA repository. The synchronization time interval has a default value of zero, indicating that automatic synchronization will not occur. If you want synchronization to occur automatically, set value to a non-zero value. The unit for this value is minutes.
cache_update_thread_size	25	Cache thread size (for debugging only)
blocking_timer_time	100	Blocking time-out value (for debugging only)
devices_cachequeuesize	100	Size of the cache used to store vPro-related Java objects that are used for performing operations such as power management, deployment of system defense features, and so on (for debugging purposes only).
scsserver_url	No default value	URL for the SCS server. You can change the SCS path currently configured.
radia_gateway	No default value	URL of the Sentient Console.

Additional configuration parameters can be found in the configuration.properties file also located in the <RCA_Install_DIR>\oobm\conf\ directory. All of these parameters have been assigned default values, but some may require reconfiguration depending on your setup. The following table "Configuration Parameters in the configuration.properties File" lists the parameters contained in this file with their default settings and descriptions.

Note: The data in this file is critical for the proper functioning of Out of Band Management. Make sure you do not modify or delete items that are listed as "Not for end users" in the **Description** column in the following table "Configuration Parameters in the configuration.properties File".

Configuration Parameters in the configuration.properties File

Parameter	Default Value	Description
Active_Directory_FQDN_or_Hostname_	name	Device identification information returned from

Parameter	Default Value	Description
property		AD. You can choose the hostname (name) or the FQDN (dnsHostName). Using the default value (name) is safer since FQDN can fail because of subdomain DNS.
BEV_BOOT_SOURCE_VALUES	BEV	Boot source names for Boot Entry Vector
CACHE_SIZE	50	Cache size of OOBM web services system. For example, <code>CACHE_SIZE=50</code> specifies that at any point in time, a maximum of 50 devices will be cached by the system. When the cache is full and a new device needs to be added, the least accessed/used device is removed to accommodate the new device.
CACHE_WAIT_DURATION	2000	Cache wait duration in milliseconds. For example, <code>CACHE_WAIT_DURATION=2000</code> specifies that the system should not wait for more than 2000 milliseconds for the cache manager to respond. If the cache manager is busy for more than 2000 milliseconds, the system will not use the cache for the current operation.
CDDVD_BOOT_SOURCE_VALUES	CD/DVD,CD-ROM	Boot source names for CD
DASH_PORTS	623	Comma-separated list of DASH ports
DASH_TEXTREDIRECTION_TIME_DELAY	2	Time delay (in seconds) between text redirection connection and the power operation invocation.
DISCOVERY_DELAY	100	Discovery delay time. You can increase this value to overcome the socket connection exhaustion problem.
DISCOVERY_REQUEST	Contains actual content of the DASH discover request	Content of the DASH request for the discovery of DASH devices.
DISCOVERY_SEQUENCE	dash,vpro	Sequence in which an OOBM device is discovered. For example, <code>DISCOVERY_SEQUENCE = "dash,vpro"</code> means that the system will first check if the device is a DASH device, and if its not , the system will check if its a vPro device.

Parameter	Default Value	Description
ENABLE_BLIND_DISCOVERY	true	Blind discovery for OOBM devices. If enabled, the system will honor operational requests to currently undiscovered OOBM devices by first discovering them automatically and then performing the requested operation. If disabled, the OOBM device should be already discovered in the OOBM system. Otherwise, the system will throw an error.
FLOPPY_BOOT_SOURCE_VALUES	Floppy,Diskette Drive	Boot source names for Floppy
HDD_BOOT_SOURCE_VALUES	Hard Drive,Hard-Disk	Boot source names for Hard Drive
HTTP_CONNECT_TIMEOUT	3000	HTTP connection time out (maximum milli seconds HTTP connection can wait for a response)
HTTP_READ_TIMEOUT	200	HTTP read time (maximum milli seconds HTTP connections can wait for read response)
IDER_CLIENT_RX_TIMEOUT	10000	Client receive time-out value in milliseconds. If the time-out value elapses before the client receives any messages from the Sentient server, the client will shut down the IDE-R session. When an IDE-R session is open, the Sentient server continually sends out messages to make sure that the receive time-out value for the client does not expire (the Sentient server heartbeat interval is based on the client receive time-out setting). Minimum value: 10000 Maximum value: 65535 Default value: 10000
IDER_CLIENT_COMMAND_TIMEOUT	0	Client command transmit time-out value in milliseconds. This is the amount of time the client waits when sending out an IDE command. If the client does not receive a response from the Sentient server to the command within the specified amount of time, the client will close the IDE-R session. A value of 0 means that no command transmit time-out is used. Minimum value: 0

Parameter	Default Value	Description
		Maximum value: 65535 Default value: 0
IDER_CLIENT_HB_INTERVAL	5000	Client heartbeat interval in milliseconds. This is the amount of time the client waits before sending a heartbeat message to the Sentient server. A value of 0 means that no heartbeat messages are sent. In this case, the Sentient server will periodically send IDE-R keep-alive ping messages to the client when there is no activity to determine if it is still alive. Minimum value: 0 Maximum value: 65535 Default value: 5000
NETWORK_BOOT_SOURCE_VALUES	Network,PXE	Boot source names for PXE
NUMBER_OF_DISCOVER_WORKER_THREADS	5	Maximum number of threads that can be used for discovery
PCMCIA_BOOT_SOURCE_VALUES	PCMCIA	Boot source names for PCMCIA(For more info: http://en.wikipedia.org/wiki/PC_Card)
REVERTBACK_PREVIOUS_BOOT_ORDER	0	Boot order reset flag. You can choose to disable (0) or enable (1) to revert back to the previous boot order of the boot configuration when booting the device with a particular boot source. The default is to disable reverting back since this has a performance impact.
RevertBack_Previous_Boot_Order_Wait_Timer	10000	Time to wait (in milliseconds) for reverting back the boot order to the previous order after initiating the reboot operation. If the default value is not working, increase the value based on the machine's performance.
SOL_CLIENT_TX_BUFFERING_TIMEOUT	100	Client transmit buffering time-out value in milliseconds. This is the amount of time the client waits for its transmit buffer to become full before sending its buffered transmit bytes. A value of 0 means that the client will transmit its data only when its buffer becomes full. Minimum value: 0 Maximum value: 65535

Parameter	Default Value	Description
		Default value: 100
SOL_CLIENT_TX_OVERFLOW_TIMEOUT	0	Client transmit overflow time-out value in milliseconds. This is the amount of time the client waits when its transmit buffer is full before starting to drop transmit bytes. A value of 0 means no time-out. Minimum value: 0 Maximum value: 65535 Default value: 0
SOL_CLIENT_HB_INTERVAL	5000	Client heartbeat interval in milliseconds. This is the amount of time the client waits between sending heartbeat messages to the Sentient server indicating that the client is active. A value of 0 means that no heartbeats are sent. In this case, the Sentient server will not monitor the receive activity from the client to determine if it is active. Minimum value: 0 Maximum value: 65535 Default value: 5000
SOL_CLIENT_RX_TIMEOUT	10000	Client receive time-out value in milliseconds. If this amount of time elapses before receiving any messages from the Sentient server, the client shuts down the SOL session. When an SOL session is open, the Sentient server periodically sends heartbeat messages to make sure that the receive time-out for the client does not expire (the interval between Sentient server heartbeat messages is based on the client receive time-out). Minimum value: 10000 Maximum value: 65535 Default value: 10000
SOL_CLIENT_FIFO_RX_FLUSH_TIMEOUT	100	Client FIFO receive flush time-out value in milliseconds. This is the amount of time the client waits when its receive FIFO buffer is full before flushing its received data. A value of 0 means that the client never flushes its received data when it is not read by the operating

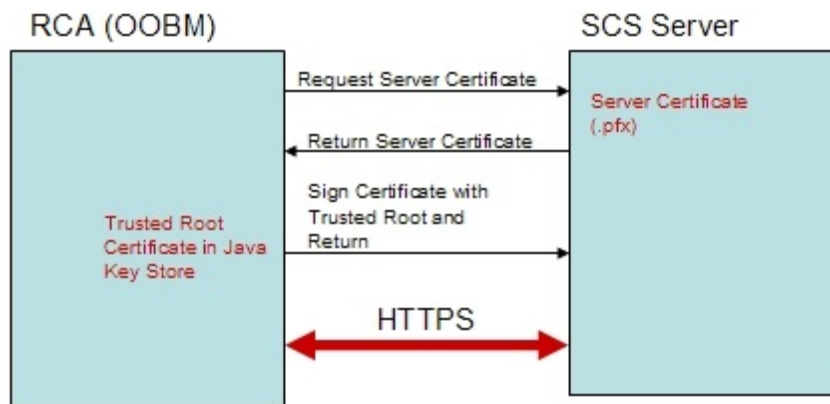
Parameter	Default Value	Description
		system. Minimum value: 0 Maximum value: 65535 Default value: 100 (The default value internal to the Sentient server is 0. Use of a value below 100 is not recommended. A value of 0 causes the client to not flush the received data. As a result, if the buffer overflows, the client will cancel the session.)
SOL_THREADS_SLEEP_TIME	500	SOL thread sleep time
USB_BOOT_SOURCE_VALUES	USB	Boot source names for USB
WSMAN_MAX_ENUMERATION_RECORDS	5	Maximum number of elements that can be fetched on a single WSMAN Enumeration or Pull call
WSMAN_TIMEOUT	30000	Time out for WSMAN calls (maximum milli seconds a WSMAN call can wait for a response)
DEVICE_DELETED_IN_SINGLE_QUERY	30	A single SQL query used to perform delete operation on multiple devices.

Note: The values you give to the *BOOT_SOURCE_VALUES parameters are used in the Sentient server GUI to provide user friendly names for these boot devices. If you do not provide values for these parameters, you may see some non-intuitive text strings displayed representing these boot devices. The string value you provide must be based on the boot source output of the DASH device. For example, if the boot source output is `BRCM:CD/DVD:3`, the user must specify the boot source as `CD/DVD` (*not* `CD`) to see `CD/DVD` in the GUI.

Configuring Secure Access between OOBM Service and SCS

You can provide secure communication between the SCS server and the OOBM Service running on the Sentient server with this same server certificate. To secure communication between these two components, you must export the trusted root certificate of the Microsoft CA on the SCS server and import it into the Java key store on the Sentient server thus enabling the Sentient server to sign the server certificate to authenticate the SCS server.

Secure Access between OOBM and SCS



Note: If you have already exported the root certificate as part of the TLS mutual authentication setup as described in ["Task 7: Exporting the Root Certificate"](#) on page 385, you do not have to perform this task again.

To export the root certificate, follow the steps in ["Task 7: Exporting the Root Certificate"](#) on page 385.

To import the root certificate into the Java key store, follow the steps in ["Importing the Root Certificate into the Java Key Store"](#) on the next page.

Disabling Secure Access between OOBM Service and SCS

After OOBM installation and configuration, the secure hypertext transport protocol (HTTPS) is enabled between the Out of Band Management Service and the SCS server. HTTPS is enabled because of the following:

- You have configured the SCS path to use HTTPS as part of the OOBM device type settings or by specifying it in the `config.properties` file.
- You have exported the trusted root certificate and imported it into the Java key store on the Sentient server. For more information, see ["Configuring Secure Access between OOBM Service and SCS"](#) on the previous page.

If you want to use HTTP instead of the secure transport protocol, you can disable HTTPS through the IIS Manager.

Note: This is not recommended. If you disable HTTPS, user credentials will no longer be encrypted. They will be transmitted in clear text.

To Disable HTTPS

1. Open the IIS Manager on the SCS machine.
2. In the navigation panel on the left-hand side of the window, navigate to **Web Sites > Default Web Site > AMTSCS**. `AMTSCS` is the virtual directory in the SCS URL.
3. Right-click **AMTSCS** and select **Properties** from its context menu. The AMTSCS Properties window opens.
4. Select the **Directory Security** tab.
5. In the **Secure communications** section at the bottom of the window, click **Edit**. The Secure Communications window opens.
6. Clear the **Require secure channel (SSL)** check box at the top of the window.
7. Click **OK** twice and exit the IIS Manager.
In the `config.properties` file, change the value for the `scsserver_url` parameter to one that specifies the HTTP protocol in its URL.
8. Restart the RCA Application Server service.

Importing the Root Certificate into the Java Key Store

For TLS authentication, it is necessary to import the trusted root certificate into the Java key store of the Sentient server. This is used by OOBM to authenticate vPro devices. You exported this root certificate as a `.cer` file in the "Task 7: Exporting the Root Certificate" on page 385.

Note: If you have already imported the root certificate into the Java key store of the Sentient server to secure communication between OOBM and the SCS server as described in the "Configuring Secure Access between OOBM Service and SCS" on page 406, you do not have to perform this procedure again.

To Import the Root Certificate into the Java Key Store

1. On the Sentient server, backup the existing trusted certificate file. This is the `cacerts` file which is typically located in `<InstallDir>\jre\lib\security`.
2. Run the following command from the JRE directory. The default `JRE` directory is `C:\Program Files\PSL\RCA\jre\bin`.
`keytool -import -noprompt -alias customcacert -keystore ..\lib\security\cacerts -storepass <store-password> -file <ca_file.cer>`
 - The `<store-password>` is the password of the certificate store. By default, this password is **changeit**.
 - The `<ca_file.cer>` is the full path to the root certificate you exported in "Task 7: Exporting the Root Certificate" on page 385 and copied over to the Sentient server (if it is different from the SCS server machine).

This command imports the root certificate into the `cacerts` store.

3. For verification, compare the size of the new `cacerts` file with the backed up version. The new file will be larger by 1 or 2 KB.

Note: The location of the `cacerts` file and the JRE bin directory can vary if the application has been installed in a non-default location. The default installation directory for the Sentient server is `C:\Program Files\PSL\RCA`.

4. Stop the RCA Application Server service.
5. Run the following commands from the Sentient Application Server bin directory. The default Sentient Application Server bin directory is `<InstallDir>\applicationserver\bin\service`.
 - a. `wildfly-service.exe //US//RCA-applicationserver ++JvmOptions "-Djavax.net.ssl.keyStore= <Client Certificate Path in PFX format>"`
 - b. `wildfly-service.exe //US//RCA-applicationserver ++JvmOptions "-Djavax.net.ssl.keyStoreType=pkcs12"`
 - c. `wildfly-service.exe //US//RCA-applicationserver ++JvmOptions "-Djavax.net.ssl.keyStorePassword=<PFX client certificate password>"`

Note: The *Client Certificate Path in PFX format* and *PFX client certificate password* are same as specified when "Configuring Security Parameters" on page 398.

6. Start the Sentient Application Server service.

Converting Certificates to PEM Format

Note: This step is required only if you have TLS configured.

For IDE-R and SOL sessions to be secure when TLS is turned on, the certificates must be available in PEM format on the Sentient server. The root certificate was exported as a `.cer` file as described in "Task 7: Exporting the Root Certificate" on page 385. The client certificate was exported as a `.pfx` file as described in the "Task 6: Exporting the Client Certificate" on page 384. These files were copied over to the Sentient server if this machine is different from the SCS server machine where the certificates were exported.

Converting the Root Certificate to PEM Format

Run the following command from the `OpenSSL` directory. The default `OpenSSL` directory is `<InstallDir>`

```
\Media\oobm\win32\sca\CertGenerator\OpenSSLRCA\Media\oobm\win32\sca\CertGenerator\OpenSSL.
```

```
openssl x509 -inform DER -outform PEM -in <root.cer> -out <root.pem>
```

where `<root.cer>` is the Microsoft CA server root certificate in the `.cer` format and `<root.pem>` is Microsoft CA server root certificate in the `.pem` format.

Example:

```
openssl x509 -inform DER -outform PEM -in C:\SCS\RootCA.cer -out  
C:\SCS\RootCA.pem
```

Converting the Client Certificate to PEM Format

Run the following command from the OpenSSL directory. The default OpenSSL directory is `<InstallDir>`

```
\Media\oobm\win32\sca\CertGenerator\OpenSSLRCA\Media\oobm\win32\  
sca\CertGenerator\OpenSSL.
```

```
openssl pkcs12 -in <client.pfx> -out <client.pem>
```

where `<client.pfx>` is the client certificate exported to the Sentient Server in the .pfx format and `<client.pem>` is the client certificate in .pem format to the Sentient server converted from the .pfx format.

Example:

```
openssl pkcs12 -in C:\SCS\ClientAuth.pfx -out C:\SCS\ClientAuth.pem
```

Use Case Scenarios

This chapter explains how you can use the Sentient Console to perform some standard scenarios when managing OOB devices. These scenarios take into account how you can discover assets on devices, perform various heal functions, and protect vPro devices on your network from malware attacks. You can use the Sentient Console to remotely manage devices regardless of their power state, the health of their operating systems, or the existence of management agents. These scenarios are not intended to be a complete, exhaustive representation of how you will use the Out of Band Management (OOBM) features in the Sentient Console in your enterprise but rather serve as illustrative examples.

The following use cases are described in this section:

- ["Hardware Failure and Replacement" below](#)
- ["Operating System Failure and Reboot" on page 412](#)
- ["Virus Infection Detection and Quarantine" on page 413](#)
- ["Device Quarantine and Remediation" on page 416](#)
- ["Monitoring Critical Software" on page 419](#)
- ["Worm Infection and Containment" on page 426](#)

Note: Most of the use cases described here are relevant to vPro devices. DASH devices are mentioned wherever applicable. However, the navigation and procedural details may vary for DASH devices. For detailed procedural information, see the *Sentient Enterprise Console Online Help*.

Hardware Failure and Replacement

This use case is divided into the following sections:

- ["Overview" below](#)
- ["Use Case Process" below](#)

Overview

A hardware sensor type failure occurs. The failure causes the Event Manager residing on the vPro chip to raise an event. Based on the information in the Event Filter, the Event Manager sends an event alert to the Sentient Console.

The administrator wants to

- Subscribe to vPro devices for event alert notification (subscription must already be in place before the hardware failure occurs).
- Discover the hardware assets on the vPro or DASH device to get the correct replacement part.

By subscribing to the vPro device for alert notification, the event alert is automatically sent to the Sentient Console. The administrator then looks up the hardware inventory for that device and orders the exact replacement part to get the machine up and running again.

Use Case Process

Complete the following tasks to implement this scenario:

- ["Subscribe for Event Alert Notifications" below](#)
- ["View Alerts" below](#)
- ["View Hardware Assets" on the next page](#)

Subscribe for Event Alert Notifications

To subscribe to vPro devices for event alert notifications, complete the following steps:

1. On the **Operations** tab under **Out of Band Management** in the left navigation menu, click **Device Management**. The Device Management window opens.
2. Select the devices that you want to subscribe to by selecting the check box for the device or by checking the select all check box in the upper left.
3. From the  alert subscription management icon pull-down list, select **Subscribe to Alerts**.

For more information on alert subscription, see the *Sentient Enterprise Console Online Help*.

View Alerts

To view alerts, complete the following steps:

1. On the **Operations** tab under **Out of Band Management** in the left navigation menu, click **Alert Notifications**. The Alert Notifications window opens.
2. View the alerts of interest that have been sent to the Sentient Console. In this case, you are specifically looking for event alerts caused by hardware failures.

For more information, see "Viewing Alerts on the vPro Device" in the *Sentient Enterprise Console Online Help*.

View Hardware Assets

To view hardware assets, complete the following steps:

1. On the **Operations** tab under **Out of Band Management** in the left navigation menu, click **Device Management**. The Device Management window opens.
2. Click the hostname link for the vPro device with the hardware failure.
3. Click the **Device Assets** link in the Device column of the table under **Diagnostics** on the left-side of the window.
4. Click **Hardware Information**.
5. Click the failed hardware component. Specifications for that component are displayed in the content area of the console.

You can then order a replacement part based on the information obtained from the vPro or DASH device through the Sentient Console.

Operating System Failure and Reboot

This use case is divided into the following sections:

- ["Overview" below](#)
- ["Use Case Process" below](#)

Overview

The operating system on a vPro or DASH device is not responding. The administrator is notified by the user of the PC.

The administrator wants to:

- Lock the front panel of the remote vPro device (in this example, it is assumed that this is supported on the vPro device) so that there is no user interference while performing the remote power operation. The front panel setting feature is not available on DASH devices.
- Reboot the PC from an operating system image file that is on the Sentient Console Server to further diagnose the problem.

Use Case Process

Complete the following tasks to implement this scenario:

- ["Lock the Front Panel on the vPro Device" on the next page](#)
- ["Reboot the System with IDE-R CD Drive" on the next page](#)

Lock the Front Panel on the vPro Device

To lock the front panel on the vPro device, complete the following steps:

1. On the **Operations** tab under **Out of Band Management** in the left navigation menu, click **Device Management**. The Device Management window opens.
2. Click the hostname link in the Device column for the OOB device with the operating system failure.
3. Click the **Front Panel Settings** link under **General Settings** on the left-side of the window.
4. Click the **here** link to enable the **Front Panel Settings** section of the dialog.
5. Ensure that the settings for the keyboard and power button are set to **Yes**.

Reboot the System with IDE-R CD Drive

To reboot the system with IDE-R CD Drive, complete the following steps:

1. On the **Operations** tab under **Out of Band Management** in the left navigation menu, click **Device Management**. The Device Management window opens.
2. Click the hostname link for the vPro device with the operating system failure.
3. Click the **Remote Operations** link under **Diagnostics** on the left-side of the window.
4. In the Remote Operations Wizard, select **Reboot to IDE-R** for the Remote Operation and select **Yes** for Apply Front Panel Settings option.
5. Enter the path to an ISO file that is on the management console server in the Drive Path field.
6. Follow the remaining steps in the wizard.

Virus Infection Detection and Quarantine

This use case is divided into the following sections:

- ["Overview" below](#)
- ["Use Case Steps" on the next page](#)

Overview

A virus attack is suspected because the vPro device has detected network traffic that matches the rate limit filter in the currently active System Defense policy. The match causes the Event Manager residing on the vPro chip to raise an event. Based on the information in the Event Filter, the Event Manager sends an event alert to the Sentient Console.

The administrator wants to:

- Subscribe to vPro devices for event alert notification.
- Create a quarantine policy with the necessary filter so that the virus is contained on the infected device and not permitted to spread to the other devices on the network.
- Enable, activate, and deploy the policy to the infected vPro device.
- Repair, replace, or remove the device.
- Disable the policy when the device is no longer a threat.

When the policy is created, deployed, and activated, the vPro device inspects packets and performs the actions specified by the filters associated with the Quarantine policy. In this case, the filter will drop all TCP packets that are transmitted by this device. After the machine is isolated from the network, the administrator performs the remediation tasks required to restore the vPro device and then disables the quarantine policy.

Use Case Steps

Complete the following tasks to implement this scenario:

- ["Subscribe for Event Alert Notifications" below](#)
- ["View Alerts" below](#)
- ["Create the Quarantine Filter" below](#)
- ["Create the Quarantine Policy" on the next page](#)
- ["Enable, Activate, and Deploy the Quarantine Policy" on the next page](#)
- ["Deactivate the Quarantine Policy" on page 416](#)

Subscribe for Event Alert Notifications

To subscribe to vPro devices for event alert notifications, follow the steps in ["Subscribe for Event Alert Notifications" on page 411](#) in use case ["Hardware Failure and Replacement" on page 410](#).

View Alerts

To view alerts, follow the steps in ["View Alerts" on page 411](#) in use case ["Hardware Failure and Replacement" on page 410](#). In this case, you are specifically looking for event alerts triggered by a rate limit filter.

Create the Quarantine Filter

To create the quarantine filter, complete the following steps:

1. On the **Configuration** tab under **Out of Band Management > vPro System Defense Settings** in the left navigation menu, click **Filters**. The Filters window opens.
2. Click the  add icon on the toolbar. The System Defense Filter wizard opens.

3. Click **Next** to continue. The Filter Details window opens. In this window, specify the following:
 - **Filter Name:** Enter **Quarantine**.
 - **Filter Type:** Select **Drop**.
 - **Create Event on Filter Match:** Select **Yes**.
4. Click **Next**. The Parameters window opens. In this window, specify the following:
 - **Packet Type:** Select **TCP**.
 - **Filter Mode or Direction:** Select **Transmit**.
 - **Network Address:** Select **Filter Packets to Network**.
 - **Port Range:** Select **Source Port Range**. Enter **1** and **655535** for the **Min Port** and **Max Port** values. This refers to the ports on the vPro device. You want to block packets from all source ports on the vPro device to all destination ports on all of the devices in the network to prevent the vPro device from infecting the other devices.
5. Click **Next** and **Close**.

Create the Quarantine Policy

To create the quarantine policy, complete the following steps:

1. On the **Configuration** tab under **Out of Band Management > vPro System Defense Settings** in the left navigation menu, click **Policies**. The Policies window opens.
2. Click the  add icon on the toolbar. The System Defense Policy wizard opens.
3. Click **Next** to continue. The Policy Details window opens. In this window, specify the following:
 - **Policy Name:** Enter **Quarantine**.
 - **Priority:** Enter **98**.
 - **Enable Anti Spoofing Filter:** Select **Yes**.
 - **Default Receive (Rx) Filter Type:** Select **Pass**.
 - **Default Transmit (Tx) Filter Type:** Select **Pass**.
4. Click **Next** to see all available filters.
5. Drag the Quarantine filter to the **Filters to assign to policy** list.
6. Click **Add Policy**.

Enable, Activate, and Deploy the Quarantine Policy

To enable, activate, and deploy the Quarantine policy to the vPro device, complete the following steps:

1. On the **Configuration** tab under **Out of Band Management > vPro System Defense Settings** in the left navigation menu, click **Policies**. The Policies window opens.
2. Check the box next to the Quarantine policy.

3. Click the  deploy icon on the toolbar. The Policy Deployment wizard opens.
4. Click **Next** to continue. The Select Devices window opens.
5. Check the box next to the infected vPro device.
6. Click **Next**. The Set Policies window opens.
7. Select the Quarantine policy as the System Defense Policy for the wired NIC. This enables the Quarantine policy as the System Defense policy for the infected vPro device. Since you specified the highest priority (98) for this policy (with regard to the other System Defense policies currently defined) when you created the policy, it also becomes the active policy for the infected vPro device.
8. Follow the remaining steps in the wizard.

Deactivate the Quarantine Policy

To deactivate the Quarantine policy from the vPro device, complete the following steps:

1. On the **Operations** tab under **Out of Band Management** in the left navigation menu, click **Device Management**. The Device Management window opens.
2. Click the hostname link of the vPro device that was infected with the virus.
3. Click the **Policies** link under the **System Defense** section on the left-side of the window. A window opens displaying the System Defense policies that have been deployed to this device.
4. Check the box next to the Quarantine.
5. Click the  enable/disable icon on the toolbar. The Quarantine is no longer the enabled System Defense policy.

Device Quarantine and Remediation

This use case is divided into the following sections:

- ["Overview" below](#)
- ["Use Case Steps" on the next page](#)

Overview

An administrator needs to quarantine a specific device from the corporate network to prevent any attacks. However, the device will need to connect to a management server to receive the remediation it requires. The remediation may consist of an update to its virus definitions, a new version of its firewall software, or remote control by the administrator into the suspect device.

If all network traffic is blocked, it makes remediation of the device virtually impossible. As a result, the administrator must block all network traffic except to and from the remediation management server, so that the infected device can be repaired.

The administrator wants to:

- Create a remediation policy with the necessary filters so that all network traffic is blocked except to the Sentient Enterprise Server that contains the necessary software to repair the infected device.
- Enable, activate, and deploy the policy to all vPro devices on the network.
- Disable the policy when the device is repaired.

When the policy is created, deployed, and activated, the vPro devices inspect packets and perform the actions specified by the filters associated with the remediation policy. In this case, the filters will pass all TCP and UDP packets that are received from or transmitted to the device whose IP address matches the one specified in the filter. For this example, it is the IP address of the Sentient Enterprise Server. All other packets will be dropped based on the default action of the policy, which is to drop the packets..

Use Case Steps

You want to create 4 filters for the remediation policy. You must complete the procedures listed in this section 4 times, once for each filter you need to create.

- ["Create the Remediation Filters" below](#)
- ["Create the Remediation Policy" on the next page](#)
- ["Enable, Activate, and Deploy the Remediation Policy" on page 419](#)
- ["Deactivate the Remediation Policy" on page 419](#)

Create the Remediation Filters

To create the remediation filters, complete the following steps:

1. On the **Configuration** tab under **Out of Band Management > vPro System Defense Settings** in the left navigation menu, click **Filters**. The Filters window opens.
2. Click the  add icon on the toolbar. The System Defense Filter wizard opens.
3. Click **Next** to continue. The Filter Details window opens. You want to specify the following for the corresponding filters you are creating:

Remediation Filters

Filter Name	Filter Type	Create Event on Filter Match
PassTCP_Recv	Pass	Yes
PassTCP_Xmit	Pass	Yes
PassUDP_Recv	Pass	Yes
PassUDP_Xmit	Pass	Yes

4. Click **Next**. The Parameters window opens. You want to specify the following for the

corresponding filters you are creating:

Remediation Filters

Filter Name	Packet Type	Next Protocol	Filter Mode	Network Address
PassTCP_Recv	IP Packets (IPv4)	TCP	Receive	Device:192.168.5.12
PassTCP_Xmit	IP Packets (IPv4)	TCP	Transmit	Device:192.168.5.12
PassUDP_Recv	IP Packets (IPv4)	UDP	Receive	Device:192.168.5.12
PassUDP_Xmit	IP Packets (IPv4)	UDP	Transmit	Device:192.168.5.12

5. Click **Next** and **Close**.

Note: You could also create the filters to drop all traffic except to or from a specific subnet instead of to or from a single device. This is useful if you have several remediation servers located on a single subnet.

Create the Remediation Policy

To create the remediation policy, complete the following steps:

1. On the **Configuration** tab under **Out of Band Management > vPro System Defense Settings** in the left navigation menu, click **Policies**. The Policies window opens.
2. Click the  add icon on the toolbar. The System Defense Policy wizard opens.
3. Click **Next** to continue. The Policy Details window opens. In this window, specify the following:
 - **Policy Name:** Enter **Remediation**.
 - **Priority:** Enter **98**.
 - **Enable Anti Spoofing Filter:** Select **Yes**.
 - **Default Receive (Rx) Filter Type:** Select **Drop**.
 - **Default Transmit (Tx) Filter Type:** Select **Drop**.
4. Click **Next** to see all available filters.
5. Drag the PassTCP_Recv, PassTCP_Xmit, PassUDP_Recv, and PassUDP_Xmit filters to the **Filters to assign to policy** list.
6. Click **Add Policy**.

Enable, Activate, and Deploy the Remediation Policy

To enable, activate, and deploy the remediation policy to the vPro device, follow the steps in ["Enable, Activate, and Deploy the Quarantine Policy" on page 415](#) in use case ["Virus Infection Detection and Quarantine" on page 413](#). In this case, select the Remediation policy.

Deactivate the Remediation Policy

To deactivate the Remediation policy from the vPro device, follow the steps in ["Deactivate the Quarantine Policy" on page 416](#) in use case ["Virus Infection Detection and Quarantine" on page 413](#). In this case, select the Remediation policy.

Monitoring Critical Software

This use case is divided into the following sections:

- ["Overview" below](#)
- ["View the vPro Event Filter" on the next page](#)

Overview

An OOBM agent is installed and started on a vPro device to monitor a list of security software applications, a monitored security application stops, and the OOBM agent stops sending heartbeats to the watchdog. This transition state causes the watchdog to take the actions that the administrator specified when creating the watchdog. In this case, the watchdog raises an event and enables the Agent Presence policy.

Typically, this type of scenario occurs when users have disabled their anti-virus software thinking it hurts their performance. The security administration wants to enforce a policy that will remove a PC from the corporate network if the anti-virus software is not running.

The administrator wants to:

- Subscribe to the vPro device for event alert notification.
- Define a System Defense policy to isolate the vPro device in the event of OOBM agent failure and set this policy as the Agent Presence policy.
- Create the watchdog and specify its actions.
- Deploy the Agent Presence policy and watchdog to the vPro device.
- Manage OOBM agent settings and deploy the settings to the vPro device.
- Install the OOBM agent on the vPro host operating system (the OOBM agent is started automatically by the installer).
- Restart the security process after the event alert is sent.

After the OOBM agent is installed and started on the vPro device, it registers with the watchdog and starts to send it heartbeats at defined intervals. On agent failure, the agent stops sending heartbeats to the watchdog. The watchdog raises an event and enables the deployed Agent

Presence policy. The Agent Presence policy becomes the active policy based on its higher priority. After the policy is activated, the vPro device inspects packets and performs the actions specified by the filters associated with the Agent Presence policy. The administrator performs the remediation task of restarting the security software that the OOBM agent was monitoring. After the security software is restarted, the OOBM agent registers again and starts to send heartbeats to the watchdog. The watchdog disables the Agent Presence policy and the enabled System Defense policy with the higher priority becomes the next active policy.

Use Case Steps

Complete the following tasks to implement this scenario:

- ["View the vPro Event Filter" below](#)
- ["Subscribe for Event Alert Notifications" on the next page](#)
- ["Create a Filter for the Agent Presence Policy" on the next page](#)
- ["Create the Agent Presence Policy" on the next page](#)
- ["Set and Deploy the Agent Presence Policy" on page 422](#)
- ["Create the Watchdog" on page 422](#)
- ["Deploy the Watchdog" on page 423](#)
- ["Configure the System Message and Software List" on page 423](#)
- ["Deploy the System Message and Software List" on page 424](#)
- ["Install and Start the OOBM Agent" on page 424](#)
- ["Activate the Agent Presence Policy" on page 425](#)
- ["Send the Alert" on page 425](#)
- ["View Alerts" on page 425](#)
- ["Restart the Security Process" on page 425](#)
- ["Deactivate the Agent Presence Policy" on page 426](#)

View the vPro Event Filter

To view the Event Filter for the vPro device, complete the following steps:

1. On the **Operations** tab under **Out of Band Management** in the left navigation menu, click **Device Management**. The Device Management window opens.
2. Click the hostname link for the vPro device whose Event Filter you want to review.
3. Click the **Event Filter** link under the **Diagnostics** section on the left-side of the window. The default event filters that exist on the selected vPro device are displayed in the content area of the console.
4. Click the name link of each event filter to determine which event filter will detect agent failure and send an alert to the management console.

Subscribe for Event Alert Notifications

To subscribe to vPro devices for event alert notifications, follow the steps in the ["Subscribe for Event Alert Notifications"](#) on page 411 in use case ["Hardware Failure and Replacement"](#) on page 410.

Create a Filter for the Agent Presence Policy

To create a filter for the Agent Presence policy, complete the following steps:

1. On the **Configuration** tab under **Out of Band Management > vPro System Defense Settings** in the left navigation menu, click **Filters**. The Filters window opens.
2. Click the  add icon on the toolbar. The System Defense Filters wizard opens.
3. Click Next to continue. The Filter Details window opens. In this window, specify the following:
 - **Filter Name:** Enter **PreventInfection**.
 - **Filter Type:** Select **Drop**.
 - **Create Event on Filter Match:** Select **Yes**.
4. Click **Next**. The Parameters window opens. In this window, specify the following:
 - **Packet Type:** Select **TCP**.
 - **Filter Mode or Direction:** Select **Receive**.
 - **Network Address:** Select **Filter Packets from Network**.
 - **Port Range:** Select **Destination Port Range**. Enter 1 and 65535 for the **Min Port** and **Max Port** values. This refers to the ports on the vPro device. You want to block packets to all destination ports on this vPro device from all source ports on all of the devices in the network to prevent infection on the vPro device.
5. Click **Next** and **Close**.

Create the Agent Presence Policy

To create the Agent Presence policy, complete the following steps:

1. On the **Configuration** tab under **Out of Band Management > vPro System Defense Settings** in the left navigation menu, click **Policies**. The Policies window opens.
2. Click the  add icon on the toolbar. The System Defense Policy wizard opens.
3. Click Next to continue. The Policy Details window opens. In this window, specify the following:
 - **Policy Name:** Enter **PreventInfection**.
 - **Priority:** Enter **99**.
 - **Enable Anti Spoofing Filter:** Select **Yes**.

- **Default Receive (Rx) Filter Type:** Select **Pass**.
 - **Default Transmit (Tx) Filter Type:** Select **Pass**.
4. Click **Next** to see all available filters.
 5. Drag the PreventInfection filter to the **Filters to assign to policy** list.
 6. Click **Add Policy**.

Set and Deploy the Agent Presence Policy

To set and deploy the Agent Presence policy to the vPro device, complete the following steps:

1. On the **Configuration** tab under **Out of Band Management > vPro System Defense Settings** in the left navigation menu, click **Policies**. The Policies window opens.
2. Check the box next to the PreventInfection policy.
3. Click the  deploy icon on the toolbar. The Policy Deployment Wizard opens.
4. Click **Next** to continue. The Select Devices window opens.
5. Check the box next to the vPro device that will be running the OOBM agent.
6. Click **Next**. The Set Policies window opens.
7. Select the **PreventInfection** policy as the Agent Presence Policy for the wired NIC. This sets the PreventInfection policy as the Agent Presence policy for the vPro device. This policy will be enabled by the watchdog if the OOBM agent stops sending heartbeats to the watchdog. Since you specified, the highest priority (99) when you created the policy, it will become the active policy if enabled by the watchdog.
8. Follow the remaining steps in the wizard.

Create the Watchdog

To create the watchdog, complete the following steps:

1. On the **Configuration** tab under **Out of Band Management > vPro System Defense Settings** in the left navigation menu, click **Watchdogs**. The Watchdogs window opens.
2. Click the  add icon to create an watchdog. The Watchdog wizard opens.
3. Click **Next** to continue. In this window, specify the following:
 - **Agent Type:** Select **OOBM agent**.
 - **Name:** Enter **AlphaWatchdog**.
 - **Agent GUID:** This field is grayed out for the OOBM agent because the GUID for the OOBM agent is known.
 - **Heart Beat Interval:** Accept the default value provided.
 - **Startup Interval:** Accept the default value provided.
4. Click **Next**. The Watchdog Actions page of the wizard opens. In this window, specify the following:

- **Transition States:**
From: Select **Agent Running**.
To: Select Agent **Stopped**.
 - **Actions:**
For **Agent Presence**, select **Enable** to specify that you want the Agent Presence policy enabled if the specified agent transition occurs.
For **Event Creation**, select **Enable** to specify that you want an event to be raised if the specified agent transition occurs.
5. Click **Add Action**. The action is added to the actions table at the bottom of the window.
 6. Add actions for another transition state. Now in this window, specify the following:
 - **Transition States:**
From: Select **Agent Stopped**.
To: Select **Agent Running**.
 - **Actions:**
For **Agent Presence**, select **Disable** to specify that you want the Agent Presence policy disabled if the specified agent transition occurs.
For **Event Creation**, select **Enable** to specify that you want an event to be raised if the specified agent transition occurs.
 7. Click **Add Action**. The action is added to the actions table at the bottom of the window.
 8. Follow the remaining steps in the wizard.

Deploy the Watchdog

To deploy the watchdog to the vPro device, complete the following steps:

1. On the **Configuration** tab under **Out of Band Management > vPro System Defense Settings** in the left navigation menu, click **Watchdogs**. The Watchdogs window opens.
2. Select the check box next to AlphaWatchdog.
3. Click the  deploy watchdog icon on the toolbar. The Watchdog Deployment Wizard opens.
4. Click **Next** to continue. The Select Devices window opens.
5. Select the check box next to the vPro device that will be running the OOBM agent.
6. Follow the remaining steps in the wizard.

Configure the System Message and Software List

To configure the system message and software list, complete the following steps:

1. On the **Configuration** tab under **Out of Band Management > vPro System Defense Settings** in the left navigation menu, click **Watchdogs**. The Watchdogs window opens.
2. Click the  OOBM agent settings icon. The Software List Dialog opens.
3. Accept the default message provided in the **System Message** text box.
4. In the **Software Name** box, enter **symantec.exe**.

5. Click **Add**. You can repeat this process to create a list of software applications that you want the OOBM agent to monitor.
6. Click **Save**. An information message is displayed to the screen.
7. Click **Close** to exit the dialog. The system message and agent software list are stored in the XML repository.

Deploy the System Message and Software List

To deploy the system message and software list, complete the following steps:

1. On the **Configuration** tab under **Out of Band Management > vPro System Defense Settings** in the left navigation menu, click **Watchdogs**. The Watchdogs window opens.
2. Click the  deploy software list and system message icon. The Software Deployment Wizard opens.
3. Click **Next**. The Software Titles window opens.
4. Select the **symantec.exe** software application.
5. Click **Next**. The Devices window opens.
6. Select the check box next to the vPro device that will be running the OOBM agent.
7. Click **Next** and follow the remaining steps of the wizard.

Install and Start the OOBM Agent

To install and start the OOBM agent on the vPro device, complete the following steps:

1. Copy the `oobmclocalagent.msi` file located in the `Media\oobm\win32\LocalAgent` directory on the Sentient Core distribution media to the vPro device. Double-click the file. Alternatively, you can also copy the `setup.cmd` file located in the same directory on the distribution media to the vPro device. Double-click the setup file or type `setup.cmd` on the command line. The `setup.cmd` file calls the `oobmclocalagent.msi` file.
2. Click **Next** and accept the license agreement.
3. Click **Next**. The Remote Configuration Parameters window opens. In this window, specify the following:
 - **SCS Configuration Client User Name:** Enter the user name of the user with the role of configuration client. In our example, it is `SCSUser@vlan1.xx.com`.
 - **SCS Configuration Client Password:** Enter the password for the SCSUser. For more information about this role, see ["Configuration Client Role" on page 390](#).
 - **SCS Profile ID:** Enter the profile ID for the vPro device. You can find this information in the Profiles area of the SCS Console.
 - **SCS Remote Configuration URL:** Enter the URL path including the virtual directory for the Intel Setup and Configuration Service (SCS) web services. In our example it is `https://provisionserver.vlan1.hp.com/amtscs_rcfg`, where `provisionserver.vlan1.hp.com` is the fully qualified domain name (FQDN) of the IIS host machine and `amtscs_rcfg` is the SCS web services virtual directory on the host machine.

4. Click **Next**. The User Information window opens, which enables you to enter the vPro administrator credentials. In this window, specify the following:
 - **User Name**: Enter the vPro username of the administrator.
 - **Password**: Enter the vPro password of the administrator.
 - Do not select the **TLS Mode** check box since in this use case, the vPro device is not provisioned in TLS mode.
5. Click **Next** and follow the remaining steps in the install wizard.

The RCA Out of Band Management service starts as soon as the OOBM agent is installed.

Note: All of the applications on the software list that the OOBM agent will monitor must be running when the OOBM agent starts. If not, the OOBM agent will shut down the watchdog as soon as it starts.

Activate the Agent Presence Policy

To activate the Agent Presence policy on the vPro device, complete the following steps:

The watchdog automatically enables the Agent Presence policy because you specified this action for the running to stopped transition state when you created the watchdog.

Also, since you defined the priority of the Agent Presence policy at 99 when you created the System Defense policy and set it as the Agent Presence policy, it has a higher priority than the current active System Defense policy and thus automatically becomes the new active policy.

Send the Alert

The watchdog automatically raises an event because you specified this action for the running to stopped transition state when you created the watchdog.

The default event filter for this type of event specifies to both log an event and send an alert, the event alert will automatically be sent to the management console if you have subscribed to the vPro device.

And finally, since you did subscribe to the vPro device running the OOBM agent for event alert notification, the Event Filter now has a known destination for the alert so that it can be sent to the management console.

View Alerts

To view alerts, follow the steps in ["View Alerts" on page 411](#) in use case ["Hardware Failure and Replacement" on page 410](#). In this case, you are looking for event alerts caused by OOBM agent failure.

Restart the Security Process

To restart the security process, enter the command line or double-click the executable file that invokes the security application.

Deactivate the Agent Presence Policy

After the OOBM agent starts sending heartbeats again to the watchdog, the watchdog automatically disables the Agent Presence policy based on its defined actions for the stopped to running transition state, and the enabled System Defense policy with the higher priority becomes the new active policy.

Worm Infection and Containment

This use case is divided into the following sections:

- ["Overview" below](#)
- ["Use Case Steps" on the next page](#)

Overview

Although firewalls and intrusion detection systems are already in place on the network, the administrator is aware of the fact that these mechanisms are useful only against known worms and are not effective against zero-day worm outbreaks. A heuristics worm-containment system is required to protect the network from such outbreaks.

The administrator wants to:

- Subscribe to vPro devices for event alert notification.
- Create a heuristics specification defining the threshold (packet count) and window size (time) values that will trigger containment actions.
- Specify the actions to be taken if the Heuristics Rules engine residing on the vPro chip detects network traffic that may indicate worm infestation based on the threshold and window size values.
- Specify the time-out value for the containment actions taken to stay in affect.
- Deploy the heuristics specification to the vulnerable vPro devices.
- Perform the necessary remediation tasks after being alerted of a possible outbreak.

When the heuristics information is deployed to the vPro device, the Heuristics Rules engine counts packets and updates counters. If the heuristics criteria are matched, the engine triggers the actions that the administrator specified. In this use case, it will raise an event and block outbound traffic on the offending port. The criteria match always causes the Event Manager residing on the vPro chip to raise an event. (In this use case, it is assumed that the Event Filters on the vPro devices have an entry for this type of event where event alerting is enabled.

After the machine is isolated from the network, the administrator performs the remediation tasks required to restore the vPro device. The heuristics actions will stay in effect for the length of time specified in the time-out value. When that time has elapsed, the actions will be lifted and the Heuristics Rules engine will resume inspecting traffic flow.

Use Case Steps

Complete the following tasks to implement this scenario:

- "Subscribe for Event Alert Notifications" below
- "Create an Heuristics Specification" below
- "Deploy the Heuristics Specification" on the next page
- "View Alerts" on the next page

Subscribe for Event Alert Notifications

To subscribe to vPro devices for event alert notifications, follow the steps in the section "Subscribe for Event Alert Notifications" on page 411 in use case "Hardware Failure and Replacement" on page 410.

Create an Heuristics Specification

To create an heuristics specification, complete the following steps:

1. On the **Configuration** tab under **Out of Band Management > vPro System Defense Settings** in the left navigation menu, click **Heuristics**. The Heuristics window opens.
2. Click the  add icon to create new heuristics information. The Heuristics wizard opens.
3. Click **Next** to continue. The Heuristics Details window opens. In this window, specify the following:
 - **Settings Type:** Select **Default**.
 - **Parameters**
 - **Name:** Enter `Zero_Worm`.
 - **Fast Packet Count:** Accept the default value of 8.
 - **Fast Time Count:** Accept the default value of 10.
 - **Slow Packet Count:** Accept the default value of 64
 - **Slow Time Count:** Accept the default value of 50 seconds (50000 milliseconds).
 - **Encounter Timeout:** Enter 50 seconds.
 - For more details, see the *Window Size and Threshold Values* and *Containment Actions and Timeout Values* sections in the see the *Sentient Enterprise Console Online Help*.
 - **Actions**
 - **Block TX Traffic:** Select **Offensive Port Only** from the pull-down list.
 - **Policy**
 - **Policy Name:** Do not select a policy name from the pull-down list as System Defense filter is not enabled if the heuristics conditions are met.
4. Click **Next**. The status of the operation is displayed.

5. Click **Close** to exit the wizard. The new heuristics information is displayed in the Heuristics table, and it is added to the repository.

Deploy the Heuristics Specification

To deploy the heuristics specification, complete the following steps:

1. On the **Configuration** tab under **Out of Band Management > vPro System Defense Settings** in the left navigation menu, click **Heuristics**. The Heuristics window opens.
2. Check the box next to the Zero_Worm heuristics specification.
3. Click the  deploy heuristics icon on the toolbar. The Heuristics Wizard opens.
4. Click **Next** to continue. The Select Devices window opens.
5. Check the box next to each device to which you want to deploy the heuristics information.
6. Click **Next**. The Heuristics Setting window opens.
7. Select the Zero_Worm heuristics specification for both the wired and wireless network interfaces on the selected devices.
8. Click **Next**. The Confirmation Summary window opens. Review the information in this window.
9. Click **Next** to continue with the deployment process. The Result window opens displaying the results of the operation.
10. Click **Close** to exit the wizard.

View Alerts

To view alerts, follow the steps in ["View Alerts" on page 411](#) in use case ["Hardware Failure and Replacement" on page 410](#). In this case, you are looking for event alerts caused by the Heuristics Rules engine.

Delayed Remote Configuration of the vPro Device

Delayed configuration is performed when the vPro device could not be provisioned within the Limited Network Access time interval that the network interface is opened after the device is connected to the network.

This section covers the following topics:

- ["Transitioning to Setup Mode" below](#)
- ["Remote Configuration Provisioning Process " on the next page](#)
- ["Delayed Remote Configuration through Sentient Console" on page 431](#)

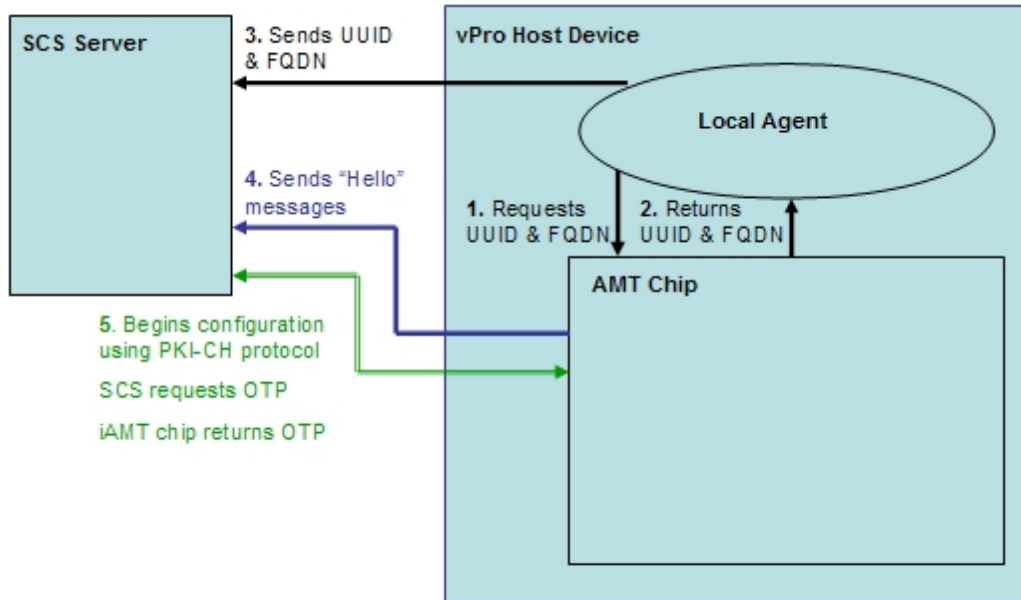
Transitioning to Setup Mode

The following diagram shows the steps involved in transitioning the vPro device to Setup mode when using the OOBM agent for Remote Configuration. This is referred to as *delayed*

configuration because the device was not provisioned immediately after being connected to the network. For an explanation of this alternative (immediate) configuration, see ["Bare Metal Remote Configuration of the vPro Device" on page 389](#).

After the vPro device transitions to Setup mode, it starts sending "Hello" messages to the SCS Server indicating that it is ready to be provisioned.

Transitioning to Setup Mode in Delayed Configuration



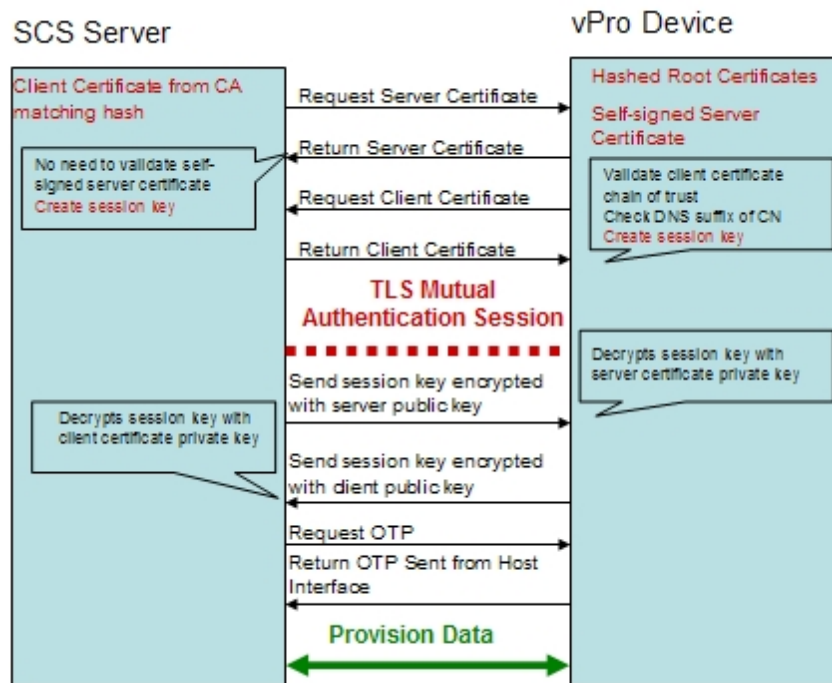
The OOBM agent must be installed on the vPro host device. The OOBM agent detects the vPro device and the following occurs:

1. The OOBM agent requests the UUID and FQDN from the vPro device.
2. The vPro device returns these values to the OOBM agent.
3. The OOBM agent sends these values to the SCS Server.
4. The vPro device starts sending "Hello" messages to the SCS Server.
5. The SCS Server starts the provisioning process using the PKI-CH protocol. The OTP is exchanged between the SCS Server and the vPro device.

Remote Configuration Provisioning Process

During the Remote Configuration provisioning process, a secure channel is created by using TLS mutual authentication. The following diagram illustrates the provisioning process flow.

Remote Configuration Provisioning Process



The provisioning process includes the following steps:

1. The OOBM agent requests that the vPro device start the configuration process. The device opens its network interface for a limited period of time (24 hours on Intel machines and 255 hours on HP desktops) and starts sending “Hello” messages. The interface is opened for the specified hours the first time OOBM is enabled. If the time runs out before setup and configuration completes, any subsequent calls from the OOBM agent to start configuration opens the interface for only six hours.
2. The SCS Server does the following:
 - Extracts the root certificates hashes from the “Hello” message to know what client certificate to send the vPro device for validation.
 - Sends a certificate chain that includes a trusted root certificate matching one of the received hashes.
3. The vPro device does the following:
 - Validates the SCS client certificate. It checks that the OID or the OU is correct and that the certificate is derived from a CA that matches one of the root certificate hashes.
 - Verifies that the domain suffix matches the DNS suffix on the SCS certificate.
4. The SCS Server and the vPro device perform a complete mutual authentication session key exchange.
 - The vPro device uses a self-signed certificate, sending its public key.
 - The SCS creates a TLS session key, encrypts it with the vPro device public key, and sends it to the vPro device.
 - The vPro device decrypts the session key with its private key. The vPro device creates another session key and encrypts the session key with the client public key that the SCS

Server sent to the vPro device for validation. The session key pair is used for the symmetric encryption of traffic during the setup and configuration TLS session.

5. One-time password (OTP) verification takes place between the SCS Server and the vPro device. The SCS server requests the OTP from the vPro device. The device sends the OTP securely, and the SCS Server checks it for correctness.
6. The setup and configuration process continues until the device is provisioned. Since the vPro device network interface is opened for a limited period after sending the first “Hello” message, the SCS Server can specify to the vPro device to extend this period by up to an additional 24 hours to complete the configuration process.

Delayed Remote Configuration through Sentient Console

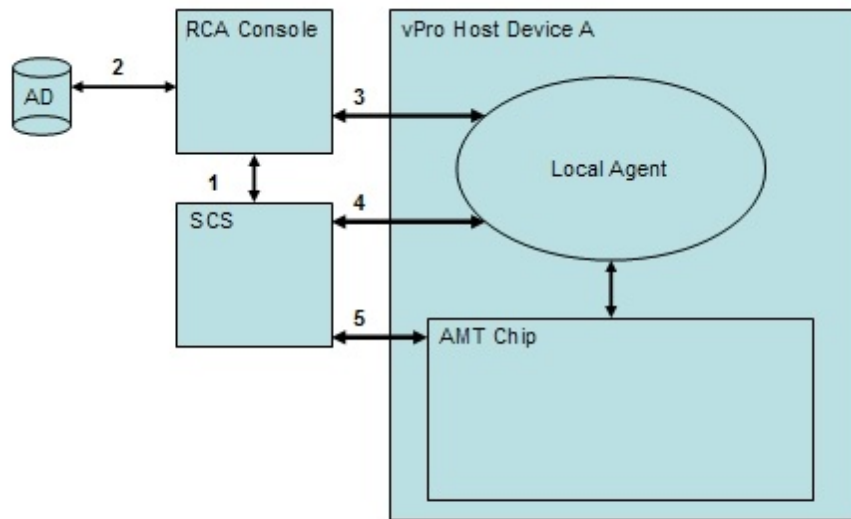
Through the Sentient Console, you can provision vPro devices that were not provisioned during the initial Setup and Configuration Service (SCS) provisioning process.

The type of provisioning performed through the Sentient Console is referred to as delayed Remote Configuration provisioning. It is considered *remote* configuration because you do not have to manually install a PID/PPS pair or enter information about the SCS server address, the domain name, and so on for each device to enable setup. Instead, this provisioning is done automatically and remotely from a management console. It is considered *delayed* configuration because the device was not provisioned in the time interval enabled for that device when it was initially connected to the network.

To use Remote Configuration, you must follow all of the requirements discussed in the ["Configuring the vPro Device Using Remote Configuration" on page 389](#).

Note: You can use Remote Configuration to provision a vPro device only if the vPro device is in the un-provisioned or in-provisioning state. A vPro device, which has already been provisioned, cannot be provisioned again using Remote Configuration. You must re-provision the vPro device manually.

To get the necessary information to provision the vPro devices, the Sentient Console communicates with the OOBM agent running on the vPro device, the SCS server, and the Active Directory.

Delayed Remote Configuration through Sentient Console

The communication exchange is the following:

1. Sentient Console communicates with SCS to get a list of already provisioned devices and devices that are currently being provisioned.
2. Sentient Console communicates with Active Directory (AD) to get a list of devices in that specific domain. The management console lists all of the devices with their respective provisioning state.
3. Sentient Console tries to communicate with the OOBM agent on the default port. If the agent is installed, the devices are displayed with a status of unprovisioned in the Sentient Console. After the communication is established with the OOBM agent, the Sentient Console requests that the OOBM agent start the delayed configuration process.
4. If the device is not provisioned or in the in-provisioning state, the OOBM agent tries to contact SCS server to store the FQDN, UUID and Profile ID of the device. The hello packets are then created.
5. SCS server provisions the vPro device using the PKI-CH protocol.

Trusted Certificates

A secure sockets layer (SSL) connection is secured by using the public key infrastructure (PKI). In PKI, certificates with asymmetric key pairs (public and private) are used to secure communications. The key pair is used to encrypt and decrypt data exchanged between clients and servers when they communicate with each other. The public key is shared and is used to encrypt data. The private key is kept private by the owner of the certificate and is used to decrypt data that was encrypted with the certificate's public key.

When using PKI in server authentication, the client uses the public key of the server certificate to encrypt messages, and the server uses its private key to decrypt messages. Conversely, in client authentication, the server uses the public key of the client certificate to encrypt messages, and the client uses its private key to decrypt messages.

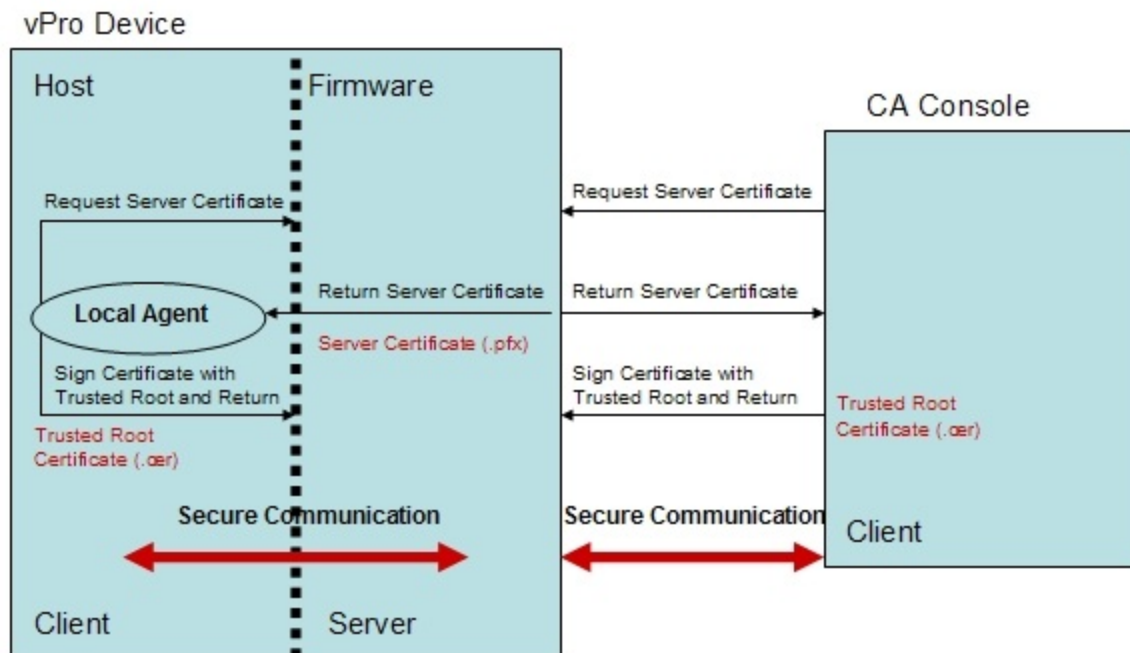
The Transport Layer Security (TLS) protocol has two kinds of authentication, "[TLS Server Authentication](#)" below (one way authentication and "[TLS Mutual Authentication](#)" below (two-way authentication). In the TLS protocol, the firmware on the vPro device is the SSL server. The Sentient Console and/or the OOBM agent running on the host vPro device act as the client.

TLS Server Authentication

When establishing a TLS session in TLS server authentication, the client attempts to verify the validity of the SSL certificate it receives from the firmware on the vPro device. To perform this verification, the client must have the public key of the Certification Authority (CA) that signed the certificate. The public key is available in the trusted root certificate created by the same CA that created the server certificate. The trusted root certificate is already populated on all the vPro systems that are connected to the Active Directory of the domain. The client signs the certificate with the trusted root certificate verifying the identity of the server and sends it back to the server. This secures communications between the components acting as the client and the firmware on the vPro device when the client sends application data to the firmware.

In the following diagram, the OOBM agent running on the host device and the Sentient Console are both clients to the vPro firmware. The functions of the OOBM agent are discussed in greater detail later in this chapter and throughout this guide.

TLS Server Authentication



TLS Mutual Authentication

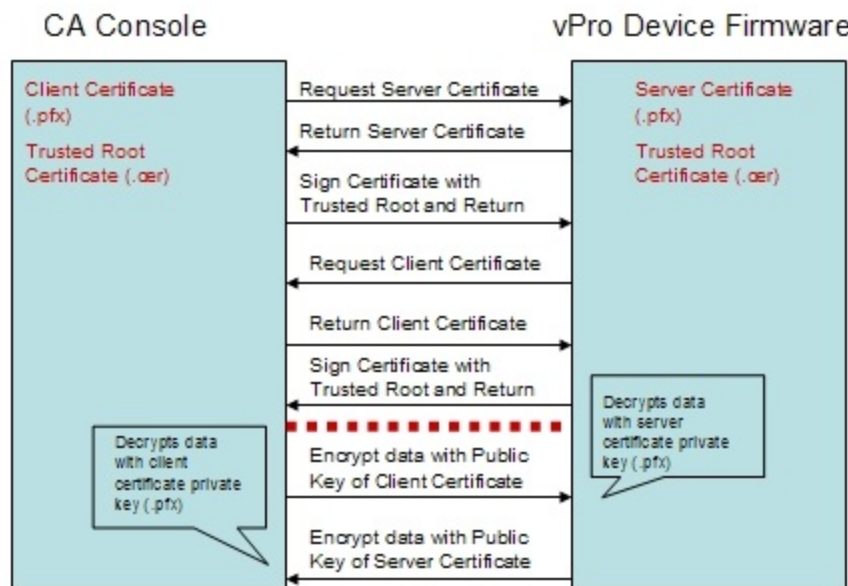
In addition to TLS server authentication where only one certificate is passed between the client and server, TLS mutual authentication provides greater security because two certificates are passed authenticating both ends of the communication. In mutual authentication, the client sends a certificate that must be signed by the server, as well as the server sending a certificate that is

signed by the client. The public and private keys of the certificates are used for data encryption and decryption as described earlier.

Again in this model, the Sentient Console and/or the OOBM agent act as the SSL client. The client must send its own SSL client certificate to the vPro device for client authentication, and the vPro device must have the trusted root certificate (public key) imported into its firmware to perform the verification (signing the client certificate).

When the Sentient Console is the client, the trusted root certificate must also be imported into the trusted key store on the Sentient Console machine. This enables the Sentient Console to sign the server certificate that the vPro device sends it authenticating the server. The client certificate installed on the Sentient Console must contain the complete certificate chain and the private key for the certificate. This feature provides mutual authentication for both the client and the server increasing the security level of TLS sessions. In the following diagram, the Sentient Console is acting as the client to the vPro firmware.

TLS Mutual Authentication between Sentient Console and vPro Device Firmware



There are certain requirements when specifying vPro client certificates. They include the following:

- The certificate must contain the 1.3.6.1.5.5.7.3.2 OID, which marks it as a TLS certificate.
- The Enhanced Key Usage OID list field of the leaf certificate must contain the 2.16.840.1.113741.1.2.1 OID. This OID is used by the vPro device to authenticate the Sentient Console.

You will use these values in the procedure for creating server and client certificate templates. To use the mutual authentication capability, the vPro device must have the root certificate that signed the SSL client certificate in its trust list. The root certificate is provided to the vPro device during the setup and configuration process. This is described in ["Creating and Configuring the Profile" on page 387](#).

Chapter 9

RCA Application Manager Agent

Sentient Application Management Agent

The Sentient Agent is used to manage devices in your environment. It has several sub-features that perform a variety of functions. The following table lists and describes the sub-features of the Sentient agent.

Sentient Agent sub-features

Sentient Agent sub-feature	Description
Sentient Application Manager	Use this sub-feature to distribute mandatory applications throughout the enterprise. This sub-feature is described in this guide.
Sentient Application Self-Service Manager	With this sub-feature, users can install, remove, and update optional applications that are available to them in a service list. This sub-feature is described in this guide. Note: Sentient Application Self-Service Manager is not supported on thin clients.
Sentient Inventory Manager	This sub-feature enables you to collect hardware information and send it to Sentient Inventory Manager for collection and reporting.
Local AIP Extension	For information on this sub-feature, see " Local AIP Support for the MSI Redirector " on page 488.
Sentient OS Manager	This sub-feature controls the provisioning of operating systems.
Sentient Patch Manager	This sub-feature analyzes and manages security patches.
PlusHP	This sub-feature is used for hardware device monitoring specific to HP hardware devices. When enabled, it installs HP System Software Manager (HP SSM) and HP Client Management Interface (HP CMI) components on the target device. It provides Self-Monitoring, Analysis, and Reporting Technology (S.M.A.R.T.) Drive Alert Monitoring and HP CMI based HP Hardware Alert Monitoring. S.M.A.R.T. and HP CMI are used to monitor and gather hardware related alerts and events for reporting. This includes events such as

Sentient Agent sub-feature	Description
	overheating of the processor, fan stall, any hardware related changes, and so on. For information on the HP CMI alerts, see <i>Sentient Enterprise Console Online Help</i> .
Sentient Agent Lockdown Mode	This sub-feature prevents non-privileged users from tampering with critical system-level content or breaching confidentiality by viewing content they should not have access to.

If you install Sentient Application Self-Service Manager and Sentient Application Manager, you can decide whether an application is mandatory or optional, and specify who controls the installation of the application. By adding Sentient Inventory Manager, you can also discover the hardware and software configurations of Sentient agent computers.

Agent Directories

The initialization settings for the Sentient agents are located in the [NOVAEDM] section in the `NVD.INI` file, on the Sentient agent computer. By default, `NVD.INI` is located in the `IDMLIB` directory.

Note: In an Sentient agent lockdown enabled environment, the `NVD.INI` file is moved from `IDMROOT` to `IDMSYS`.

NOVAEDM Parameters

Parameter	Description
IDMDATA	When Sentient installs software, the Sentient agent temporarily stores compressed files received from the Configuration Server in this directory. Once the files are decompressed and installed on the Sentient agent computer, the compressed files are deleted. The default is <code>C:\Program Files\PSL\RCA\Agent\Lib\Data</code>
IDMLIB	The dynamic directory that stores the objects for the service that is currently being managed. The default is <code>C:\Program Files\PSL\RCA\Agent\Lib</code>
IDMSYS	The directory that stores the Sentient agent executables, such as the <code>.EXE</code> and <code>.DLL</code> files. The default is <code>C:\Program Files\PSL\RCA\Agent</code>
IDMROOT	The base directory for <code>IDMLIB</code> . The default is <code>C:\Program Files\PSL\RCA\Agent\Lib</code>
IDMLOG	The directory in which the Sentient agent logs are stored. The default is <code>C:\Program Files\PSL\RCA\Agent\Log</code>

The initialization settings for the Sentient Application Self-Service Manager for Linux and Mac OS are located in the `.nvdrc` file on the agent device. This is located, by default, in the home directory of the account used to install the agent.

Parameter	Description
IDMDATA	When Sentient installs software, the agent temporarily stores compressed files received from the Configuration Server in this folder. After the files are decompressed and installed on the agent device, the compressed files are erased. Default: <code>/opt/PSL/RCA/Agent/lib/Data/</code>
IDMLIB	Dynamic directory that stores the objects for the service currently being managed. Default: <code>/opt/PSL/RCA/Agent/lib/</code>
IDMSYS	Stores the agent executables, such as .EXE and .DLL files. Default: <code>/opt/PSL/RCA/Agent/</code>
IDMROOT	The base directory for IDMLIB. This is a static path. Default: <code>/opt/PSL/RCA/Agent/lib/</code>
IDMLOG	Stores the agent logs. Default: <code>/opt/PSL/RCA/Agent/log/</code>

The following table explains the directory structure. The directories are preceded with `/opt/PSL/RCA/Agent/`.

Agent directories

Directory (<code>/opt/PSL/RCA/Agent/..</code>)	Description
<code>/CACertificates</code>	SSL Certificates
<code>/lib/MAINT</code>	Maintenance Storage Directory
<code>/lib/BACKUP</code>	Upgrade Maintenance Backup folder
<code>/lib/SYSTEM</code>	Starting Directory (<code>startdir</code>) created during connect (Name will vary).
<code>/lib/SYSTEM/CM-CS</code>	Configuration Server name (<code>mname</code>)
<code>/lib/SYSTEM/CM-CS/SOFTWARE</code>	Directory Name (<code>dname</code>)
<code>/lib/SYSTEM/CM-CS/SOFTWARE/ZSERVICE</code>	ZSERVICE Class
<code>/lib/SYSTEM/CM-CS/SOFTWARE/ZSERVICE/DRAGVIEW</code>	Sample application directory

Agent Objects

Sentient agent objects are stored in the IDMLIB directory on the Sentient agent computer. When an Sentient agent connects to a Configuration Server, an information exchange (called resolution) takes place, during which Sentient checks the status of services, and updates the Configuration Server with information from the Sentient agent objects.

Sentient agent objects can be used to:

- determine the hardware configuration of the Sentient agent computer
- check the status of a service, for example, check if the service is successfully installed

- determine when the service was installed
- determine the Sentient agent computer's name and the most recently logged on user
- determine the possible data sources for the Sentient agent computer

While there are multiple Sentient objects on an Sentient agent computer at any time, there is a core group of five Sentient agent objects that supply information about the status of the current Sentient agent connect.

ZCONFIG (Hardware Configuration Information)

This object is created at the start of the Sentient agent connect process and contains basic hardware information such as processor, operating system, and drives.

Note: A connect is the Sentient agent connecting to a Configuration Server in order to perform resolution and achieve its desired state, see [""Sentient Application Management Agent"" on page 435](#)".

Use the Sentient Administrator Agent Explorer to view the ZCONFIG object. The following table describes the attributes of ZCONFIG arranged in alphabetical order. These attributes could vary depending on the configuration of the Sentient agent computer.

Note: The ZCONFIG object is sent to the Configuration Server automatically for viewing with Sentient Administrator CSDB Editor. If you do not want this object sent to the Configuration Server, set the POLICY.USER.ZCONFIG attribute to **N** in the _BASE_INSTANCE_. This will stop collection from ALL users. The object will still exist on the Sentient agent computer.

ZCONFIG Attributes for Windows

Attribute	Description
BOOTDRV	The boot drive.
BOOTDRVI	The type of boot drive, such as IDE.
DHCPSP0n	The IP address of the DHCP Server of the LADAPT0n adapter.
DHCPSPR	The IP address of the DHCP Server for the current LAN adapter.
DNSDMN01	The name of the domain that is used by the Sentient agent computer for the LADAPT0n adapter.
DNSDOMN	The name of the domain that is currently being used.
DNSHNM01	The host name that is used by the Sentient agent computer for the LADAPT0n adapter.
DNSHOSTN	The host name that is currently being used.
GATEWY01	The Gateway Address of network adapter 1.
HALCOMP	The company of HAL.DLL.

Attribute	Description
HALDATE	The date and time of HAL.DLL.
HALFNAME	The original name of HAL.DLL.
HALFVER	The internal version of HAL.DLL.
HALINAME	The name of HAL.DLL.
HALLANG	The language of HAL.DLL.
HALPNAME	The product name of HAL.DLL.
HALPVER	The product version of HAL.DLL.
HALSIZE	The size of HAL.DLL.
IPADDR01	The IP address of network adapter 1.
LADAPT01	LAN Adapter 1
LASTUSER	The most recent user to have logged on to the system.
REBOOTD	The reboot date.
REBOOTT	The reboot time.
SCANTYPE	The type of hardware scan.
SUBNET01	The Subnet Mask for LADAPT01.
ZGATEWAY	The Gateway Address.
ZGATEWAY	The Gateway Address.
ZHDWARCH	The operating system architecture.
ZHDWBIOS	The BIOS type.
ZHDWCDDR	The CD-ROM drive letter.
ZHDWCOMP	The computer name.
ZHDWCPU	The CPU type.
ZHDWCPUN	The number of CPUs that are installed.
ZHDWCPUS	The CPU speed.
ZHDWCTYP	The computer type (desktop or laptop).
ZHDWD00	The drive name for drive 00.
ZHDWD00C	The drive classification for drive 00.
ZHDWD00F	The current free space on drive 00.

Attribute	Description
ZHDWD00S	The type of file system on drive 00.
ZHDWD00T	The total space for drive 00.
ZHDWD01	The drive name for drive 01.
ZHDWD01C	The drive classification for drive 01.
ZHDWD01F	The current free space on drive 01.
ZHDWD01S	The file system on drive 01.
ZHDWD01T	The total space for drive 01.
ZHDWDF_A	The information for floppy drive A.
ZHDWDLST	The list of assigned drive letters.
ZHDWDNUM	The number of assigned drive letters.
ZHDWFPU	The current FPU type.
ZHDWIPAD	The IP address.
ZHDWKYBD	The keyboard type.
ZHDWLANA	The LAN Adapter.
ZHDWLANG	Language
ZHDWMEM	The total physical memory (RAM).
ZHDWMEMF	The current total free memory (RAM).
ZHDWMOUS	The type of mouse (pointing device).
ZHDWNET1	Network adapter 1 information
ZHDWNNET	The number of network adapters installed.
ZHDWOS	The operating system and version.
ZHDWOSCL	The operating system classification (workstation or server).
ZHDWOSDB	The operating system build.
ZHDWOSOG	The operating system organization.
ZHDWOSOW	The operating system owner.
ZHDWPA00	Printer 00 information
ZHDWPA01	Printer 01 information
ZHDWPPAR	The number of parallel ports.

Attribute	Description
ZHDWPPRN	The number of available printers.
ZHDWPSE	The number of serial ports.
ZHDWSVCP	The applied service pack.
ZHDWVIDEO	The video type.
ZHDWXPAG	The page size.
ZHWCPU01	First CPU type
ZHWFPU01	First FPU type
ZHDWVIE	Microsoft Internet Explorer version
ZHDWVMSI	MSI version
ZHDWVRES	Video resolution
ZMODEM	Modem present?
ZOBJRRC	Resolution return code
ZOBJRSTY	Resolution type
ZUSERID	User ID or computer name

ZCONFIG attributes for Linux and Macintosh

Attribute	Description	Example
RUNLEVEL	Current run level at time of scan	5
ZHDWCOMP	Computer Name	qalinux
ZHDWCPU	CPU type	i686
ZHDWD00	Drive name for drive 00	/dev/hda2
ZHDWD00F	Current free space on drive 00	26913026048
ZHDWD00M	Mount Point for Drive 00	/
ZHDWD00T	Total space for drive 00	35152932864
ZHDWDNUM	Number of drive letters assigned	3
ZHDWMACH	Machine Type	i686
ZHDWMEM	Total physical memory (RAM)	133,619,712
ZHDWOS	Operating system	Linux
ZHDWSVCP	Service pack applied	2.4.20-8

Attribute	Description	Example
ZHDWXHID	Host ID (output of hosted command)	771039E4
ZHDWXHN	Host Name	qalinux
ZOBJNAME	Name of Object	HARDWARE_SCAN
ZOSMAJOR	Major Component of OS version	2
ZOSMINOR	Minor Component of OS version	4
ZOSREV	OS revision (output of <code>uname -v</code>)	#1 Thu Mar 13 17:54:28 EST 2003
ZOSVER	OS version (output of <code>uname -r</code>)	2.4.20-8
ZUSERID	User ID or computer name	LINUXUSER

SYNOPSIS (Client Operation Profile Summary)

This object contains a job summary and is transferred to the Configuration Server at the end of the Sentient agent connect. It reports some of the parameters from the RADSKMAN command line and information on the number of files and bytes added, removed, and repaired.

Note: Client Operations Profiles must be enabled for this object to be present. For more information, see ["SYNOPSIS Attributes" below](#).

The SYNOPSIS object is created on Sentient agents that are using Client Operations Profiles. It summarizes the most recent Sentient agent connect, and can be used to confirm the success or failure of the Sentient agent connect process.

SYNOPSIS Attributes

Attribute	Description
STARTIME	The start time, in ISO8601 time format. For example, 1997-08-15T11:12:00-0400
ENDTIME	The end time, in ISO8601 time format.
EXITCODE	The exit code from the job.
ERRORMSG	The text message corresponding to the EXITCODE described in "Messages and Codes" on page 460 .
PRIORAPP	The total number of applications that existed in the service list (installed/not installed) before this job started.
PRIORINS	The total number of installed applications that existed in the service list before this job was started.
PRIORERR	The total number of applications in the service list that have errors before this job started.

Attribute	Description
CURRAPP	The number of applications in the service list after the job completed.
CURRINS	The number of applications in the service list that have been installed.
UPDNUM	The number of updates found in the service list.
UPDSKIP	The number of updates skipped.
UPDDONE	The number of updates processed.
UPDFAIL	The number of updates that failed.
ADDNUM	The number of new applications found in the service list.
ADDSKIP	The number of installs skipped (possibly optional applications).
ADDDONE	The number of installs processed.
ADDFAIL	The number of installs that failed.
DELNUM	The number of deletes found in the service list.
DELSKIP	The number of deletes skipped.
DELDONE	The number of deletes processed.
DELFAIL	The number of deletes that failed.
VERNUM	The number of applications that were verified.
VERSKIP	The number of verifications skipped.
VERDONE	The number of verifications processed.
VERFAIL	The number of verifications that failed.
REPNUM	The number of applications that were repaired.
REPSKIP	The number of repairs skipped.
REPDONE	The number of repairs processed.
REPFAIL	The number of repairs that failed.
CREFRESH	Catalog Refreshed (Y/N)?
JOBID	The job ID that was passed in on the command line using notify.
ZUSERID	The user ID for this job.
ZCONTEXT	The (machine or user) context of this job.
DNAME (Only for Windows)	The RCA Configuration Server Database domain.

Attribute	Description
MNAME (Only for Windows)	The name of the Sentient Configuration Server.
MACHNAME	The machine name of the Sentient agent computer from which this was run.
STARTDIR (Only for Windows)	Specifies the IDMLIB starting directory
USEREXEC	The user that executed the job.
CMDLINE	The command-line parameters used to execute this job.

SAPSTATS (Service Access Profile Status)

This object is updated by any network-bound modules (such as RADCONCT, RADSTGRQ, and RADSTGMS) that need to access the Server Access Profile (SAP). It has one instance for each Sentient agent computer's SAP. For each SAP it summarizes information such as speed, number of files sent and received, and the role of the SAP. RADSKMAN deletes the SAPSTATS object at the beginning of the job.

Note: Client Operations Profiles must be enabled for this object to be present. For more information, see ["SAPSTATS Object Attributes" below](#).

The SAPSTATS object is generated on Sentient agents that are using Client Operation Profiles, and is used to report the Server Access Profile (SAP) status and usage statistics from the Sentient agent. The SAPSTATS object contains all the variables that are defined in the SAP Class in the CSDB along with the following usage related variables.

SAPSTATS Object Attributes

Attribute	Description
BANDWIDTH	The percent of bandwidth to use (between 1 and 99).
BYTERCVD	The number of bytes received.
BYTESENT	The number of bytes sent.
ENABLED	Is this SAP is enabled (Y N)?
ERRCOUNT	The number of errors.
FILEMISS	The number of files not found.
FILERCVD	The number of files received.
FILESENT	The number of files sent.

Attribute	Description
LASTAXSD	The last date/time accessed, in ISO format.
NAME	The friendly name of the SAP.
OBJRCVD	The number of objects received.
OBJSEND	The number of objects sent.
PRIORITY	The priority for this SAP (obtained from the CLIENT.LOCATION Class instance).
PROXY	The Internet proxy URI through which the Sentient agent will connect to the SAP. This value is maintained by the Sentient agent.
ROLE	The role of the SAP. The valid values are: • O: Client Operations Profiles • M: Self-maintenance • S: Services • R: Reporting • P: Patch Manager Gateway • D: Data • A: All roles
SPEED	The speed to the SAP from the Sentient agent computer measured in bytes per second.
STATUS	The status of this SAP. • 000 = SAP was successfully accessed • 920 = SAP could not be accessed • 999 = SAP was not used
STREAM	Specifies if streaming is used. This overrides the Sentient agent setting in ZMASTER.ZNORSPNS.
THROTYPE	The type of bandwidth throttling used. The valid values are NONE , ADAPTIVE , and RESERVED .
TIMEOUT	The communications timeout, in seconds.
TYPE	The type of SAP. The valid values are: • RCS: Configuration Server • DATA: Proxy Servers, Staging Servers or a CD-ROM.
URI	The Universal Resource Identifier for the SAP.

PREFACE (RADSKMAN Execution)

This object contains core information about each invocation of RADSKMAN and is sent to the Configuration Server at every phase of a RADSKMAN process. For more information, see ["PREFACE Object Attributes" below](#).

PREFACE object is sent to the Configuration Server at every phase of a radskman process including:

- Client Operations Profile resolution
- Self Maintenance resolution
- Catalog resolution
- Single service resolution (This can happen multiple times depending on what services are processed.)
- Outbox flush

At each new phase of the agent connect, the PREFACE object is updated. The variables in the PREFACE object can be used for resolution and reporting. For resolution, use the attributes of the PREFACE object for ZSTOP expressions, for symbolic substitution, and for dispatching messages. For reporting, you can combine MACHNAME, ZUSERID, ZCONTEXT, JOBID, and CTYPE to know which user ran the agent connect, the type of connect, and the context.

PREFACE Object Attributes

Attribute	Description
CMDLINE	The RADSKMAN command-line parameters that were used for the current Sentient agent connect.
COMPDN	The distinguished name of the computer in the Active Directory format. This field will be blank if the system is not part of an Active Directory or a Domain environment. Windows operating systems that do not authenticate to Active Directory would show this as their DomainName/MachineName. Example: CN=ALEE,CN=Computers,DC=usa,DC=asdfoods,DC=com
CTYPE	The type of Sentient agent. The valid values are: • RSM: Sentient Application Self-Service Manager • RAM: Sentient Application Manager • RPS: Proxy Server or Staging Server (for preloading application resources)
JOBID	The job ID that was specified on the command line for this connect
LOCALUID	The starting directory under IDMROOT on the Sentient agent computer. The value is derived from the STARTDIR RADSKMAN parameter. So, if STARTDIR = \$USER, LOCALUID would contain the user's ID. If STARTDIR = SYSTEM, LOCALUID would contain SYSTEM. Note: UID stands for <i>user's initial directory</i> ; not user's identification.

Attribute	Description
MACHNAME	The Sentient agent computer's machine name.
USEREXEC	The user who is currently logged on and who executed the command. For Notify and Timers, this would be SYSTEM. For logon scripts, this would be the subscriber's network account name.
ZCONTEXT	The value of ZCONTEXT as passed on the RADSKMAN command line. • M indicates that RADSKMAN was run in a machine context. • U indicates that RADSKMAN was run in a user context. • A blank indicates that no context was specified on the RADSKMAN command line; the context will default to the context in which the Sentient agent connect was launched.
ZDOMNAME	The CSDB domain that is specified in the DNAME parameter of the RADSKMAN command line. The default is SOFTWARE .
ZMGRNAME	The Configuration Server name that is specified in the MNAME parameter of the RADSKMAN command line.
ZPKGREL (Only for Windows)	This field contains the same value that is found in the Sentient agent's ZMASTER.ZPKGREL. The ZMASTER.ZPKGREL value is set during Agent install process and updated during the deployment of self-maintenance process to indicate the current level of an Agent installation. Sentient Core console uses this value to report the version of the Agent corresponding to a connection from a managed device. The value found in this field is used as the key for policy resolution to determine whether an Agent upgrade service is applicable to an Agent or not.
ZUSERID	This field contains the same value that is found in the Sentient agent's ZMASTER.ZUSERID. In most scenarios, it represents the machine name of the Sentient agent computer, but it could also contain the current user name or another value. The value found in this field is often used as the key for policy resolution or reporting. The UID RADSKMAN parameter sets this value.

SMINFO (Systems Management Information)

This object is created during Client Operations Profiles resolution but it does not require Client Operations Profiles. The SMINFO (Systems Management Information) object is created on all Sentient agent computers. It summarizes hardware-specific information that is independent of the operating system and software that is installed on the Sentient agent computer. Sentient uses SMBIOS standards to access data about the BIOS. SMINFO also includes some network and user ID information. For more information, see ["SMINFO Attributes" on the next page](#).

Note: Unlike the other objects that are discussed in this section, this object is one level lower, under RADSETUP.

SMINFO Attributes

Attribute	Description
ASSETTAG	The Unique Asset Tag number of the Sentient agent computer from the BIOS.
BIOSDATE (only for Windows)	The date of the computer's BIOS.
BIOSVEND (only for Windows)	The vendor of the computer's BIOS.
BIOSVERS (only for Windows)	The version of the computer's BIOS.
COMPDOMN	The computer domain.
COMPNAME	The computer name.
FLASHMEM (only for Windows)	The amount of flash memory on the machine.
IPADDR	The Sentient agent computer's IP address.
MACADDR	The Sentient agent computer's MAC address.
MACHUUID	The unique machine user ID.
SNENCLOS	The serial numbers for the system enclose structures (from the BIOS).
SNSYSTEM	The serial numbers for the system structures (from the BIOS).
SUBMASK	The subnet mask.
SUBNET	The subnet.
SYSMANUF	The system manufacturer information (from the BIOS).
SYSPROD	The system manufacturer product information (from the BIOS).

Controlling Default Permissions for Directories and Objects

Directories, objects, and log files created by Sentient are assigned permissions based on current umask settings and execute permissions on objects and log files are removed. In order to change the default permissions assigned when new directories, objects, and log files are created by

Sentient within `IDMLIB`, you can use environment variables or you can create a `DEFAULTS.EDM` file in `IDMROOT`.

Note that environment variables will always take precedence. If the environment variables are set and a `DEFAULTS.EDM` file exists, values defined using the environment variables are used.

These methods for controlling permissions apply only to newly created, service-related directories and objects within `IDMLIB`. For example,
`/opt/PSL/RCA/Agent/lib/SYSTEM/CMCS/SOFTWARE/ZSERVICE/SAMP_
APP/00000000.000.`

To control permissions using environment variables:

Set the following environment variables with the permissions you want assigned by default:

- For directories: `IDMLIBPERM`
- For objects: `IDMOBJPERM`
- For log files: `IDMLOGPERM`

To control permissions using `DEFAULTS.EDM`:

1. In `IDMROOT` directory, create an object, `DEFAULTS.EDM`.
2. Add the following variables with the permission value to be used when new objects, log files, or directories are created by Sentient.
 - For directories: `LIBPERM`
 - For objects: `OBJPERM`
 - For log files: `LOGPERM`

For example, to exclude write permissions for objects for group and other, create a `DEFAULTS.EDM` file with the following:

```
OBJPERM 0644
```

To exclude write permissions for logs for group and other, create a `DEFAULTS.EDM` file with the following:

```
LOGPERM 0644
```

To set the default permissions of directories to read and write for everybody, create a `DEFAULTS.EDM` file with the following:

```
LIBPERM 0777
```

Logs

The agent has three primary modules: `RADSKMAN`, `RADPINIT`, and `RADCONCT`. However, the activity-reporting of these three modules is shared in one log file, `connect.log` (the default name).

Note: The default location of `connect.log` is `/opt/PSL/RCA/Agent/log`.

Open the `connect.log` file in the `IDMLOG` directory of the host system and, using a text editor, search on the word “version.” You can also check the **Version** tab of the Properties of `RADSKMAN` in the `IDMSYS` directory.

When `connect.log` reaches 1 MB in size, a backup log, `connect.bak` is created.

As stated, `connect.log` and `connect.bak` are the default names given to these logs. You can rename the log using the parameter, `log` in a format that better suits your needs. For example, you might prefer to name your logs based on the date and time. Then, if you notice a problem occurring on a certain date, you can retrieve only the logs that you need to review. Additionally, you have the option of appending information to a log by using the parameter, `alog`.

Each of the three primary agent modules can be instructed to use a specific log file by simply adding the **log** parameter to its command line. The three primary agent modules take command-line parameters in the following format.

Keyword = value (in comma-delimited format)

Use the optional `log` and `alog` parameters on the command line to name the log file and append information to an existing log file, respectively. For example, you could add the **log** parameter to a RADSKMAN command line in a Notify in order to generate a specific log name, as in:

```
radskman log=notify10012003.log
```

Parameters for Log Files

Parameter	Description
log	The name of the log file that is to be created, such as <code>connect.log</code> , the default. Use a valid filename without a path (by default, logs are stored in the <code>IDMLOG</code> folder). If there is a log file with the same name, Sentient creates a backup of that file called <code>logname.bak</code> . If there is an existing <code>logname.bak</code> , it will be overwritten.
alog	The name of the log file to which the information will be appended. For example, <code>alog=Application1.log</code> . Use a valid filename without a path (by default, logs are stored in the <code>IDMLOG</code> folder). This parameter has no default; if it is not specified, the information will be appended to the log file that is named in the <code>log</code> parameter.

The value for the `log` parameter is stored in the `LOGNAME` attribute, which is located in the `ZMASTER` object in the catalog and application directories.

RADSTATE (Diagnostic Module)

RADSTATE is a diagnostic module that gives an overview of the current state of the Sentient agent. The information in the RADSTATE output is based on data that is retrieved from numerous Sentient agent objects.

Usage

The following is a sample of the RADSTATE syntax.

```
RADSTATE
mode=<abcdeimoprsv>, USERNAME=UserJoe, UID=UserID, MNAME=
<ConfigServer>, DNAME=<DB_domain>, SNAME=<service>
```

- **IDMROOT**: use to set `IDMROOT` (optional; defaults to the current `IDMROOT` setting)
- **Mode**: the mode of RADSTATE module.
- **USERNAME**: the user name; used for reporting

- **UID**: the user ID (optional)
- **MNAME**: the name of the Sentient Configuration Server (optional)
- **DNAME**: the Sentient Configuration Server Database domain (optional)
- **SNAME**: the name of the service (optional)

The following table lists and describes the valid values for the mode parameter.

RADSTATE Modes

Mode	Description
a	Display the ZVERLIST object
b	Verify instance data (temporarily disabled)
c	Check for duplicates and conflicts in FILE objects
d	Display an output log in the native editor
e	Check for EDM duplicates and conflicts (valid only with c mode)
i	User/Machine context report
m	Show module information
o	Create objects
p	Display Patch data (by default, this mode is skipped)
r	Display all resources
s	Display service detail
u	Display service user summary
v	Verbose mode
?	Print this help message

When RADSTATE is run with the **Verbose** parameter (mode **v**), it provides basic information regarding the agent environment:

- Global object statistics
- Current date and time
- Current operating system
- Locations of the `IDMSYS`, `IDMLIB`, and `IDMLOG` directories
- Environment settings
- Emulator settings
- Trace levels
- Timeout settings

- All Service status including component totals by User and Service including instance totals and byte totals.
- Timer Information

Use radstate at anytime to check agent configurations. For example, run radstate at the end of each agent connect. After radstate is run using mode option o, the ZRSTATE and ZRSTATES objects are built and can be sent to the Configuration Server as needed.

Radstate should be run:

- Whenever agent-specific information is required.
- If it is suspected that some files may not have deployed correctly.
- If desktop updates have not occurred.

Manual execution of radstate produces a summary style report, radstate.log, regarding the current state of services and resources installed on the agent desktop. Radstate is executed from the command line using the appropriate parameters, separated by a comma, for example:

```
radstate mode=vo, IDMROOT=/opt/PSL/RCA/Agent/lib
```

For a technical document with additional information, see the Persistent Systems support web site.

Method Dispatching

When the client-connect module (RADCONCT) dispatches methods, it creates an object called **ZDSPM000**, which contains the information for the instance that is being instantiated. The methods then read the information that is stored in ZDSPM000 and do their work. When the methods need to convey the results to RADCONCT, they create a **ZMRESULT** object with two variables, **ZMRC** and **ZMMSG**.

- ZMRC contains the extended error information that is found in the subscriber error codes.
- ZMMSG contains a corresponding message.

When the method exits, it will do so with one of the exit codes described in the following table.

Method Exit Codes

Exit Code	Description
0	No errors
4	Warning; continue the process
8	Failure; abort process
16	Fatal error; abort process

Exit Code 4: RADCONCT logs the information that is contained in ZMRC and ZMMSG.

Exit Codes 8 and 16: RADCONCT creates a ZERROR object with ZMRC and ZMMSG.

Lockdown Mode

The goal of the Sentient agent lockdown mode is to ensure the integrity, confidentiality, and availability of the content and methods that are stored and used by the management agent. This prevents non-privileged users from tampering with critical system-level content or breaching confidentiality by viewing content they should not have access to.

In Windows, the operating system enables you to define HOME directories for each user. The administrator configures the user data store to use the HOME directory style approach. The Sentient processing methods and other objects are owned by the SYSTEM user. This was done intentionally so that migration to Lockdown mode is less disruptive to the SYSTEM mode.

To solve the “What to trust” issue, the Sentient agent does not trust any parameters that are passed in from a USER request (for example, RADSKMAN command). The SYSTEM mode is predefined with trusted and secured objects either at install time or by performing a priming connection after installation to the Configuration Server to get the default parameters.

The Sentient agent has a machine (SYSTEM) mode and user mode. The machine mode is controlled by the administrator and has elevated privileges, whereas the user mode managed by end user runs with potentially limited privileges. Typically, an administrator deploys all the standard applications to a device using the machine mode. A user who logs into the system might deploy optional software that has been entitled to him/her by the site administrator.

To install applications on a system, the install process must run with elevated privileges. Since the end user can decide which entitled applications to deploy, the SYSTEM side needs to be available to the user and the site administrator. The Sentient agent uses an object model retrieved from the Configuration Server and stored on the target device to manage software on the systems. These objects are accessed using SYSTEM or end user privileges.

Note: For an application, with ZSERVICE.ZSYSACCT=Y, to work successfully in lockdown mode it must be entitled to both, the USER and MACHINE.

Implementations that choose not to secure their Sentient managed devices folders may be exposed to a variety of issues, including:

- **Local Tampering of Data Store:** The user can influence the machine mode to do unauthorized operations by tampering with the objects stored in `IDMROOT`. The Agent Lockdown Mode segregates the user and machine data stores so that each is only accessible in its authorized processing context. This independently secures the SYSTEM and USER objects.
- **Secure Environment Trust Issues:** Sentient agent in Lockdown Mode only trusts IP addresses that are specified in secured locations, for example, SAP object, and cannot be modified by a non-privileged user. Once a Client Operations Profile (COP) connect has run, there is a secure list of servers that the Sentient agent is allowed to contact.

Administrators may have a variety of reasons for choosing not to deploy security lockdown restrictions on end users:

- Users may all be machine administrators.
- Users may all be trusted users. Another means of security may be employed.

Entitlement Settings

In general, the Sentient Administrators would have the following entitlement scenarios for deploying software into target systems:

- **Machine Only Entitlements:** These are applications, which will only be deployed by using SYSTEM context and the application is owned by the machine level.
- **User Only Entitlements:** These are the applications, which will only be deployed by user context. Hence, entitlements are done in per use per user based policy, and the application is owned by the end-user of the target machine.
- **Machine/User (Hybrid) Entitlements:** These are the kind of applications, which are partly deployed by SYSTEM and partly deployed by USER. Machine Side has to be run first, before running the first user connect. This is called **Priming Connect**. To facilitate the Priming Connects, there is a `MACHDEF.EDM` file, which will be created by collecting the Connect parameters from the Administrator during the install time.

Machine Defaults (MACHDEF.EDM)

MACHDEF is an abbreviation for Machine Defaults. The `MACHDEF.EDM` file carries the Machine default parameters. You can create the `MACHDEF.EDM` file by adding the MACHDEF variables in the `[objects]` section of the `install.ini`. The following table lists the variables of `MACHDEF.EDM` along with their description and values.

MACHDEF Variables

Variable	Description	Value
STARTDIR	The directory under machine IDMLIB	SYSTEM, \$MACHINE, or \$USER
UID	The value to use for entitlement	\$MACHINE or \$USER
IP	The IP address of the Configuration Server	A valid IP address or IP name
COP	Client Operation Profile resolution enabled	Y or N
ASK	Prompt user	Y or N

When the customer environment has Software Manager with differing user entitlements and no machine entitlement, they should specify the values of MACHDEF variables as follows:

STARTDIR	\$USER
UID	\$USER

Similarly, when you want to entitle applications to machine and user and machine has all the user applications entitled, you should specify the values of MACHDEF variables as follows:

STARTDIR	SYSTEM
UID	\$MACHINE

Suppose a customer always runs a Priming Connect for the machine side after the product is installed, where the machine has all the entitlements. In this scenario, you will not need a `MACHDEF.EDM` value to get all the default parameters.

User Defaults (USERDEF.EDM)

USERDEF is an abbreviation for User Defaults. The `USERDEF.EDM` file carries the User default parameters. You can create the `USERDEF.EDM` file by adding the USERDEF variables in the `[objects]` section of the `install.ini`. The following table lists the variables of `USERDEF.EDM` along with their description and values.

USERDEF Variables

Variable	Description	Value
UID	The value to use for entitlement	<code>\$MACHINE</code> or <code>\$USER</code>
IP	The IP address of the Configuration Server	A valid IP address or IP name
COP	Client Operation Profile resolution enabled	Y or N
ASK	Prompt user	Y or N

When your environment has Software Manager with differing user entitlements, you should specify the values of USERDEF variables as follows:

UID	<code>\$USER</code>

Similarly, when you want to entitle applications to machine and user and machine has all the user applications entitled, you should specify the values of USERDEF variables as follows:

UID	<code>\$MACHINE</code>

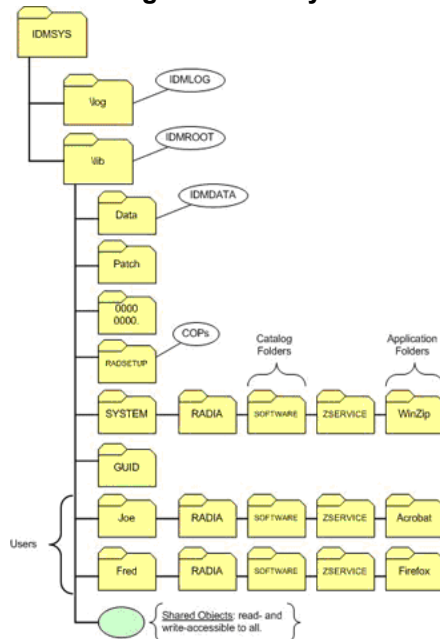
Directory Structure

This section describes the directory structure in Lockdown Mode and Non-Lockdown Mode.

Directory Structure in Non-Lockdown Mode

The following figure illustrates the directory structure in a non-Lockdown Mode. It is clear that SYSTEM directories are not isolated from USER directories. If the Machine-mode Lockdown is not enabled, they may be accessible to any user of the machine. Also, various USER directories would be susceptible to unauthorized access, as well as unauthorized modifications.

Sentient Agent Directory Structure in Non-Lockdown Mode



Directory Structure in Lockdown Mode

For administrators who are enabling Sentient Agent Lockdown mode there are choices to be made on where to isolate and store USER data. In Windows, the operating system defines “home” directories for each user and inherently prevents unauthorized access to these directories.

There are several locations that could be used to separate user data, but a typical implementation would have the Sentient administrator configure the user data store to use the home-directory approach, resulting in an existing managed device’s data stores being solely owned and accessible by the SYSTEM. This approach facilitates the migration of an existing environment to Lockdown enabled environment.

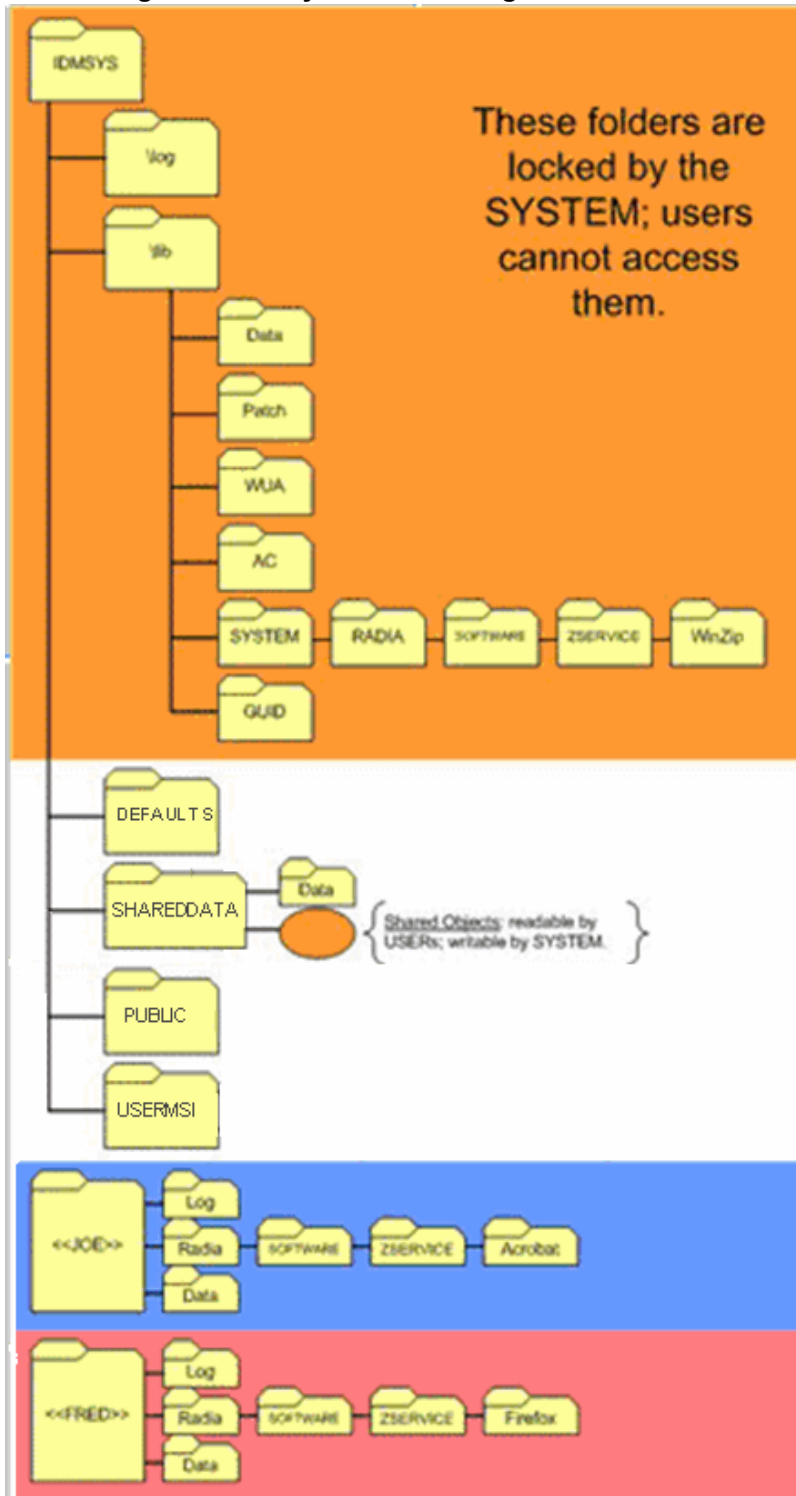
If, in your environment, the applicable directory is hidden, use the operating system-specific “view hidden folders” procedure to access and view it.

On the supported Windows operating systems, the home directories are:

- Windows XP and Windows 2003 Server:
C:\Documents and Settings\\Local Settings\Application Data\RCA\Agent
- Windows Vista®, Windows 7 and Windows 2008 Server:
C:\Users\\Appdata\Local\RCA\Agent

The following figure illustrates the Sentient agent directory structure in Lockdown Mode. The SYSTEM-based directories are no longer on the same “branch” as the USER directories. Therefore, can be secured and are no longer be accessible to any user of the machine — they are accessible to authorized administrators only. Also, each user has its own (USER-specific) directory, which again, is not on the same “branch” as another user. Hence, it is not susceptible to unauthorized access and modifications.

Sentient Agent Directory Structure in Agent Lockdown Mode



The location of the USER folders (JOE, FRED, and others) is determined at run time. It is based on the user that is logged in and by the value in `nvd.ini` file of `IDMUSR=CSIDL_LOCAL_APPDATA\RCA\Agent`. For example, for user JOE, this location would be:

- On Windows XP and Windows 2003 Server:
`CSIDL_LOCAL_APPDATA` is replaced with `C:\Documents and Settings\Joe\Local Settings\Application Data\RCA Agent`.
- On Windows Vista, Windows 7, and Windows 2008 Server:
`C:\Users\Joe\AppData\Local\RCA\Agent`

Features

- Sentient agent runs using the existing Sentient infrastructure. There are no changes required for the Sentient infrastructure with the exception of policy entitlements.
- USER directories are moved from the `IDMROOT` directory to `CSIDL_LOCAL_APPDATA\RCA\Agent`.
- SYSTEM privately owns all of `IDMROOT`. No read or write access by USERS.
- USERS get private `Log` and `Data` directories specific to each user.
- USERS can add a custom directory with their own script to access permissions. USERS can use the sample script, `setaccls.bat`, to create a customized script. The customized script can be used to set access permissions after installation of the Sentient agent.
- SYSTEM's `Log` and `Data` directories cannot be accessed by USERS.
- `IDMSYS\DEFAULTS` stores the priming objects that are needed to run an Sentient agent.
- `ZSYSACCT=Y` applications will be owned by SYSTEM; USERS cannot access the objects for the specified application.
- USERS have access to execute some files from `IDMSYS`, however SYSTEM methods such as `daemons`, `upgrdmaint`, and `radtimeq` will be executed by SYSTEM only.
- USERS cannot create `TIMER` instances.
- SYSTEM side implementations like Patch Manager and OS Manager, that are entirely owned by the SYSTEM, will keep working without any changes.
- USER MSI files will be stored in a separate directory.
- SYSTEM MSI files will not be accessible to USER.

Limitations

Following are the known limitations of Lockdown Mode.

Note: Some of these are intentional limitations to ensure the security aspect of the Sentient agent.

- Agent Lockdown Mode does not support FAT16 or FAT32 file systems because these do not support ACLs.
- USER connects cannot create `TIMER` instances. This is achieved by locking out `ZTIMEQ` as well as removing execute permissions to `radtimeq.exe`.
- Maintenance is entitled to SYSTEM only; USERS cannot initiate Sentient agent maintenance.

Retrieving Sentient Agent Information using Sentient WMI

The Sentient WMI (Windows Management Instrumentation) Provider provides all the necessary methods to create, delete, read, and update Sentient objects as well as query Sentient path settings and the number of instances in an object. Through WMI's use of DCOM (Distributed-COM), a program can easily access objects on a remote machine as well as a local machine. The primary goal of the Sentient WMI Provider is to allow administrators to write scripts in any language to access Sentient objects. For example, a person who is comfortable using VB Script can continue writing scripts to perform methods in VB Script rather than having to learn REXX. The WMI provider interface is available to any language that can create WMI objects.

VBScript Example - How to retrieve Sentient Agent root path using Sentient WMI provider

```
set WshShell = CreateObject("Wscript.Shell")
set wshenv = WshShell.Environment("Process")
set process = GetObject("winmgmts://./root/novadigm:NVD_Agent")
set method = process.Methods_ ("GetLocation")
set inParameters = method.inParameters.SpawnInstance_()
inParameters.Properties_.Item("Location").Value = "System"
set OutParameters = process.ExecMethod_ ("GetLocation", inParameters)
valueLocation = outparameters.Path
WScript.StdOut.Write "RCA Agent root path is " & valueLocation.
```

Note: In the lockdown mode, a user will be provided with the appropriate folder paths only - based on the context of the calling user.

Sentient WMI Provider Functions

Function	Description
Sint32 AddInstance Add an instance to an object.	[IN] string Path, [OUT] sint32 Index, [OUT] sint32 InstanceCount, [OUT] string ReturnMessage
Sint32 DeleteInstance Remove specified instance from an object.	[IN] string Path, [IN] stint32 Index (Default= -1) [OUT] sint32 InstanceCount, [OUT] string ReturnMessage
Sint32 DeleteProperty	Delete a property from an object [IN] string Path [IN] string Property [OUT] string

Function	Description
version 3.1 only The length of the property cannot be more than 8 characters.	ReturnMessage
Sint32 DeleteProperties version 3.1 only	Delete an array of properties from an object [IN] string Path [IN] string Property[] [OUT] string ReturnMessage
Sint32 GetLocation Get path to specified location.	[IN] string Location, [OUT] string Path, [OUT] string ReturnMessage
Sint32 GetValue Get specified value from object	[IN] string Path (Default = ZMASTER), [IN] sint32 Index (Default = 0), [IN] string Property, [OUT] string Value, [OUT] string ReturnMessage
Sint32 GetValues Get all variables and values from object	[IN] string Path (Default = ZMASTER), [IN] sint32 Index (Default = 0), [OUT] string Property[], [OUT] string Value[], [OUT] string ReturnMessage
Sint32 NumberOfInstances Returns number of instances in object	[IN] string Path, [OUT] sint32 InstanceCount, [OUT] string ReturnMessage
Sint32 SetValue Sets value in object	[IN] string Path, [IN] sint32 Index (Default = 0), [IN] string Property, [IN] string Value, [OUT] string ReturnMessage
Sint32 SetValues Sets and array of properties	[IN] string Path, [IN] sint32 Index (Default = 0), [IN] string Property[], [IN] string Value[], [OUT] string ReturnMess

Result on a Windows 2003 server machine with 'joe' logged in:

Sentient Agent root path is C:\Documents and Settings\joe\Local
Settings\Application Data\RCA\Agent\

Results on the same Windows 2003 server machine when the script is run in system context:

Sentient Agent root path is C:\Program Files\PSL\RCA\Agent\Lib\

Messages and Codes

This section is aimed to increase an administrator's understanding of the messages that might be generated by the Sentient Enterprise agent connect process. It also offers some probable causes for errors, as well as remedial actions that can be taken.

This section contains the following information:

- Numbers and descriptions of the Sentient messages and codes that are produced during the connect process.
- Possible causes of an error, and the steps that you need to take to troubleshoot and correct a problem.

To connect Sentient agent and the Configuration Server, the following conditions must be met:

- The Configuration Server must be running.
- The Configuration Server Database must be configured for the user and for managing the user's software applications.
- On the user's computer there must be sufficient available resources for the programs that are associated with the connect process, and for the management of the subscriber's software applications.
- Hardware and communications links must be properly operating.

Even with these conditions met during the connect process, other conditions can exist or events (such as the inadvertent deletion of needed files) can arise that prevent a successful completion. When this happens, Sentient produces informational messages. These messages are enumerated and described in the sections that follow.

Message Types

There are two types of messages:

- **Note messages** provide information about a condition that allows the connect process to continue.
- **Error messages** describe a condition that prevents the connect process from proceeding to a successful completion.

Message Details

The connect process issues messages in dialog boxes. The appearance of the dialog boxes will vary slightly, depending on whether the message is a **Note** or an **Error**.

- **Note messages** have a button, **Details**, which you can click in order to view additional information for the condition that triggered the message.
- **Error messages** also have a **Details** button. When it is selected, the message box expands to display the message number or the error code, the severity code, and the logs that should be checked for additional information.

Message Logs

When a message is issued, its number and text are recorded in the appropriate log on the user's computer. The log files are located in the **log** subdirectory of the directory in which the Sentient agent was installed. The default directories for log files are listed below.

- For Windows: `C:\Program Files\PSL\RCA\Agent\log`
- For Linux and Macintosh: `opt/PSL/RCA/Agent/log`

The messages are written in the following three files:

- `RADPINIT.LOG`
- `RADCONCT.LOG`
- `RADAPI.LOG`

If the cause of an error is not immediately apparent, note the steps that were taken immediately before the message appeared.

Caution: Do not do anything with Sentient until the log files are copied to a backup location. This will preserve information that might prove valuable in resolving the issue.

Message Categories

The messages that Sentient can produce during the connect process are organized into the following categories:

- API Errors
- Catalog Processing
- Client Processing
- External Data Download Codes
- Client Automation Internal Errors
- Invalid Data Errors
- Method Execution Errors
- SAP Errors
- Server Errors
- SSL Errors
- Transmission Errors
- User Exceptions
- User Interface Errors
- Verification Errors

These categories are high-level indicators of which part of the connect process is active when the message is produced.

Receiving a Message

All Sentient messages are numbered. The numbering is divided into groups, each of which is associated with a different phase in the connect process.

This section contains the Sentient messages in tables based on the connect-process phases.

Catalog Processing Messages

Catalog-processing messages can be produced as the Sentient agent attempts to create and populate the Service List, from which the user can select applications to install, verify, and uninstall. Except as noted in the following table "Catalog processing messages", the probable cause is corruption of the Service List either on the subscriber's desktop or in the Configuration Server Database.

Consult your Configuration Server Database administrator.

Catalog processing messages

Number	Text	Type	Additional Information
100	Error obtaining catalog from manager. Verify UserID and retry request.	Error	Select Refresh Catalog and verify the user ID.
101	Unique Identifier missing from this item in catalog.	Error	
102	Name missing from this item in catalog.	Error	An application that lacks a name is configured in the software catalog.
103	Description missing from this item in catalog.	Error	
104	Unable to create a directory to store catalogs from this Manager.	Error	Possible problem with security rights.
105	Unable to create a directory for this catalog.	Error	Contact your system administrator for assistance.
106	Unable to locate directory where catalogs are stored for this Manager.	Error	
107	Unable to locate directory for this catalog.	Error	
108	Unable to locate catalog	Error	Contact your system administrator for assistance.
109	No services available, empty catalog	Note	Probable CSDB configuration problem. There are no software applications configured for this subscriber in the CSDB.
110	Force service to be displayed in catalog.	Error	
111	The requested application does not exist in the catalog.	Error	

Transmission Error Messages

Transmission errors indicate that there is a problem establishing or maintaining the communications link with the Configuration Server. The Configuration Server might not be running, or there could be a problem with its host. Contact the operator of the Configuration Server computer to determine if it is running.

Another potential cause is lack of connectivity to the Configuration Server computer. Verify that the Sentient agent has connectivity to it by running the PING program from an MS-DOS session. Type

PING <*ip address*>

on the command line, where *ip address* is the IP address of the Configuration Server host machine. If the connectivity is good, a line will be displayed in the MS-DOS session window indicating the transmission time for each successful packet that was exchanged.

Transmission error messages

Number	Text	Type	
200	Unable to perform request - connection to manager failed.	Error	Configuration Server not running or stopped running during the connect process.
201	Invalid record received from Manager.	Error	
202	Template not received from Manager.	Error	
203	Instance not received from Manager.	Error	
204	Data not received from Manager.	Error	
205	Requested object not found.	Error	
206	Error sending data to server.	Error	Contact your system administrator for assistance.
207	Error receiving data from server.	Error	
208	Received patch from Manager instead of full file.	Error	
209	Error transferring physical application files from server.	Error	Sentient agent failed to get files. Contact your system administrator for assistance.
210	Expected files not found on server.	Error	

Number	Text	Type	
211	Sentient could not connect through the proxy server.	Error	

External Data-Download Return Codes (Multicast and Staging Requestor)

External data-download return codes

Number	Text	Type
240	No error, all data files successfully downloaded from the requested source.	Note
241	Not all data files could be downloaded from the requested source.	Error
242	Not all data files found at the requested source, do not try alternate sources.	Error
243	The data files will be downloaded at a later time using multicast.	Note
244	The Internet Explorer is set to work offline.	Error

User Exception Messages

User exception messages

Number	Text	Type	Additional Information
319	Sign-on failed - invalid user ID-password combination.	Error	Verify user ID and password. Click Refresh to retry.
320	Sign-on panel cancelled.	Note	User clicked Cancel in the Logon dialog box.
321	Cancelled at user's request.	Note	Cancels the entire process.
322	Versioning not available.	Note	Versioning is a feature of Sentient Application Manager; it is not installed on the computer.
323	User could not be authenticated on the proxy server.	Error	
324	Cancelled service at user's request.	Note	Cancels only this service, but will continue the installation of additional services.
325	Bootstrap method has decided to abort the connect process.	Error	

Number	Text	Type	Additional Information
326	The user opted to defer the current request.	Note	

Invalid Data Errors

Invalid data errors indicate probable corruption in the Configuration Server Database.

Contact your CSDB administrator and preserve the log files for further analysis.

Invalid data error messages

Number	Text
450	Missing ZRSCSIZE variable.
451	Invalid object id.
452	Missing ZOBJID variable.
453	Missing ZOBJDATE variable.
454	Missing ZOBJTIME variable.
455	Missing ZOBJCRC variable.
456	Error fetching path information.
457	Missing ZRSCCFIL variable.
458	Variable is empty.
459	Local object instance name missing.
460	'More Info' URL missing from service.
461	Catalog is either incompatible or corrupted.
462	Missing variable.
463	Invalid ownership specified in database.
464	The application package contains path that cannot be resolved.
465	Missing COMPxxx for product.
466	Attempted invalid operation on service group.

Sentient Agent Processing Error Messages

Sentient agent processing errors are usually caused by a malfunction, incorrect configuration, or misuse of the user's desktop. Possibilities include:

- Incomplete Sentient agent self-maintenance that has resulted in Sentient modules of different release levels attempting to interact on the desktop;
- Lack of necessary system resources or available memory on the desktop;
- Conflicting concurrent processes, such as deleting files or running a disk re-organization program, on the desktop.

Sentient agent processing error messages

Number	Text	Type	Additional Information
500	Error opening file.	Error	
501	Error writing data to file.	Error	
502	This application could not be repaired locally; connect to the Manager to fix it.	Error	
503	Error reopening file, for checkpoint restart.	Error	
505	Error decompressing data.	Error	
506	Insufficient disk space to install application.	Error	Free up additional disk space and re-try the connect.
507	Error creating directory.	Error	
508	Incomplete file path.	Error	The location for a file to be deployed on the user's desktop is incorrectly configured in the CSDB, or was incorrectly specified by the user in the Set Installation Directory dialog box.
509	Error getting drive statistics.	Error	
510	Missing file from temp directory.	Error	A file that was downloaded from the Configuration Server was deleted before it could be deployed to the proper location on the user's desktop.
511	Error deleting file.	Error	

Number	Text	Type	Additional Information
512	Error deleting directory.	Error	
513	Error applying patch.	Error	
514	Error setting file date/time.	Error	
515	Error setting file attributes.	Error	
516	Error in versioning.	Error	Error activating a version either because a method failed or there is insufficient disk space.
517	Agent not authorized to change versions.	Note	See the CONTROL variable in the VGROUP class, which indicates whether the Configuration Server or the Sentient agent controls version activation. The user is attempting to activate a version, which is under control of the Configuration Server, or the reverse.
518	Error deactivating version.	Error	The active version could not be deactivated because one of the methods to delete a file or link is missing.
519	Error setting file ownership.	Error	
520	Application cannot be installed because drive/file system cannot be accessed.	Error	

User Interface Errors

User interface error messages

Number	Text
550	Invalid message header received.
551	Error receiving response to a dialog from UI.
552	Error in XML Parser.
553	Error in building UI message.
554	Error sending message to UI.

Number	Text
555	Error receiving message from UI.
556	Error connecting to UI Server.
557	Error connect process still has not finished.
558	Error connecting to remote notify daemon.
559	Error sending data to remote notify daemon.
560	Error executing program from RCA UI Agent.
561	Entity not supported.
562	Entity not opened.
563	Error creating UI socket.
564	End of entity, no more data.
565	Error reading entity.
566	Error writing entity.

Verification Error Messages

Verification errors occur when the user attempts to verify files on the user's desktop, according to the verification settings configured for those files in the Configuration Server Database. These errors can indicate damage to the file on the subscriber's desktop or incorrect configuration of the verification option settings for the file in the CSDB.

You can correct the former by having Sentient repair the software application. To correct the latter, consult your CSDB administrator.

Verification error messages

Number	Text	Type
600	Verification of resources failed for one or more Applications.	Error
601	Resource CRC does not match.	Error
602	Resource size does not match.	Error
603	Desktop file is newer.	Error
604	Desktop file is older.	Error
605	Missing file.	Error
606	Missing directory.	Error
607	Environment has changed. Application needs to be updated.	Note

Number	Text	Type
608	Environment has changed and some resources failed verification. Application needs to be updated.	Error
609	Some of the data files are missing from temporary storage.	Error
610	Resource internal version does not match.	Error
611	File is maintained by the Windows File Protection system.	Error
612	Desktop file has wrong UID.	Error
613	Desktop file has wrong GID.	Error
614	Desktop file has wrong permissions.	Error
615	Missing link.	Error
616	File is maintained by the Desktop DNA.	Note

Server Errors

Server error messages

Number	Text
650	Server stopped application configuration.
651	Server does not contain a license for Agent's operating system.
652	Server is at task limit and will not accept connections.
653	Server is down for maintenance, updating database.
654	Server has disabled Agent connections.

Method Execution Errors

Methods are programs that run on the subscriber's desktop as part of the connect process. The methods can be supplied by Sentient or by the subscriber. Method-execution errors indicate that a problem occurred while running one of these methods. Typical problems include misspelling the method name, resulting in a "File not found" condition, coding invalid parameters on the method's command line, and lack of needed system resources or memory to run the method. Usually, the cause is incorrect configuration of an instance in the Configuration Server Database.

Contact the CSDB administrator for assistance and preserve the logs for further analysis.

Method execution error messages

Number	Text
701	Lock method failed.

Number	Text
702	Init method failed.
703	Create method failed.
704	Delete method failed.
705	Update method failed.
706	Unable to install downloaded resources. The programs/methods that are used to install the resources could not be found.
707	Internal error in method.
708	Unable to install downloaded resources. The programs/methods that are used to install the resources could not be executed.
709	Installation of a component failed. The program/method used to install the component returned an error.

SSL Errors

SSL error messages

Number	Text
750	Unable to get CA certificate from server.
751	Unable to get certificate from server.
752	Not yet valid CA certificate.
753	Not yet valid certificate.
754	Expired CA certificate.
755	Expired certificate.
756	Missing certificate revocation list.
757	Error decrypting CA certificate.
758	Error decrypting CA certificate
759	Error decrypting CRL.
760	Error decoding CA public key.
761	Error decoding public key.
762	Error in CA certificate signature.
763	Error in certificate signature.
764	Error in certificate revocation list signature.

Number	Text
765	Expired CRL.
766	Time stamp error in CRL.
767	Self-signed certificate found at depth 0.
768	Self-signed certificate found in certificate chain.
769	Missing local CA certificate.
770	Missing local certificate.
771	Error verifying leaf certificate signature.
772	Verify chain too long.
773	CA certificate has been revoked.
774	Certificate has been revoked.
775	Invalid or corrupt local CA certificate.
776	Invalid or corrupt local certificate.
777	Expired CA Certificate
778	Expired Certificate
779	Internal error in Secure Sockets Layer interface.
780	Missing Private Key file.
781	Password not supplied for private key
782	Bad password supplied for private key
783	Invalid private key.
784	Certificate required to connect to server.

Sentient Internal Error Messages

Internal errors occur when Sentient encounters an unexpected condition. Possible causes include corruption of Sentient desktop objects, incompletely applied Sentient self-maintenance, and conflicting processes running concurrently on the desktop. Sentient desktop objects can be inspected and modified, if necessary, using the Sentient Administrator Agent Explorer. For more information on Sentient Administrator Agent Explorer, see *Sentient Enterprise Console Online Help*.

Preserve the logs for further analysis and contact technical support.

Sentient internal error messages

Number	Text	Type	Additional Information
800	Internal error	Error	
801	Error allocating RAM.	Error	
802	Null object pointer.	Error	
803	Invalid arguments to function.	Error	
804	Error getting template.	Error	
805	Error getting instance.	Error	
806	Insufficient buffer size.	Error	
807	Empty object.	Error	
808	Invalid heap number	Error	
809	Invalid password.	Error	
810	Reboot required to complete request.	Note	
811	Machine is being rebooted to complete request.	Note	The user confirmed the restart request to complete the installation.
812	No configuration changes required.	Note	
813	Invalid template.	Error	
814	Failed to start graphical progress indicator	Note	
815	Error occurred trying to save local information	Error	
816	Error adding heap to local object.	Error	
817	Unable to process request - could not read local information.	Error	
818	Null pointer.	Error	
819	Unable to find folder containing local objects (IDMLIB). Check settings.	Error	
820	Unable to find folder containing executable files (IDMSYS). Check settings.	Error	

Number	Text	Type	Additional Information
821	Buffer too small.	Error	
822	Serialization error occurred. Process aborted?	Error	
823	Serialization error occurred. Process aborted?	Error	
824	Invalid operation.	Note	
825	Unable to create directory for Service information. Installation cannot continue.	Error	
826	Invalid mode	Error	
827	Version mismatch between RADAPI.DLL & RADIA.DLL. Incompatible RCA subscriber components found on desktop.	Error	Probable incomplete Sentient agent self-maintenance. Install Sentient agent again from installation media and retry to connect to the Configuration Server.
828	Invalid Request.	Error	
829	Error copying self- maintenance.	Note	
831	Error creating directory for this user.	Error	
832	Unable to determine User ID.	Error	
833	Error uninstalling service	Error	
834	Error sending previous error information to Administrator.	Error	
835	Error occurred during configuration determination; configuration information may be missing or inaccurate.	Error	
836	Failed to select service version	Error	
837	Failed to activate service	Error	
838	Client failed to execute program, after application installation.	Error	
839	Unsupported operating system	Error	

Number	Text	Type	Additional Information
840	Dialog REXX method failed	Error	
841	Missing reference object.	Error	
842	Error loading reference object.	Error	
843	Error opening RCA object.	Error	
844	Missing instance.	Error	
845	Unregistered RCA agent.	Note	
846	Missing version group instance.	Error	A non-existent version group instance was specified to the version-activation function.
847	Missing version instance.	Error	The specification of the version to activate within a version group is invalid. Likely, the INITIAL field in the VGROUP instance indicates a non-existent VERSION instance.
848	Error copying file	Error	
849	Error getting a variable value from instance	Error	
850	Error setting a variable value	Error	
851	Error saving an object. Check disk space.	Error	
852	Instance is corrupted, may be missing key variables.	Error	
853	Remote Execution Failure	Error	Contact your system administrator for assistance.
854	Failed to create a backup of a file that was to be updated.	Error	
855	Invalid Variable Name.	Error	
856	Invalid Variable Value.	Error	
857	Application wasn't installed completely.	Error	
858	Application wasn't installed completely. Reboot needed.	Error	
859	Installation is not complete (phased install process).	Error	

Number	Text	Type	Additional Information
860	Current process has timed out	Error	
861	Server rejected Agent identity (Invalid license).	Error	
880	Client is ready to apply maintenance immediately.	Error	
881	RCA_ERROR_INSTALL_INPROGRESS_AND_REBOOT	Error	Restart pending for the application's installation

API Errors

API error messages

Number	Text
901	Missing the registry settings.
902	Update engine has already been installed.
903	SDK settings could not be initialized.
904	Incompatible version of RCA Application Management SDK.

SAP Error Messages

SAP error messages

Number	Text	Type
000	Request successfully completed.	Note
920	SAP is not accessible.	Error
940	System is shutting down.	Error
996	Application has been skipped as part of processing.	Error
997	Application should be ignored as part of processing.	Error

NVDAgent Provider Errors

NVDAgent Provider error messages

Number	Text
0	OK
1	Invalid object Index

Number	Text
2	Object not found
3	Invalid object name
4	Error allocating array memory
5	Error unable to determine path
6	Object access denied
7	Unknown error occurred
8	Property name is too long
9	Property name not found
10	Property value not found

RADSKMAN Command Line Parameters

Regardless of which deployment method you choose, you will need to create and run a RADSKMAN command line. You can specify a RADSKMAN command line from a command prompt, a Scheduler (TIMER) instance, and a Notify command. Use RADSKMAN to:

- Check the status of all existing mandatory applications.
- Add new mandatory applications.
- Remove any mandatory applications that are no longer assigned to the subscriber.

Before using RADSKMAN in a production environment, test the command-line parameters. The RADSKMAN parameters can be divided into the following categories:

- ["Core Parameters" below](#)
- ["Machine or User Connect parameters" on page 480](#)
- ["Operations Parameters" on page 483](#)
- ["Process parameters" on page 484](#)

Core Parameters

Core parameters are used in most RADSKMAN commands. These parameters include the location of your Configuration Server, and how to identify the Sentient agent computer for policy.

RADSKMAN Core Parameters

Parameter	Explanation
cat	• Set <code>cat=prompt</code> to run self-maintenance, display the logon panel, and check the status of other services. • Set <code>cat=y</code> to check the status of services only.

Parameter	Explanation
	Set <code>cat=m</code> to use the local machine catalog for resolving the user's service list. This is used with <code>context=u</code>. Typically, this is also used with <code>local=y</code>. The default for Sentient Application Manager is prompt. The default for Sentient Application Self-Service Manager is dependent on the request type.
<code>dname</code>	The CSDB domain name for the services. This is the directory in which the service catalog (<code>ASERVICE.EDM</code>) is stored. For example, <code>dname=SOFTWARE</code> . The default for both Sentient agents is SOFTWARE . However, if <code>preload=y</code> , the default is RADSTAGE .
<code>IP</code>	The IP address of the Configuration Server. Note: If you do not specify the IP address, Sentient uses the IP address that is specified in the ZMASTER object stored in <code>IDMLIB</code> (by default, <code>C:\Program Files\PSL\RCA\Agent\LIB\</code>). The default for both Sentient agents is NOVARCS.
<code>IPVER</code>	IP version. Set <code>IPVER</code> to 4, 6, 46, or 64. <code>IPVER</code> overrides the <code>IPVER</code> value passed during the installation of the agent using the <code>install.ini</code> . If the command line argument is passed then the <code>radskman.exe</code> will validate and update the <code>IPVER</code> value of <code>ZMASTER.EDM</code> . If an invalid value (other than 4, 6, 46, or 64) is passed as the IP version, the <code>ZMASTER.IPVER</code> value is set to the default value, which is 46. The initial connect to the Sentient server happens based on the <code>IPVER</code> value in the <code>ZMASTER.EDM</code> . If the connect does the COP resolution, the subsequent connects happen based on the <code>IPVER</code> value set in <code>SAP.EDM</code> . <code>IPVER</code> is applicable for Windows systems only.
<code>mname</code>	The name of the Configuration Server. For example, <code>mname=RADSVR01</code> . The default for both Sentient agents is Sentient . However, if <code>preload=y</code> , the default is RADSTAGE .
<code>port</code>	The Configuration Server port. Note: If this is not specified, Sentient uses the port that is specified in the ZMASTER object stored in <code>IDMLIB</code> (by default, <code>C:\Program Files\PSL\RCA\Agent\LIB\</code>). The default for both Sentient agents is 3464.
<code>sname</code>	Specifies the name of the service that you want to process. If you do not specify a service, all mandatory services are processed.
<code>startdir</code>	Specifies the <code>IDMLIB</code> starting directory. Note: It is recommended to specify <code>startdir</code> on the command line if <code>uid</code> has been specified on the command line. If <code>startdir</code> is not specified, it will be set to the same value as <code>uid</code>. - Specify <code>startdir=\$MACHINE</code> to use the computer name. - Specify <code>startdir=\$USER</code> to use the currently logged on user. - Specify <code>startdir=value</code> to specify a custom starting directory. If value

Parameter	Explanation
	contains embedded spaces, enclose the entire name in quotation marks. During notify, you can also specify value to be retrieved from the EDM objects. For example, <code>startdir=&(ZMASTER.ZUSERID)</code> will retrieve value from ZMASTER. The defaults for both Sentient agents are: \$USER (if started in a user context, <code>context=u</code>); SYSTEM (if started in machine context, <code>context=m</code>). Note: Sentient Application Self-Service Manager does not pass a context by default.
uid	The identification that is used to identify the current session. Note: It is recommended to specify <code>startdir</code> on the command line if <code>uid</code> has been specified on the command line. If <code>startdir</code> is not specified, it will be set to the same value as <code>uid</code>. <code>uid=\$MACHINE</code> identifies the current session by the name of the computer. Note: It is not recommended to create agent <code>uid</code> with any of the following special characters: - backslash (\) - slash mark (/) - colon (:) - asterisk (*) - question mark (?) - quotation mark (") - less than sign (<) - greater than sign (>) - vertical bar () <code>uid=\$USER</code> identifies the current session by the name of the currently logged on user. <code>uid=custom</code> is used to identify the current session by a custom value that you specify. During notify, you can also specify value to be retrieved from the EDM objects. For example, <code>uid=&(ZMASTER.ZUSERID)</code> will retrieve value from ZMASTER. The defaults for both Sentient agents are: \$USER (if started in a user context, <code>context=u</code>); SYSTEM (if started in machine context, <code>context=m</code>). Note: If you do not specify a context for Application Self-Service Manager, the LOCALUID—as specified in the ZMASTER object stored in IDMLIB (by default, <code>C:\Program Files\PSL\RCA\Agent\Lib\</code>)—will be used.

Parameter	Explanation
	On Linux and Macintosh systems, prefix <code>\$MACHINE</code> and <code>\$USER</code> with <code>'\'</code> when you use them in <code>radskman</code> command line.

Machine or User Connect parameters

These parameters are beneficial when there are multiple users on the same Sentient agent computer, and with applications with machine and user components. These parameters control the frequency of connections to the Configuration Server, the display of the user logon panel, and when to send objects to the Configuration Server.

RADSKMAN Machine/User Parameters

Parameter	Explanation
cat	- Set <code>cat=prompt</code> to display the logon panel and check the status of other services. - Set <code>cat=y</code> to check the status of services only. - Set <code>cat=m</code> to use the local machine catalog for resolving the user's service list. This is used with <code>context=u</code>. Usually, this is also used with <code>local=y</code>. The default for Sentient Application Manager is prompt. The default for Sentient Application Self-Service Manager is dependent on the request type.
catexp	Use this parameter (in the format <code><attribute name>:<value></code>) to process applications based on a specific attribute in the ZSERVICE Class. Note: Specify multiple "or" conditions with a forward slash (/).
context	Set <code>context=m</code> when installing an application in the machine context; <code>context=u</code> when installing an application in the user context. - If <code>context=m</code> the following defaults are assumed. ■ <code>uid=\$machine</code> ■ <code>startdir=system</code> ■ <code>cat=prompt</code> ■ <code>ulogon=n</code>
	- If <code>context=u</code> the following defaults are assumed. ■ <code>uid=\$user</code> ■ <code>startdir=\$user</code> ■ <code>cat=prompt</code> ■ <code>ulogon=y</code> The default for Sentient Application Manager is: <code>u</code> if started with a user logged on; <code>m</code> if no user is logged on. There is no default for Sentient Application Self-

Parameter	Explanation
	Service Manager; all components are processed.
flushu	If <code>local=y</code> (see "local " below): - Specify <code>flushu=y</code> on user connects in order to send reporting objects to the Configuration Server at the end of the local connect for immediate feedback. This is the default behavior on user connects. - Specify <code>flushu=n</code> to prevent the reporting objects being sent to the Configuration Server. Be aware that the user's objects will continue to grow until they are sent to the Configuration Server. Note: <code>flushu=n</code> is applicable only for user connects and cannot be used for machine connect. On a machine connect, set <code>flushu=a</code> in order to send all reporting objects to the Configuration Server. The default for both Sentient agents is <code>Y</code>.
local	Specify <code>Y</code> to install resources, from the local Sentient agent computer, for the user's services. Use this only with <code>context=u</code> . Usually, this is used with <code>cat=m</code> .
machfreq	This parameter can be used to limit the number of Sentient machine connects on an Sentient agent computer. This is a global setting and is valid only if <code>context=m</code> . If <code>machfreq=0</code> , the Sentient agent connect will run on every restart of the Sentient agent computer. If <code>machfreq</code> is set to an integer (<i>n</i>), a machine connect will run only if it has been <i>n</i> number of hours since the last connect. For example, if <code>machfreq=1</code> , the machine connect will run after one hour of the last agent connect. However, if a connect runs the installation of a service that requires a restart of the computer, followed by a connect to complete the installation, the <code>machfreq</code> value is ignored and the connect is run. Once the connect is completed, the <code>machfreq</code> value is taken into consideration for all subsequent connects. For example, this reduces the OS Manager commits on a Windows Thin Client agent computer by limiting the number of connects.
mnt	Specify (<code>Y N</code>) whether to process Sentient agent self-maintenance on this connect. The default for both Sentient agents is <code>N</code> .
ulogon	This parameter display/hide the logon panel; it is valid only if <code>cat=prompt</code>. Note: If using the Sentient System Tray, specify <code>ulogon=n</code> to display the Sentient logon panel, which is not supported by the System Tray. The default for Sentient Application Manager is <code>N</code>. The default for Sentient Application Self-Service Manager is <code>Y</code>.
userfreq	Use this variable to prevent Sentient from running every time a user logs into the Sentient agent computer. It is valid only if <code>context=u</code> . If set to an integer (<i>n</i>) a user connect will run only if it has been <i>n</i> hours since the most recent user connect or if a machine connect has run.

Parameter	Explanation
	- If the value of <code>userfreq</code> is blank or not supplied, a user connect will run every time an Sentient agent connect is run with <code>context=u</code>. - If <code>userfreq=0</code>, a user connect will run only if the previous connect was a system connect. Agent uses the <code>USEREXEC</code> and <code>USERSID</code> variables in <code>RADUSERS.EDM</code> object to analyze whether the previous connect was a user or system connect and the <code>ISOTIME</code> variable to determine how much time elapsed since the last connect. System connect is run using Notify or Timer

Operations

These parameters influence how the Sentient agent will connect. Its features include computer restart handling, log specifications, and the display options for the user.

RADSKMAN Operations Parameters

Parameter	Explanation
ask	- Set <code>ask=y</code> to prompt the user before restarting the computer, which gives them a chance to save their work and close applications. - Set <code>ask=n</code> to restart the computer without prompting the subscriber. This is useful for unattended computers. The default for Sentient Application Manager is <code>y</code> if Sentient System Tray is running; <code>n</code> if System Tray is not running or there are no users logged on. The default for Sentient Application Self-Service Manager is <code>y</code>.
hreboot	- Specifying <code>hreboot=y</code> will allow RADSKMAN to handle a computer restart if it is required by the service. - Specify <code>hreboot=p</code> to power off the computer. The Sentient agent computer will shut down regardless of the service's reboot settings. - Specify <code>hreboot=n</code> to override any reboot settings defined for a service. If <code>hreboot=p</code> is used, the system assumes a quiet mode and the user does not receive an alert on the screen. If an alert is required, an alert type needs to be passed as follows: - Use <code>hreboot=py</code> to receive an Ok/Cancel alert on the screen. - Use <code>hreboot=pa</code> to receive an OK alert on the screen. - Use <code>hreboot=pq</code> or <code>hreboot=p</code> to suppress the alert. The default for Sentient Application Manager and Application Self-Service Manager is <code>y</code>.
ind	- Specify <code>ind=n</code> to hide the status indicator for each service. - Specify <code>ind=y</code> to show the status indicator for each service. The default for both Sentient agents is <code>y</code>.
jobid	Use <code>jobid</code> to further describe the source of this command line. It shows up in the <code>APPEVENT</code> , <code>IDENTITY</code> , <code>PREFACE</code> , and <code>SYNOPSIS</code> objects as <code>JOBID</code> . The

Parameter	Explanation
	defaults for both Sentient agents are: UserConnect (if started in a user context); MachineConnect (if started in machine context).
log	Create a name for the log that is stored in the <code>IDMLOG</code> directory.
logsize	Specify the maximum size (in bytes) of the log file. When the log size is reached, a backup file (<code>.bak</code>) is created (by default, <code>connect.bak</code>). If a backup file exists, it will be overwritten. The default for both Sentient agents is 1000000 .
rtimeout	Specify the number of seconds to wait—before rebooting the Sentient agent computer—if a reboot panel has been requested for a service. This timeout will allow a user to save and close applications before a reboot.
mcui	Set <code>mcui=y</code> to display the reboot dialog box. You must also select <code>ask=y</code> , <code>rtimeout=[>0]</code> , and <code>hreboot=y</code> to display reboot dialog box. By default, reboot dialog is not displayed.
checksig	Set <code>checksig=1</code> to enable Cyclical Redundancy Check (CRC) during install or update of any application on your device.
cmdpriority	When you run consecutive agent connects, only one connect is executed and remaining connect commands are saved into <code>ZTIMEQ.edm</code> . So you can set the execution priority of <code>ZTIMEQ</code> commands by using <code>cmdpriority=<positive integer value></code> as command parameter. Default value of command execution priority is 99. Lesser the value, higher is the priority of execution.

Operations Parameters

These parameters are used for specifying how to use Client Operations Profiles.

RADSKMAN Client Operations Profiles Parameters

Parameter	Explanation
cop	- Specify <code>Y</code> to enable Client Operations Profile resolution for this Sentient agent connect only. - Specify <code>N</code> to disable Client Operations Profiles resolution for this Sentient agent connect only. The default for both Sentient agents is <code>N</code> .
cdf	- Specify <code>Y</code> to enable the Connect Deferral feature. - Specify <code>N</code> to disable the Connect Deferral feature. For more information on the Connect Deferral feature, see <i>Sentient Enterprise Console Online Help</i> .
datauri	Add <code>datauri</code> (in the Universal Resource Identifier format) to the RADSKMAN command line to override the use of the SAP object for the DATA type.
product	If you used the <code>SAP.PRODUCT</code> attribute to identify that a SAP can be used with

Parameter	Explanation
	a specific product only, specify that product using this parameter. Multiple product filters must be separated by a comma.
rcsuri	Add <code>rcsuri</code> (in the URI format) to the RADSKMAN command line to override the use of the SAP object for the RCS type.

Process parameters

Process parameters involve service processing such as whether to repair or add applications on the current Sentient agent connect. These parameters also allow you to specify criteria for: service processing, sending application data to a Proxy Server, and handling SSL security for your Sentient agents.

RADSKMAN Process Parameters

Parameter	Explanation
Add	Specify (Y N) whether to install applications during this Sentient agent connect. The default for both Sentient agents is Y.
autofix	Specify (Y N) whether to automatically repair broken applications. The default for both Sentient agents is Y.
catexp	Use this parameter (in the format <i><attribute name>:<value></i>) to process applications based on a specific attribute in the ZSERVICE Class. Note: Specify multiple "or" conditions with a forward slash (/).
del	Specify (Y N) whether to delete applications during this Sentient agent connect. The default for both Sentient agents is Y.
merge	Specify an object name to have all variables in that object included in the ZMASTER object, and sent to the Configuration Server.
mnt	Specify (Y N) whether to process Sentient agent self-maintenance on this connect. The default for both Sentient agents is N.
preload	Used for Proxy Server preloading. Specify Y to use the IDMDATA directory that is specified in <code>NVD.INI</code> . Otherwise, specify the location of a directory into which the files will be copied.
process	Used to run a specific process based on the command line argument called "process". This is called as staccato processing. If the process argument contains any of the following variables then the application will enable the staccato processing. COMMIT : commits the changes from the differencing which involves invoking the methods and moving the files from cache to live location. DIFF : enables tree differencing, where the client differences the local object with the new objects coming from the Configuration Server. REPT : sends reporting objects to finish the connect process. VER : verifies the objects. Acts as a precursor to DIFF and COMMIT. If it is used before commit, it validates if all

Parameter	Explanation
	data has been downloaded before activating the state machine. XFER: downloads data for all the objects that require data from different sources into IDMDATA.
rep	Specify (Y N) whether to repair applications during this Sentient agent connect. The default for both Sentient agents is Y.
sendcat	Specify Y to send the service list, stored in the Sentient agent computer's ASERVICE object, to the Configuration Server at the end of the connect so that additional analysis can be done on it.
sslmgr	Specify the host name or IP address of the Configuration Server. Note: To perform Sentient agent self-maintenance over a secure channel (SSL), add the flag : : sm to the end of the SSL Manager IP address.
sslport	Specify the SSL communications port (normally 444).
upd	Specify (Y N) whether to update applications during this Sentient agent connect. The default for both Sentient agents is Y.
ver	Specify (Y N) whether to verify applications during this Sentient agent connect. The default for both Sentient agents is Y.

RADSKMAN Examples

The following examples are provided to illustrate common uses of RADSKMAN. On Linux and Macintosh systems, prefix \$MACHINE and \$USER with '\' to avoid their usage as variables.

- Perform a first catalog refresh that brings down the catalog (aservice.edm), runs self-maintenance, does not display the user logon panel if using Application Manager, and processes all mandatory applications:

```
radskman
ip=10.10.10.15,port=3464,mname=RCAAgent,dname=software,cat=prompt
```
- Perform a silent, full connect for user <machine name> with no user logon or progress indicator panels. This is a typical command used by a daily timer. Note that the value of ip= can be either a DNS name or an IP address:

```
radskman
ip=test.corp.com,port=3464,mname=RCAAgent,dname=software,cat=prompt,
uid=$machine,ulogon=n,ind=n
```
- Verify mandatory applications without updating the catalog, running self-maintenance, or repairing broken applications. Note that, before using cat=n, the machine must perform a first refresh catalog using cat=prompt at least once to bring down the catalog:

```
radskman
ip=10.10.10.15,port=3464,mname=RCAAgent,dname=software,cat=n,
autofix=n
```

- Install an application (WinZip) while updating only the catalog. Note that the Configuration Server uses a custom port number:

```
radskman
ip=10.10.10.15,port=5004,mname=RCAAgent,dname=software,cat=y,
sname=WinZip
```
- Install an application (WinZip) while updating only the catalog. Note that the Configuration Server uses IPv6:

```
radskman
ip=2001:db8:0:1:f8f3:a7bb:2bcb:6037,port=3464,mname=RCAAgent,
dname=software,cat=y,sname=WinZip
```
- Process all mandatory applications, handle reboot requests, and prompt the user with a panel to confirm the reboot request:

```
radskman
ip=10.10.10.15,port=3464,mname=RCAAgent,dname=software,cat=prompt,
hreboot=Y,ask=Y
```
- Ensure that filter handling in thin clients during connects is not handled by Sentient:

```
radskman
ip=10.10.10.15,port=3464,mname=RCAAgent,dname=software,HANDLE_FLTR=N
```
- Preload a stager using a location `d:\stager` and display the progress indicator panels. Note that if the Sentient System Tray feature is enabled, the progress indicator will be displayed in the System Tray information bubble. If the System Tray is disabled, the progress indicator will be displayed in a separate panel:

```
radskman ip=10.10.10.15,port=3464,uid=STAGER,preload=d:\stager,ind=Y
```
- Perform a machine connect:

```
radskman context=m
```

 - Note that because `context=m` and no other parameters were passed, the following default values are assumed:

```
ip=NOVARCS, port=3464, uid=$machine, startdir=system, cat=prompt,
ulogon=n, mname=RCAAgent, dname=software
```
- Perform a user connect:

```
radskman context=u
```

 - Note that because `context=u` and no other parameters were passed, the following default values are assumed:

```
ip=NOVARCS, port=3464, uid=$user, startdir=$user, cat=prompt,
ulogon=y, mname=RCAAgent, dname=software
```
- Perform the notify connect using IP version:

```
radntfyc <agent_host> -v <IP version> radskman ip=<configuration_
server_IP address>
```

In this instance,

 - `agent_host` is the IP address or hostname of the agent computer. Note that if the agent is in dual stack, the communication occurs based on the IP address you specify, whether it is an IPv4 or IPv6 host. This IP version overrides the value you specify for the `-v` switch. For example, if you specify the address as 192.168.xx.xx, and set the value for IP version as 6, the notify communication will not work. However, if you specify 64 as IP version, the communication will failover to IPv4.

- `-v` specifies the ip version for notifying the Sentient agent. IP version can be 4, 6, 46, or 64.

- `configuration_server_IP` is the IP address of the Sentient Configuration server.

Example 1: `radntfyc <192.168.255.255> -v 64 radskman ip=NOVARCS`

In this example, although you have specified the IP version over which the communication should happen as 6, the communication happens over IPv4, because the agent IP address that you have specified is in IPv4 format.

Example 2: `radntfyc <foo.bar.doo.com> -v 64 radskman ip=NOVARCS`

In this example, the communication happens over IPv6, since the agent address is provided in host name format, and the IP version as 6.

- Perform local and remote notify connect with the password encrypted:

```
radntfyc <agent_host> [-l username -p password -s port -e ENCRYPT]
<command>
```

In this instance,

- `agent_host` specifies the IP address or the hostname of the agent computer
- `-l` specifies the username
- `-p` specifies the password
- `-s` specifies the port number
- `-e` enables the password encryption
- `command` specifies the radskman command

You must specify all optional parameters (username, password or port) when you perform a notify. Missing any optional parameter may result in unexpected behavior of the command. If you do not specify optional parameters, these values are retrieved from the ZMASTER object.

Example: `RADNTFYC 1.2.3.4 -l USER1 -p XYZ -s 3465 -e ENCRYPT`

```
'C:\PROGRA~1\Novadigm\radpinit req="Apply
```

```
Updates",mname=ABC,dname=SOFTWARE,sname=AMORTIZE'
```

In this example, USER1 on the agent device with IP 1.2.3.4 is notified at port 3465 for the AMORTIZE application update and password is sent in the encrypted format.

- Perform a system connect:

System connect is used if elevated privileges are required during an Agent connect.

All connects launched via Notify or Timer are system connects.

On a local machine, the 'notify client' module `Radntfyc` can be used to invoke a System connect:

```
Radntfyc localhostradskman ip=10.10.10.15, port=3464, uid=$machine,
startdir=system, cat=prompt,ulogon=n, mname=RCAAgent,dname=software
```

- Perform a user connect only if: a machine connect has occurred *and* at least 12 hours have passed since the most recent user connect:

```
radskman context=u,userfreq=12
```

- Perform verification, differencing and stop the connect.

```
RADSKMAN PROCESS="VER,DIFF",ip=NOVARCS, port=3464
```

Use comma (,) as the delimiter between the directives and enclose the directives in the double quotes (" ").

- Validate that all the data has been downloaded, run the state machine, and send the reporting object to the configuration server.

```
RADSKMAN PROCESS="VER, COMMIT, REPT", ip=NOVARCS, port=3464
```

Use comma (,) as the delimiter between the directives and enclose the directives in the double quotes (" ").

- Download data.

```
RADSKMAN PROCESS="XFER", ip=NOVARCS, port=3464
```

- Perform a connect to httpstcp protocol

```
Radskman rcsuri=httpstcp://<full-service
```

```
satellite:443>, sname=<servicename>, uid=<uid>, log=connect.log, cop=y
```

Note: When you connect to the httpstcp protocol using the rcsuri parameter, the IP version is not read from the SAP.EDM object. Instead, it is read from the ZMASTER.EDM object. Therefore, the connect.log file stores the IP version mentioned in the ZMASTER.EDM object. To override the value of IPVER value in ZMASTER.EDM, you can run a connect using IPVER. For more information on IPVER, see parameter list in the section, "[Core Parameters](#)" on page 477.

- Using RADSKMAN with IPVER.

```
radskman ip=10.10.10.15, port=3464, IPVER=46
```

Local AIP Support for the MSI Redirector

The Sentient MSI Redirector is a specialized, local-host HTTP server that accepts and satisfies file requests that are made during an MSI installation. On receiving the HTTP request from MSI, the Redirector retrieves the file from its local cache (if it exists); if the file is not in its local cache, the Redirector requests the file from an upstream Configuration Server or Proxy Server. This process requires that the MSI installation supports HTTP, although some vendors (including Microsoft) have removed HTTP support from their product installations. Without HTTP support, the MSI installation will not be able to directly request the files from the MSI Redirector.

In order to continue to use the MSI Redirector, a level of redirection exists at the local file-system level. When using this method, MSI is told that the **Application Installation Point (AIP)** is local, and requests the files directly from the file system. This request is captured and forwarded to the MSI Redirector, which satisfies the request in its usual way. The file is then placed in the defined local AIP where MSI can process it. The local AIP is temporary; it is removed after the installation is completed.

Local AIP support for MSI redirector provides the following advantages:

- Increased level of flexibility and performance.
- Ability to apply date or time stamps and support long file names.
- Ability to run Custom Actions that do not support http as an Install Point.

To enable using the Local AIP:

Caution: It is recommended keeping your LOCALAIP as short as possible to accommodate AIPs that have deep directory structures.

1. Use the CSDB Editor to navigate to the MSI Resources (MSI) Class in the SOFTWARE Domain.
2. Each MSI application will have an MSI instance and an IDX instance.
3. Right-click the MSI instance and select **Edit Instance**.
4. Set **MSIDRVR (Use Local AIP [Y/N])** to **Y**.

Note: If MSIDRVR is not in your database, create it in the MSI Resources (MSI) Class as a 1-byte variable with a description of `Use Local AIP [Y/N]`. It is recommended to back up your database before making changes to a Class template. For information on editing Class templates, see *Sentient Enterprise Console Online Help*.

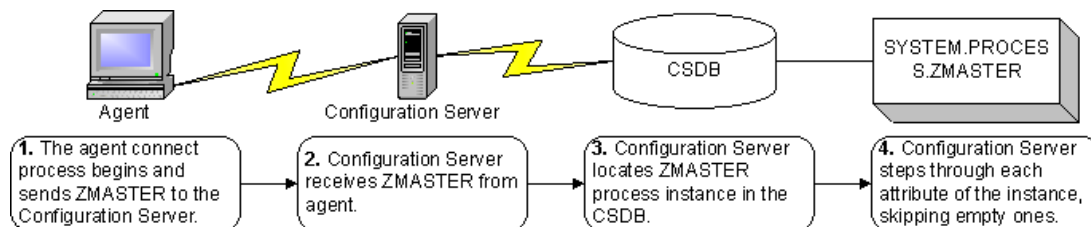
5. Click **OK**.
6. Click **Yes** to confirm the changes.
7. Configure the SETTINGS.LOCALAIP attribute in COPs to control the destination of the local AIP folder on the Sentient agent desktop. For example, `C:\localaip`.

Note: By default, the LOCALAIP attribute is not available in PRIMARY.CLIENT.SETTINGS class. However, you can edit the SETTINGS class to create the LOCALAIP attribute by using Sentient Administrator CSDB Editor. Make sure that the attribute type is set to **Variable**. For information about how to edit a class to create an attribute, see the *Sentient Enterprise Console Online help*.

Resolution Process

The Configuration Server uses the **resolution process** to accomplish a unit of work in response to a service request. The unit of work is defined by the contents of the CSDB and parameters included in the request. In other words, what the Client Automation infrastructure does depends on what information is stored in its CSDB and what information accompanies the request for Client Automation to perform some action. For example, the agent connect submits service requests to the Configuration Server, and the Configuration Server performs a resolution in response to each request.

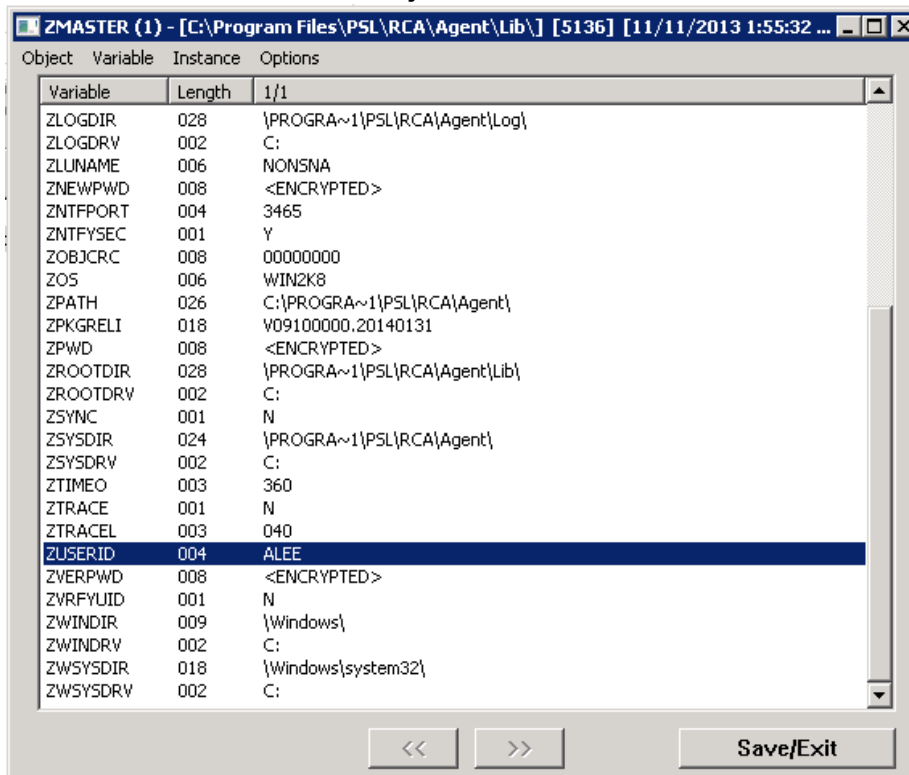
Configuration Server Performs Resolution



The ZMASTER object is sent to the Configuration Server during the agent connect. The ZMASTER object contains information about the agent computer that is needed to run Client Automation, such as the identity of the subscriber and the IP address of the agent computer.

The Configuration Server stores the ZMASTER object in **global memory**. Global memory is a temporary storage area in the Configuration Server. The Configuration Server maintains global memory's contents for the duration of the resolution process.

Subscriber's local ZMASTER object



The screenshot shows a window titled "ZMASTER (1) - [C:\Program Files\PSL\RCA\Agent\Lib\] [5136] [11/11/2013 1:55:32 ...]". It contains a table with columns: Object, Variable, Instance, and Options. The table lists various system variables and their values. The variable ZUSERID is highlighted in blue.

Object	Variable	Instance	Options
	Variable	Length	1/1
	ZLOGDIR	028	{PROGRA~1\PSL\RCA\Agent\Log\}
	ZLOGDRV	002	C:
	ZLUNAME	006	NONSNA
	ZNEWPWD	008	<ENCRYPTED>
	ZNTFPORT	004	3465
	ZNTFYSEC	001	Y
	ZOBJCRC	008	00000000
	ZOS	006	WIN2K8
	ZPATH	026	C:\PROGRA~1\PSL\RCA\Agent\
	ZPKGREL	018	V09100000.20140131
	ZPWD	008	<ENCRYPTED>
	ZROOTDIR	028	{PROGRA~1\PSL\RCA\Agent\Lib\}
	ZROOTDRV	002	C:
	ZSYNC	001	N
	ZSYSDIR	024	{PROGRA~1\PSL\RCA\Agent\}
	ZSYSDRV	002	C:
	ZTIMEO	003	360
	ZTRACE	001	N
	ZTRACEL	003	040
	ZUSERID	004	ALEE
	ZVERPWD	008	<ENCRYPTED>
	ZVRFYUID	001	N
	ZWINDIR	009	{Windows\}
	ZWINDRV	002	C:
	ZWSYSDIR	018	{Windows\system32\}
	ZWSYSDRV	002	C:

After storing ZMASTER in global memory, the Configuration Server finds the Process instance for the ZMASTER. This is the **process entry point**. Its location is SYSTEM.PROCESS.ZMASTER.

The Configuration Server reads each attribute of SYSTEM.PROCESS.ZMASTER. Based on an attribute's value, the Configuration Server may:

- Set variable values.
- Evaluate an expression.
- Execute a method.
- Connect to other instances

If there is a connection to another instance, the Configuration Server processes the connected instance. Then the resolution process resumes in the referring instance at the next attribute after the connection attribute. For example, in the following figure, the first connection instance links to SYSTEM.ZINTENT.&(SESSION.INTENT). After processing this connection instance, the resolution process will return to PRIMARY.SYSTEM.PROCESS.ZMASTER.

PRIMARY.SYSTEM.PROCESS.ZMASTER instance

Radia Processes class ZMASTER Instance Attributes:		
Name	Attribute Description	Value
 _ALWAYS_	Method	
 _ALWAYS_	Method	
 _ALWAYS_	Connect To	
 _ALWAYS_	Connect To	SYSTEM.ZINTENT.&(SESSION.INTENT)
 _ALWAYS_	Method	
 _ALWAYS_	Method	
 _ALWAYS_	Method	
 _ALWAYS_	Method	
 _ALWAYS_	Method	
 _ALWAYS_	Method	
 _ALWAYS_	Method	
 DESCRPT	Process Description	Processing Client Request for &ZCUROBJ
 ZMAXOKRC	Max acceptable method Return Code	008

During resolution, the Configuration Server performs symbolic substitution to set values and connect to other instances. For example, in SYSTEM.PROCESS.ZMASTER there is a connection to SYSTEM.ZINTENT.&(SESSION.INTENT). The Configuration Server substitutes the value of the INTENT from the SESSION object that is in global memory. If the INTENT is CATALOG_RESO, the resolution process will connect to SYSTEM.ZINTENT.CATALOG_RESO and resolve that instance. In the SYSTEM.ZINTENT.CATALOG_RESO instance, there is a connection to SYSTEM.ZPRODUCT.&(PREFACE.ZDOMNAME). The Configuration Server substitutes the value of the ZDOMNAME from the PREFACE object that is in global memory. If the ZDOMNAME is SOFTWARE, the resolution process will connect to SYSTEM.ZPRODUCT.SOFTWARE and resolve that instance. In SYSTEM.ZPRODUCT.SOFTWARE there is a connection to POLICY.USER.&(ZMASTER.ZUSERID). The Configuration Server substitutes the value of the ZUSERID from the ZMASTER object that is in global memory.

In the figure ["Subscriber's local ZMASTER object" on the previous page](#), the value of ZUSERID is alec. Therefore, the resolution process will connect to POLICY.USER.ALEE, and resolve that instance.

POLICY.USER.ALEE instance

Database Tree View:		Users class ALEE Instance Attributes:		
		Name	Attribute Description	Value
Database		UNAME	Name	
LICENSE		ZCONFIG	Collect Hardware Info [Y/N]	Y
PRIMARY		ZSETMSGA	Send Message to Audit Resource	DAILY
ADMIN		ZDLIMIT	Maximum Disk Space	0
AUDIT		USERID	Enterprise User Id	
CLIENT		ZTIMEO	Client Timeout (Seconds)	240
NOVADIGM		ZTRACEL	Trace Log Level [0-999]	040
PATCH		ZTRACE	Trace On or Off [Y/N]	N
POLICY		ZPRIORIT	Exec. Priority	000
Country / Region (COUNTRY)		ZSHOW	Display Status Indicator [Y/N]	N
Departments (DEPT)		_ALWAYS_	Utility Method	
Machine Manufacturer (MANUFACT)		ic_ALWAYS_	Member of	SOFTWARE.ZSERVICE.AMORTIZE
Machine Models (MODEL)		ic_ALWAYS_	Member of	
Machine Roles (ROLE)		ic_ALWAYS_	Member of	
Machine Subnets (SUBNET)		ic_ALWAYS_	Member of	
Mobile Device Config (MBLCONF)		ic_ALWAYS_	Member of	
Multicast (MULTICAST)		ic_ALWAYS_	Member of	
PDACONFG (PDACONFG)		ic_ALWAYS_	Member of	
Server Stagers (STAGER)		ic_ALWAYS_	Member of	
Users (USER)		ic_ALWAYS_	Member of	
Workgroups (WORKGRP)		ic_ALWAYS_	Member of	
PRDMAINT		ic_ALWAYS_	Member of	PRDMAINT.ZSERVICE.MAINT_40
SOFTWARE		NAME	Friendly name	ALEE
SYSTEM		ZVERDT	Verify Desktop [Y/D/R/I]	Y
Application Manager (ZCOMMAND)		ZSELFIND	Self Maintenance Display [Y/N]	N
Consoles (ZCONSOLE)		ZSLFINTVL	Self Maintenance Interval (hours)	0
DB Version (DBVER)		ZYPESEL	Type Selection - Calc. Pack Sizes	Typical
Methods (ZMETHOD)		EMAIL	E-mail Address	
Radio Intent Class (ZINTENT)		ZOBJPFUE	Free Unused Pool Elements	Y
Radio Processes (PROCESS)		MSITRACE	activates MSI verbose Trace="vpath	
_BASE_INSTANCE_		ZGRPINFO	Gather Group membership info[Y/N]	N
_NULL_INSTANCE_		ZOBJPTCH	Perform Patching [Y/N]	N

In the figure "POLICY.USER.ALEE instance" above, after setting a number of variables, the first connection attribute is to SOFTWARE.ZSERVICE.AMORTIZE. In your implementation, the POLICY instance may connect to a workgroup that connects to a service.

ZSERVICE.AMORTIZE instance

Database Tree View:		Application class Amortize Instance Attributes:		
		Name	Attribute Description	Value
PRIMARY		ZSTOP000	Expression Resolution Method	WORDPOS(EDMGETV(ZMASTER.ZD
ADMIN		ZSTOP001	Expression Resolution Method - 001	
AUDIT		ZSTOP002	Expression Resolution Method - 002	
CLIENT		ZSTOP999	Stop Unless Radia Connect	
NOVADIGM		ZSVCNAME	Service Name/Description	Amortize
PATCH		ZSVCTTYP	Application Target Type [A/S]	
PATCHMGR		ZSVCMO	Mandatory or Optional Service [M/O]	0
POLICY		ZSVCCSTA	Service Status on Client [999]	999
PRDMAINT		ZSVCPRI	Service Create Ordering [01-99]	
SOFTWARE		ic_ALWAYS_	Contains	SOFTWARE.PACKAGE.AMORTIZE2_
Alert / Defer (ALERTDEF)		ic_ALWAYS_	Contains	SOFTWARE.PACKAGE.AMORTIZE2_
Application (ZSERVICE)		ic_ALWAYS_	Contains	
_BASE_INSTANCE_		ic_ALWAYS_	Contains	
Amortize		ic_ALWAYS_	Contains	
Drag & View		ic_ALWAYS_	Contains	
GS-CALC		ic_ALWAYS_	Contains	
Redbox Organizer		ic_ALWAYS_	Contains	
Remote Control		ic_ALWAYS_	Contains	

A Service instance links to packages. The figure "ZSERVICE.AMORTIZE instance" above begins with a ZSTOP expression variable. An expression variable contains statements that, if evaluated to "true," stops the resolution of the current instance. An expression allows alternative paths to be

taken during resolution based on variable data. In this case, the expression checks to be sure that the operating system of the agent computer is authorized for the Amortize software.

If the agent computer's operating system is Windows 2003, Windows XP, or Windows Vista, the resolution process continues with this instance, connecting the file instances, registry entries, path instances, and shortcuts. If the agent has an operating system other than the ones previously mentioned, resolution returns to SOFTWARE.ZSERVICE.AMORTIZE, and to the next connection instance.

Eventually, the resolution process will return to the User instance, finish resolving it, and return to the process entry point, SYSTEM.ZPROCESS.ZMASTER.

After processing all attributes in the SYSTEM.PROCESS.ZMASTER instance, resolution terminates.

Chapter 10

Configuration Server

Configuration Server

The Configuration Server is a main component and essential function of Core and Satellite servers. In conjunction with the Configuration Server Database (CSDB), it configures and maintains the desired state for your enterprise devices and agent computers.

Configuration Server Management

The Configuration Server can:

- **Manage a Sentient agent's Desired State**
Dynamically create an Sentient agent's **desired state** based on situation-specific data, thereby creating a software environment that automatically adapts to user and device changes.
- **Distribute Configuration Server Database Objects**
Synchronize distributed objects (such as application components, packages, device configurations, and policy relationships) across the network, and automatically manage object transfer between Sentient Enterprise components.
- **Deliver and Maintain Sentient Policies**
Maintain enterprise policies in the CSDB. When a device that is being managed by Sentient connects to the Configuration Server all current policies are automatically applied.
- **Issue Connects and Fulfill Requests**
Contact devices causing them to initiate data requests and desired-state requests. These requests can be according to a schedule or on notification from an administrator.

Configuration Server Directories

The table below lists the directories that are automatically created during the Configuration Server installation.

Configuration Server Directories

Directory	Contents
Bin	The shell scripts that enable you to start, stop, clean up, and query the Configuration Server.
DB	The CSDB files.
Exe	The Configuration Server methods and executable files.
internet	The internet HTML and graphics files.
lib	The files for proper Configuration Server operation. Note: Do not modify or

	delete these files.
Log	The Configuration Server log.
Rexx	This directory is for storing customized REXX methods. Note: Its sub-folder, NOVADIGM, contains the default Configuration Server REXX methods. See the caution that follows.

Note: Altering the Configuration Server methods that are in the ConfigurationServer\rexx\NOVADIGM directory could adversely effect Sentient processing.

Therefore, it is recommended that a method be copied up one level, to the ConfigurationServer\rexx directory, before being modified.

Backing up Configuration Server

It is recommended that the Configuration Server (and CSDB) be periodically backed up. To facilitate doing so, a list of the Configuration Server directories (that are installed by default) and their applicable platforms and contents are provided in the table below.

Note: It is recommended that backups be done in accordance with each environment's in-house, corporate protocol.

Configuration Server Directories

Directory	Contents
bin	The Configuration Server binary files and the edmprof file.
DB	The Configuration Server Database files. The DB directory is created under <datadir>.
lib	This directory contains files for proper Configuration Server operation; do not modify or delete these files.
log	The Configuration Server log.
modules	The .tkd files for OS Manager.
rexx	This directory is for storing customized REXX methods. Note: Its subfolder, NOVADIGM, contains the default Configuration Server REXX methods.

Configuration Server as a Windows Service

During the installation, the Configuration Server was set up to run as a Windows Service. Confirm this by checking the Services area of the Windows machine. As a Windows Service, the Configuration Server can be started, stopped, and queried, as described in this section.

Windows Service Options

If the Configuration Server is set up to run as a Windows Service, it can be started, stopped, and queried, as described in this section.

To start the Configuration Server as a Windows Service:

1. On the Sentient Core server, click **Start>Control Panel>Administrative Tools>Services**.
2. Right-click the service Sentient Configuration Server and select **Start**.

The Configuration Server will start.

To stop the Configuration Server as a Windows Service:

1. On the Sentient Core server, click **Start>Control Panel>Administrative Tools>Services**.
2. Right-click the service Sentient Configuration Server and select **Stop**.

It will take approximately two minutes for the Configuration Server to shut down.

Using the Windows Event Viewer with the Configuration Server

If the Configuration Server is running as a Windows Service, the Windows Event Viewer can be used to see key Configuration Server messages, such as start-ups and shutdowns.

The Event Viewer displays three types of messages: error, warning, and informational. There are five Event Viewer messages (two informational and three errors) that work with Configuration Server processing. These are described in ["Configuration Server Messages in Event Viewer" on the next page](#).

Accessing the Event Viewer

The Event Viewer is invoked in a variety of ways on the various Windows operating systems. For instructions on how to access the Event Viewer on your operating system, consult the documentation that was distributed with it.

Filtering for Configuration Server Messages

After the Event Viewer is open, locating messages that are specific to the Configuration Server is easy.

To locate Configuration Server messages:

Note: The steps outlined below are applicable to most Windows operating systems, and are accurate as of this writing.

1. Select **Application**.
2. From the Menu bar, click **View** and select **Filter**.
3. From the Event Source drop-down list, select **ZTopTask**, and click **OK**.

Now, the right panel of the Event Viewer window displays only those messages that result from Configuration Server processing. Verify this by checking that **ZTopTask** is listed in all the rows of the Source column.

Configuration Server Messages in Event Viewer

Detailed below are the five Event Viewer messages that are related to the Configuration Server processing.

- **Start-up Message**
`Configuration Server HAS STARTED`
The Configuration Server for Windows is beginning its start-up processing by running the Configuration Server program.
- **Error Message 1**
`START: Configuration Server CONTROL DISPATCHER FAILED`
The Configuration Server for Windows failed during service initialization. Start-up processing has stopped.
- **Error Message 2**
`Configuration Server CONTROL HANDLER REGISTRATION FAILED IN SERVICE MAIN`
During second stage initialization, the Windows service controller failed to update the registry keys. Start-up processing has stopped.
- **Error Message 2**
`SET Configuration Server FAILED IN REPORT STATUS`
An attempt was made to report the status of the Configuration Server for Windows start-up process to the Configuration Server service. However, the Configuration Server service did not receive the status report.
The Configuration Server service may actually be running, or the start-up service has stopped. The Configuration Server is unable to discern what the actual condition is.
- **Shut-down Message**
`Configuration Server STOPPED BY STOP REQUEST OR SHUT-DOWN REQUEST`
The Configuration Server has shut down.

Tuning the Configuration Server

The performance of the Configuration Server depends on a number of factors, such as the number of Sentient agents being concurrently processed, the complexity of the configurations for those Sentient agents, the volume of the data being processed, and network bandwidth. The configuration of the Configuration Server log, which documents system status for informational purposes and problem determination, can also alter performance.

The Configuration Server operational parameters are contained in its `edmprof` file. The performance of the Configuration Server can be tuned by working with the settings of the `edmprof` file.

The Configuration Server `edmprof` file contains the parameters that determine how the Configuration Server will operate. This file is organized into **sections**—with each section containing keywords, called **settings**. The sections can be categorized into functional areas. Further detail for the settings in each of the sections is provided in this chapter.

- **Identification**

- MGR_STARTUP specifies the Configuration Server by ID, name, type, and communications port.
- MGR_DIRECTORIES identifies the directory paths for the Configuration Server Databases, REXX methods, and non-REXX methods.
- MGR_LICENSE contains your unique license string.

- **Specification**

Configuration Server settings establish application wide technical parameters.

- MGR_CACHE contains cache-processing options.
- MGR_TPINIT identifies communications packet sizes.

- **Initialization**

- MGR_ATTACH_LIST determines which Configuration Server programs are initiated at startup, and has options to define their functioning.
- MGR_CLASS specifies which classes and instances of the CSDB are cached during the Configuration Server initialization process.
- SECTION_DELIMITERS specifies which characters are used to distinguish sections in the `edmpprof` file.

- **Operations**

A number of sections in the Configuration Server `edmpprof` file contain parameters for system operations.

- MGR_ACCESS determines Configuration Server access to administrator and console functions.
- MGR_DB_VERIFY specifies the parameters for automatic CSDB verification at initialization.
- MGR_DIAGNOSTIC specifies the timing, size, and logging options for verifying adequate disk space for CSDB operations.
- MGR_DMA specifies parameters for Sentient Enterprise Distributed Configuration Server (Distributed Configuration Server).
- MGR_METHODS identifies options for method processing.
- MGR_NOTIFY specifies the Configuration Server defaults for Sentient agent notifications.
- MGR_OBJECT_RESOLUTION specifies parameters used in object resolution.
- MGR_POOLS (Windows only) allocates available pools of memory (in different sizes) for system tasks.
- MGR_RETRY values define how long a Sentient agent is to wait before attempting to reconnect to the Configuration Server.
- MGR_TASK_LIMIT identifies the maximum concurrent Configuration Server tasks, ongoing and deferred, system related and/or Sentient agent-connect related.
- MGR_TIMEOUT specifies the amount of time a Configuration Server will wait for an inactive Sentient agent.

- **Monitoring**
 - MGR_LOG and MGR_TRACE identify where the Configuration Server log is located, how flexible it is, and which individual system traces are to be captured in the log.
 - MGR_SMTP_MAIL specifies the parameters for using SMTP mail messages to support Configuration Server monitoring.
 - MGR_MESSAGE_CONTROL specifies where log messages are to be sent and if they are to be suppressed.
 - MGR_USERLOG enables an administrator to establish a user logging facility.

Viewing Configuration Server Settings

The `edmprof` file can be opened in a text editor so that its parameters can be viewed and, if necessary, adjusted. This section describes how to access the `edmprof` file, as well as important information about editing and saving the file.

The `edmprof` file can be found in varying locations, based on operating system, as described below:

- **Windows**

The `edmprof` file settings are located in the `edmprof.dat` file, located in the **bin** subdirectory of the Configuration Server directory.

Caution: While the settings in the `edmprof` file are readily accessible and easy to modify, some of the values are critical to the operation of the Configuration Server. Therefore, it is imperative that you *do not*:

- alter any `edmprof` file settings without consulting this guide first, and then using the recommended values.
- delete any `edmprof` file settings unless instructed to do so by a member of Persistent Systems Technical Support.

Editing Configuration Server Settings

The `edmprof` file can be edited in a standard text-editing application.

Caution: Be sure to review the important information in this section before editing this file. Failure to do so could adversely affect your Sentient environment.

It is recommended to back up the `edmprof` file before editing it.

- You must use a UTF8-aware text editor when editing the `edmprof` file.
- Be sure to select **UTF-8** as the *encoding type* when saving the file.
 - Windows users are advised to use **Notepad**.

Failure to use a recommended text-editing application and saving the file as recommended in this section could result in the file's changes not being correctly applied to the Configuration Server. This could adversely effect your Sentient environment.

Most of the sections of the `edmpprof` file can be independently configured, whereas some are based on operating system and communications requirements. While many of the sections are optional, a number are required for proper Configuration Server functioning.

Note: Since some of the sections in the `edmpprof` file are optional, and others are platform-specific, it's possible that not all of the settings will be visible in every `edmpprof` file.

The `edmpprof` file is created during the installation of the Configuration Server. Much of its information is derived directly from parameters that are specified during the installation; while others are automatically entered during the installation.

Two types of values are in the `edmpprof` file.

- An **as-installed** value represents a manual input, specified during installation or a derived entry for a required setting.
- A **default** value is established by the Configuration Server if there is a blank value for that setting, whether it is required or manually entered.

Note: If the value of a setting/parameter is documented as **N/A**, there is no value of that type for that setting/parameter.
If a value of a setting/parameter has **NONE** (all uppercase) specified, then this is an accepted, optional value for that setting/parameter.

EDMPROF File

The `edmpprof` file is organized into *sections*. Each section contains individual keywords called *settings*. Each setting receives an acceptable *value*, which can be numeric, alphabetic, or alphanumeric.

Caution: Some of the settings and values are critical to the operation of the Configuration Server. Do not alter or delete these unless instructed to do so by a member of Persistent Systems Technical Support.

The following is an example of the format of a section of the `edmpprof` file.

Example

```
[MGR_SECTION]
SETTING = VALUE
SETTING = VALUE
SETTING = VALUE
```

The following table presents a list of the `edmpprof` file sections, a brief description, and whether the section is required or optional.

Sections of the EDMPROF File

Section Name and Description	Required or Optional
MGR_ACCESS	Optional

Specifies access to the Administrator and Console functions.	
MGR_ATTACH_LIST Specifies the Configuration Server attach list that defines the programs to be attached when the Configuration Server is started.	Required
MGR_CACHE Specifies cache-processing options, such as cache segments, size, and statistics.	Optional
MGR_CLASS Specifies processing parameters for classes and instances.	Optional
MGR_DB_VERIFY Specifies the extent of automatic database verification.	Optional
MGR_DIAGNOSTIC Specifies the parameters for diagnostic Manager (zdiagmgr) tasks.	Optional
MGR_DIRECTORIES Specifies the path for the databases, REXX methods, and non-REXX methods.	Recommended but not required
MGR_DMA Specifies the parameters for Distributed Configuration Server operations.	Optional
MGR_ERROR_CONTROL Specifies how to process errors that are encountered while the Configuration Server is running.	Optional
MGR_LOG Specifies the logging directory and options for the Configuration Server logging facility.	Recommended, but not required
MGR_MESSAGE_CONTROL Specifies where messages are to be sent and whether they are to be suppressed.	Optional
MGR_METHODS Specifies options for method execution.	Recommended, but not required
MGR_NOTIFY Specifies the parameters for notify processing.	Optional
MGR_OBJECT_RESOLUTION Specifies the parameters used in object resolution.	Optional
MGR_POLICY Specifies the IP address/name, port, and details of Policy Resolution process of the Sentient Enterprise Policy Server (Policy Server), if it has been enabled.	Optional
MGR_POOLS Specifies the allocation of pool sizes on startup.	Optional

MGR_RESOLUTION_FILTERS Specifies the filtering rules for resolution, based on the connection type.	Optional
MGR_RETRY Specifies when an Sentient agent should attempt to reconnect to the Configuration Server, following rejection.	Optional
MGR_RIM Specifies the IP address/name and port of the Inventory Manager, if it has been enabled.	Optional
MGR_ROM Specifies information used for OS management.	Optional
MGR_RMP Specifies the IP address/name and port of the Portal, if it has been enabled.	Optional
MGR_SMTP_MAIL Specifies the parameters the Configuration Server uses to interface with SMTP.	Optional
MGR_SNMP Contains parameters to specify where SNMP traps are to be sent, and controls the behavior of the built-in SNMP agent.	Optional
MGR_SSL Specifies the parameters for the Sentient Enterprise Configuration Server SSL Manager task.	Optional
MGR_STARTUP Specifies the Configuration Server ID and TCP/IP port number.	Required
MGR_TASK_LIMIT Specifies the number of concurrent tasks permitted.	Required
MGR_TIMEOUT Specifies how long the Configuration Server will wait for a request from a connected Sentient agent before disconnecting it.	Optional
MGR_TPINIT Specifies communications packet sizes.	Required
MGR_TRACE Controls and influences diagnostic logging for the Configuration Server.	Optional
MGR_USERLOG Specifies the logging directory and options for the user logging facility.	Optional
OBJECT_SIZES Specifies the number of heaps and the heap size for CSDB objects that are being created on the Configuration Server as in-storage CSDB objects.	Optional

RCS_TUNING_CONTROL Provides a mechanism to override the default values that are specified in the Configuration Server self-tuning tool.	Optional
SECTION_DELIMITERS Specifies the left and right delimiters that are to be used for enclosing the section names within the Configuration Server <code>edmprof</code> file.	Optional

The remainder of this chapter documents each section of the `edmprof` file, covering the individual settings and the values for each. Also described is the effect of tunable settings on Configuration Server performance.

Caution: Before editing this file, be sure to review the important information in the section, ["Editing Configuration Server Settings" on page 500](#). Failure to do so could adversely affect your Sentient environment.

MGR_ACCESS

This section determines access to the Administrator and Console.

MGR_ACCESS Settings

Setting	Description
ADMIN	Specifies access to the Administrator. The values are DENY, ALLOW, and IGNORE.
CONSOLE	Specifies access to the Console. The values are DENY, ALLOW, and IGNORE.

Example

```
[MGR_ACCESS]
ADMIN = DENY
CONSOLE = DENY
```

MGR_ACCESS Values

Setting	Value as Installed	Default Value
ADMIN	DENY	DENY
CONSOLE	DENY	DENY

Note: Access can be controlled by native operating system security features also.

Performance and Usage Considerations

- ADMIN=ALLOW will provide access to the Administrator without checking the ZACCESS domain of the CSDB. Therefore, if ADMIN=ALLOW, and an Administrator attempts an action

for which no access rules have been defined, the attempted action will be permitted.

- When ADMIN=DENY, access rules governing Administrator actions are defined in the ZACCESS domain of the CSDB. Therefore, if ADMIN=DENY, the Administrator will be unable to perform an action unless there is an access rule specifically allowing it.
- If ADMIN=IGNORE the Administrator will be able to perform any action because the access rules will be ignored by the Configuration Server. Setting ADMIN=IGNORE essentially disables all access rules as they relate to Administrator functions.
- Access to the Console is determined by local password security policy. If CONSOLE=ALLOW and local security is not configured, read-only access is granted. If CONSOLE=IGNORE, local Console security is bypassed.

Sentient Administrator Access Level Values

Access Value	Definition
ALLOW	This value will result in the Configuration Server <i>not</i> checking the administrator access rules before granting access to an administrator to perform CSDB administrator functions. This value disables all administrator access-rule checks, even if they are defined in the database.
DENY	This value will result in the Configuration Server checking administrator access rules before granting access to an administrator to perform CSDB administrator functions. An administrator will be <i>unable</i> to perform an action unless there is an access rule defined in the database for that administrator specifically <i>allowing</i> it. This is the recommended setting when configuring administrator security.
	Note: If this option is specified, undefined administrators will not have access to any CSDB administrator functions, unless the ADMINID._NULL_INSTANCE_ is modified to allow such access.
IGNORE	This value will result in the Configuration Server checking administrator access rules before granting access to an administrator to perform CSDB administrator functions. An administrator will be <i>able</i> to perform an action unless there is an access rule defined in the database for that administrator specifically <i>prohibiting</i> it. Note: If this option is specified, undefined administrators will have full access to all CSDB administrator functions, unless the ADMINID instance exists for that administrator to prohibit such access.

MGR_ATTACH_LIST

In this section, specify which programs (Configuration Server tasks) are to be attached at startup, and set the options for these processes.

MGR_ATTACH_LIST Settings

Setting	Description
ATTACH_LIST_SLOTS	Number of slots for the attach list kept in shared memory of the Configuration Server. Every entry is 132 bytes long. It is recommended that this setting be one more than the number of CMD_LINE settings used. For example, if there are seven CMD_LINE settings, set this value to 8. Note: No process starts are required. However, system ability is limited if ZTCPMGR, ZREXXMGR, and ZNFYTMGR are not attached.
CMD_LINE	Command line to use when starting processes. Blanks are not permitted in CMD_LINE= substring. A second format is allowed when multiple instances of the same task are required and need to be separately identified. This format is: CMD_LINE=(NAME=,ADDR=,PORT=). These names should be unique within the Configuration Server.
LIMIT (=CLOSE)	The Configuration Server will drop any Sentient agent connection attempts when LIMIT=CLOSE is specified, <i>and</i> either the Task Limit or Storage Limit of the Configuration Server is reached. The Configuration Server will not return an object to the Sentient agent (no return EDMLOCTP), nor will it request the Sentient agent retry the connection. Note: This setting is valid with ztcpmgr only.
RESTART	Use this setting to determine if a Configuration Server task will be restarted when abnormally terminated. Blanks are not allowed in RESTART= substring. The default is NO .
RESTART_LIMIT	Number of attempts to restart an attached process that has terminated.
VERIFY_INTERVAL	Interval (in minutes) between verifications that attached processes are still running. If this value is 0, no verification will occur. The default is 1 (minute).

The following table lists the Configuration Server tasks.

Configuration Server Tasks

Task Name	Description
zbldpmgr	Patch Build Manager task
zdiagmgr	Diagnostic Manager task
znfytmgr	TCP Notify Manager task
zrexxmgr	REXX Manager task
zrtrymgr	Notify Retry Manager task
zsmtrmgr	SMTP receive Manager task
zsmtmgr	SMTP send Manager task

zsnmpmgr	SNMP Manager task
zsslmgr	SSL Manager task
ztcpmgr	TCP Manager task
zutilmgr	Utility Manager task

Example

[MGR_ATTACH_LIST]

ATTACH_LIST_SLOTS	= 15
RESTART_LIMIT	= 7
VERIFY_INTERVAL	= 5
CMD_LINE	= (zutilmgr) RESTART=YES
CMD_LINE	= (zrexsmgr) RESTART=YES
CMD_LINE	= (zsnmpmgr) RESTART=YES
CMD_LINE	= (zsmtrmgr) RESTART=YES
CMD_LINE	= (zsmtsmgr) RESTART=YES
CMD_LINE	= (znfytmgr) RESTART=YES
CMD_LINE	= (ztcpmgr) RESTART=YES
CMD_LINE	= (ztcpmgr LIMIT=CLOSE) RESTART=YES
CMD_LINE	= (zbldpmgr)

Table MGR_ATTACH_LIST Values

Setting	Value as Installed	Default Value	Minimum Value	Maximum Value
ATTACH_LIST_SLOTS	15	15	Number of CMD_LINES	Number of CMD_LINES + 1
CMD_LINE	Determined by installation options	Determined by installation options	N/A	N/A
RESTART	YES	NO	N/A	N/A
RESTART_LIMIT	7	7	0 = No restart	3200

Performance and Usage Considerations

- If the ATTACH_LIST_SLOTS value is too low, the Configuration Server will attach as many tasks as there are slots available. Any remaining tasks will not be attached until a slot becomes vacant. A too-high value could degrade overall system performance by unnecessarily setting aside resources that remain unused.
- The RESTART_LIMIT value should be set higher when critical Configuration Server functions are being performed. Note that regardless of the value in RESTART_LIMIT, the task will not be reinitiated if RESTART=NO.
- To ensure that vital processes continue running, set VERIFY_INTERVAL lower when critical Configuration Server functions are being performed. A higher setting, on the other hand, might save CPU cycles when total demand is a critical factor.
- The ztcpmgr task supports virtual IP addresses. It accepts the IP address and port number on the command line, as in:

```
CMD_LINE = (ztcpmgr addr=1.1.1.94,port=4438)
```


If the address is not specified, the machine address is used.

MGR_CACHE

This section specifies cache-processing options, such as cache size, statistics, load type, and error response.

Note: The settings in this section should be established based on operating system environment and performance needs.

MGR_CACHE Settings

Setting	Description
AVERAGE_OBJECT_SIZE	Average size of an object that will be cached.
CACHE_SEGMENTS	Number of cache segments.
CACHE_SIZE	Size of each cache segment.
CACHE_STATS	A YES/NO switch to accumulate statistics.
ICACHE_COUNT_ERROR	Instructs the Configuration Server on what action to take (shut down or log a warning) if the ICACHE instance counts don't match the DCS instance counts when it (the Configuration Server) is starting up. The default is SHUTDOWN . Note: The default is SHUTDOWN because running the Configuration Server without the correct number of cached items it is not recommended.

ICACHE_LOAD_TYPE	The DOMAIN.CLASS entries that are specified in the MGR_CLASS section are loaded, one class at a time, into Index cache via one of the following methods. AUTOMATIC instructs the Configuration Server to auto-determine the loader method (FULL or CHUNKY) to be used. This is the default. FULL instructs the Configuration Server to always use the FULL loader method—loading the database as a single entity. CHUNKY instructs the Configuration Server to always use the CHUNKY loader method—loading each CSDB service one at a time. For more information, see "ICACHE_LOAD_TYPE Considerations" on the next page .
ICACHE_SIZE	Size of the Index cache.

Example

[MGR_CACHE]

AVERAGE_OBJECT_SIZE	= 2048
CACHE_SEGMENTS	= 2
CACHE_SIZE	= 5242880
CACHE_STATS	= NO
ICACHE_COUNT_ERROR	= SHUTDOWN
ICACHE_LOAD_TYPE	= AUTOMATIC
ICACHE_SIZE	= 0

MGR_CACHE Values

Setting	Value as Installed	Default Value	Minimum Value	Maximum Value
AVERAGE_OBJECT_SIZE	2048 Bytes	2048	2048	6144
CACHE_SEGMENTS	2	0 (No caching)	0 (No caching)	System resource dependent
CACHE_SIZE	5,242,880 Bytes	0	0	System resource dependent
CACHE_STATS	NO	NO	N/A	N/A
ICACHE_COUNT_ERROR	SHUTDOWN	SHUTDOWN	N/A	N/A

ICACHE_LOAD_TYPE	AUTOMATIC	AUTOMATIC	N/A	N/A
ICACHE_SIZE	N/A	N/A	0 (No Index caching)	N/A
OBJECTID_VERDB_INITIAL_ENTRIES	N/A	N/A	Approximately 1.2 times the total number of instances in the database	N/A

Performance and Usage Considerations

Note: When modifying any of the cache parameters in this section, take care not to exceed the amount of virtual memory that is available. Also, sufficient virtual memory must be available to handle the maximum concurrent resolutions workload.

- Classes that are to be cached are defined in the MGR_CLASS section. At a minimum, the SYSTEM.PROCESS and SYSTEM.ZMETHOD classes should be cached.
- If the value of AVERAGE_OBJECT_SIZE is too low, the Configuration Server will have to reconfigure the cache until the object is accommodated. If the value of AVERAGE_OBJECT_SIZE is too high, the caching function might not be used efficiently.
- The ICACHE_SIZE setting should be used in conjunction with the CACHE_SEGMENTS setting and the ["MGR_CLASS" on the next page](#) section. ICACHE_SIZE will be enabled only if the value of CACHE_SEGMENTS is greater than zero.
- If ICACHE_COUNT_ERROR=SHUTDOWN and the Configuration Server shuts down, areas to check are:
 - Check the Configuration Server's log file,
 - Check the integrity of the connection to the database,
 - Verify the load type for the database connection method, and
 - Verify the validity of the database.

ICACHE_LOAD_TYPE Considerations

This section presents information that should be considered when specifying a value for ICACHE_LOAD_TYPE.

- If ICACHE_LOAD_TYPE=FULL, the Configuration Server will attempt to load into ICACHE, as a single element, all of the CSDB classes that are specified in the ["MGR_CLASS" on the next page](#) section. The Configuration Server sequentially accesses all instances in the specified directories.
- If ICACHE_LOAD_TYPE=CHUNKY, the Configuration Server will attempt to load all of the CSDB classes that are specified in the ["MGR_CLASS" on the next page](#) section—but as individual entities.

- **Component Classes** The Configuration Server uses the PACKAGE class in the current domain to control the loading of Component class instances—that is, all instances that are associated with a PACKAGE instance will get loaded into ICACHE, but orphan instances (those not associated with a PACKAGE instance) will not get loaded into ICACHE. The Configuration Server uses the PACKAGE class in the current domain to control the loading of Component class instances—that is, all instances that are associated with a PACKAGE instance will get loaded into ICACHE, but orphan instances (those not associated with a PACKAGE instance) will not get loaded into ICACHE.
- **Non-Component Classes** The Configuration Server uses a systematic approach to load the instances of instances of non-Component classes. The Configuration Server uses a systematic approach to load the instances of instances of non-Component classes.

Note: It is recommended to use the CHUNKY method for large databases because it minimizes the load on the network, thereby decreasing the likelihood of problems and the possibility of the database integrity being compromised.

- If a Configuration Server has a UNC connection (database path starts with \\) to the CSDB and the DMA count exceeds 300K instances, ICACHE_LOAD_TYPE=AUTOMATIC will choose the CHUNKY load method. See ["UNC Connectivity Issues" on page 518](#)

Purging Dynamic Cache

This section provides precautionary information about purging the dynamic cache on a Configuration Server. It includes information about protection that Sentient has introduced in order avoid purging the CSDB when a Sentient Enterprise Proxy Server (Proxy Server) is co-located with the Configuration Server and dynamic cache is enabled.

If the dynamic cache root is a CSDB then, by default, this parameter (default=0) prevents automatic dynamic cache purging of aged files when the dynamic cache index is saved. By default, the new parameter automatically safeguards against purging dynamic cache files from a CSDB.

To remove the safeguard and permit a purge of shared resource, dynamic cache files, set the parameter to 1, as shown in the following example.

Example

```
-dynamic -allow -shared -resource -purge 1
```

MGR_CLASS

This section specifies which classes and instances will be cached during the initialization of the Configuration Server.

MGR_CLASS Settings

Setting	Description

CLASS	Specifies which classes and instances will be cached at initialization.
	Format: DOMAIN.CLASS={VALUE1,VALUE2,VALUE3,VALUE4} If multiple domains have identical class names, the class templates must be identical. If they are not, performance will be adversely effected and objects, larger than necessary, might be created from the resolution process. <i>Cache_Class_Template_&_Base_Instance, Cache_All_Instances, Heap_Size, Maximum_Number_of_Heaps.</i>
	VALUE1 To cache (at Configuration Server startup) the <code>_BASE_INSTANCE_</code> and class template of the associated class, specify <code>Y</code> . Otherwise, specify <code>N</code> .
	VALUE2 To cache (at Configuration Server startup) all instances in the associated class, specify <code>Y</code> . Otherwise, specify <code>N</code> .
	VALUE3 This value is numeric and represents the initial size of the resolved object for the associated class. This is the size that each heap in the associated DOMAIN.CLASS will occupy in persistent objects. It should include any attributes that have been classed into the in-storage object as the result of resolution. Typically, the size of the transient class instance can be used for transient objects (such as, ZLOCMGR, ZLOCCLNT, ZSCHEDULE). The default (2048 bytes) can be refined for the ZSERVICE and PACKAGE classes, which are typically 3–4 KB in size. Examine the Sentient agent object resulting from a representative resolution to determine the most appropriate value. Values specified are generally in 512-byte increments.
	VALUE4 This is a numeric value that indicates an estimate of the number of heaps that are required for resolution. As each Sentient agent begins resolution, memory is allocated in blocks equal to the sum of all of the products of VALUE3*VALUE4. When memory is exhausted for a persistent class that is being cached, the next increment of storage is obtained in a block determined by the product of VALUE3*VALUE4 for the associated class. For example, if SOFTWARE.FILE = Y,Y,2048,100 was specified, the class template and <code>_BASE_INSTANCE_</code> are cached at startup. All of the instances of the SOFTWARE.FILE class are cached in memory (if there is sufficient room in <code>CACHE_SEGMENTS*CACHE_SIZE</code> for the whole class). The working value for the size of each FILE instance after resolution is estimated to be 2048 bytes. An initial allocation for the in-storage FILE object is made for 100 heaps that will require 100*2048 bytes (20 KB). If the resolution mode for the average Sentient agent requires 1000 FILE instances, then one initial allocation of 20 KB and nine subsequent allocations of 20 KB would be required to satisfy the entire 1000 FILE instance in-storage objects.

Note: At startup, classes are cached in the order they appear if VALUE1=Y and VALUE2=Y.

Example

[MGR_CLASS]

SYSTEM.PROCESS	= Y, Y, 2048, 1
SYSTEM.ZMETHOD	= Y, Y, 2048, 1
SOFTWARE.FILE	= Y, N, 4096, 1
SOFTWARE.REGISTRY	= Y, Y, 4096, 1
SOFTWARE.DESKTOP	= Y, Y, 2048, 1
SOFTWARE.ZSERVICE	= Y, Y, 3072, 1

MGR_CLASS Values

Setting	Value as Installed	Default Value
CLASS	N/A	N/A

Performance and Usage Considerations

- You can also specify your own unique classes in this section.
- Note that class instance size and number of instances specified in MGR_CLASS have an effect on storage and performance. A class instance size larger than the actual class size represents wasted storage. A class instance size smaller than the actual class size results in continual resolution performance degradation.
- Only the classes listed will be cached. All instances of listed classes are cached, if required, regardless of the values of the first two parameters in the list.
- You can use the following wildcard characters when you specify the DOMAIN and CLASS:

Wildcard Character	Behavior
&	Matches any string
!	Excludes whatever follows it

The ampersand (&) character is used instead of the asterisk (*), because the asterisk is used for commenting in the `edmprof.dat` file.

For example:

DOMAIN&.CLASS	!DOMAIN&.CLASS
DOMAIN.&	!DOMAIN.&
&.CLASS	!&.CLASS
DOMAIN&.&	!DOMAIN&.&

The following limitations apply:

- a. You can only use one & in the DOMAIN name and one & in the CLASS name.
- b. The maximum number of entries in MGR_CLASS is 256 (not counting those excluded with !).
- c. The maximum number of exclusions is 256.

MGR_DB_VERIFY

This section establishes the level of CSDB verification and whether errors are automatically corrected.

Note: If this function is configured, the Configuration Server will scan the CSDB for Y2K compliance, extended format (ZBASE), and appropriate CSDB ownership (Configuration Server IDs) at startup.

If errors are found, the CSDB utility can be run against the CSDB to correct the error. See ["Configuration Server Database Utility \(RadDBUtil\)" on page 667](#) for more information.

MGR_DB_VERIFY Settings

Setting	Description
DB_AUTOFIX	A YES/NO switch to determine if the CSDB should be automatically fixed (where possible) if errors are discovered. The default is NO .
VERIFY_DEPTH	Specifies the level of CSDB verification. The values are DOMAIN, CLASS, INSTANCE, and RESOURCE for this option. The default is CLASS . If DOMAIN is selected, the verification process will verify down to the DOMAIN level. If CLASS is selected, the verification process will verify down to the CLASS level. If INSTANCE is selected, the verification process will verify down to the INSTANCE level. If RESOURCE is selected, the verification process will verify the entire CSDB.

Example

[MGR_DB_VERIFY]

MGR_VERIFY_DEPTH	= CLASS
DB_AUTOFIX	= YES

MGR_DB_VERIFY Values

Setting	Value as Installed	Default Value
VERIFY_DEPTH	N/A	CLASS
DB_AUTOFIX	N/A	NO

Performance and Usage Considerations

- A value of VERIFY_DEPTH=RESOURCE will result in longer startup times, as the level of detail is deeper. Conversely, VERIFY_DEPTH=CLASS will result in a much quicker startup time, because the level of verification is decreased.
- Refer to the MGR_ERROR_CONTROL section, which offers handling parameters for errors found during the CSDB verification process.

MGR_DIAGNOSTIC

This section establishes the values that the Configuration Server will use to verify that sufficient disk space is available for CSDB operations and logging, as well as the frequency of verification occurrences.

MGR_DIAGNOSTIC Settings

Setting	Description
DIAGNOSTIC_INTERVAL	Interval (in seconds) for the diagnostic Configuration Server to verify that sufficient disk space is available for the CSDB and log. If set to 0, monitoring is disabled.
DIAGNOSTIC_MIN_DB_BYTES	The minimum number of bytes established for CSDB operations. The maximum value is 2 GB.
DIAGNOSTIC_MIN_LOG_BYTES	The minimum number of bytes established for logging operations. The maximum value is 2 GB.

Example

[MGR_DIAGNOSTIC]

DIAGNOSTIC_INTERVAL	= 900
DIAGNOSTIC_MIN_DB_BYTES	= 50
DIAGNOSTIC_MIN_LOG_BYTES	= 25

MGR_DIAGNOSTIC Values

Setting	Value as Installed	Default Value
DIAGNOSTIC_INTERVAL	N/A	900
DIAGNOSTIC_MIN_DB_BYTES	N/A	50M
DIAGNOSTIC_MIN_LOG_BYTES	N/A	25M

Performance and Usage Considerations

- Include `CMD_LINE=(zdiagmgr)` in the `MGR_ATTACH_LIST` section to configure and use this setting.
- On a Configuration Server Database:
 - When the `DIAGNOSTIC_MIN_DB_BYTES` threshold (2GB) is reached, the following will be issued:
 - an SNMP trap of 2040.
 - a message to the Configuration Server log:
(9282 - Warning: The volume containing the Configuration Server Database has only %.0lf free bytes).
 - When the `DIAGNOSTIC_MIN_LOG_BYTES` threshold (2GB) is reached, the following will be issued:
 - an SNMP trap of 2045.
 - a message to the Configuration Server log:
(9283 - Warning: The volume containing the Configuration Server log has only %.0lf free bytes)
- On the Configuration Server, you can program a REXX that calls the `EDMMAILQ` method to send a notification e-mail.
- Set the following line in the `MGR_MESSAGE_CONTROL` section of the `edmprof` file,
`9282, 9283=LOG,EVENTLOG`
(This triggers messages 9282 and 9283 to be created in the Configuration Server log and Windows Event Log.)

MGR_DIRECTORIES

This section specifies the path for the databases, REXX methods, and non-REXX methods. See ["MGR_LOG" on page 521](#) to specify the directory path for the Configuration Server log.

- The `EXPORT_PATH` setting can be used to define a directory for export operations.
- The five `USER_PATHn` settings can be customized for each Client Automation environment.

MGR_DIRECTORIES Settings

Setting	Description
DBPATH	Fully qualified directory path for object databases.
EXPORT_PATH	Fully qualified directory path for EXPORT operations, accessible via REXX.
METHOD_PATH	Fully qualified directory path for non-REXX methods.
REXX_PATH	Fully qualified directory path for REXX methods.
USER_PATH1	Working directory to be used by users, accessible via REXX.

USER_PATH2	Working directory to be used by users, accessible via REXX.
USER_PATH3	Working directory to be used by users, accessible via REXX.
USER_PATH4	Working directory to be used by users, accessible via REXX.
USER_PATH5	Working directory to be used by users, accessible via REXX.

Example

For Windows:

[MGR_DIRECTORIES]

DBPATH	= D:/MGR/DB
EXPORT_PATH	= D:/MGR/BIN/EXPORT
METHOD_PATH	= D:/MGR/BIN
REXX_PATH	= D:/MGR/REXX
USER_PATH1	= D:/MGR/BIN/USER1
USER_PATH2	= D:/MGR/BIN/USER2

MGR_DIRECTORIES Values

Setting (REXX Name)	Value as Installed	Default Value
DBPATH	As specified during installation	<i>Current_directory</i>
EXPORT_PATH (EXPTPATH)	As specified during installation	<i>Current_directory</i>
METHOD_PATH	As specified during installation	<i>Current_directory</i>
REXX_PATH	As specified during installation	<i>Current_directory</i>
USER_PATH1 (USRPATH1)	As specified during installation	<i>Current_directory</i>
USER_PATH2 (USRPATH2)	As specified during installation	<i>Current_directory</i>
USER_PATH3 (USRPATH3)	As specified during installation	<i>Current_directory</i>
USER_PATH4 (USRPATH4)	As specified during installation	<i>Current_directory</i>
USER_PATH5 (USRPATH5)	As specified during installation	<i>Current_directory</i>

Performance and Usage Considerations

- The REXX directory specified in this section is further defined by the samples subdirectory, which contains a set of sample REXX methods.

Sentient supports Universal Naming Code (UNC), which enables paths in the `edmprof` file to be specified as shown below:

```
\\VFH_LAPTOP\DRIVE_D\CMCSDB
```

Note: When using UNC, the address must be preceded by two backslashes (\\), with a single backslash (\) used to separate each subfolder.

UNC Connectivity Issues

Interruptions in UNC connectivity to a CSDB might result in critical database classes not being accessed during the resolution process. This failed access could lead to incomplete and erroneous resolution of Client Automation services and, possibly, the inadvertent removal of applications. If this happens, the following errors will appear in the Configuration Server log:

```
NVD7005E 06:19:52 <172.26.132.24 /1930> Radia Client--! ERROR: CLASS  
<PRIMARY.POLICY.USER> NOT FOUND
```

```
NVD7005E 06:19:52 <172.26.132.24 /1930> Radia Client--! ERROR: CLASS  
<PRIMARY.POLICY.ZBASE> NOT FOUND
```

```
ERROR: CLASS <LICENSE.80d40ad0fd2a4ded8c9d26254e8daf0c.MDEVICE> NOT  
FOUND
```

```
NVD5113E 02:25:17 <172.31.18.5 /668> Radia Client--! ERROR RC=<4>  
CREATING INSTANCE<LICENSE.80d40ad0fd2a4ded8c9d26254e8daf0c.MDEVICE.  
3B06A9B26C5047D886942D1E2EC4A46E>
```

Also, during Configuration Server startup, the following will be seen in the Configuration Server log:

```
Configuration Server Database is on <Remote> <NTFS> <Uncompressed>  
Drive <\\example\RCSDB\>
```

```
NVD9268I 02:17:34 <ztoptask /FB4> System Task--- Drive  
<\\example\RCSDB\> supports <255> character file names
```

```
NVD9269I 02:17:34 <ztoptask /FB4> System Task--- Drive  
<\\example\RCSDB\> supports <Case-Sensitivity Case-Preservation  
Unicode File-Compression>
```

```
NVD9271I 02:17:34 <ztoptask /FB4> System Task--- Database resides in  
<\\example\RCSDB\DB\>
```

Note: The UNC-mapped drive will be reflected in the DBPATH setting of the MGR_DIRECTORIES section in the `edmprof` file.

Recommended Preventive Measures

Note: In addition to the preventive measures detailed here, it is recommended that the Configuration Server be monitored—especially if UNC disconnects are frequently occurring.

It is recommended to take the following measures to minimize the effect of UNC connectivity interruptions.

- At Configuration Server startup, cache all CSDB classes that are used for resolution. (See "MGR_CACHE" on page 508, and "MGR_CLASS" on page 511.)
- In the MGR_CACHE section of the `edmprof` file, type `ICACHE_LOAD_TYPE=CHUNKY`. (See "MGR_CACHE" on page 508.)
- In the CSDB ensure the following settings for the PRIMARY.SYSTEM.ZMETHOD.LDAP_RESOLVE method:
 - ZMTHTYPE (method type) is set to `REXX`
 - ZMTHNAME (method name) is set to `RADISH`

Caution: If policy resolution is being driven by a method other than `LDAP_RESOLVE`, and the configuration is uncertain, contact Persistent Systems Technical Support before making any changes to the CSDB.

These settings will result in the Sentient agent connects and resolutions being stopped—preventing the Sentient agents from doing anything, thereby protecting them in the event they are presented an empty catalog. This also prevents the removal of applications.

MGR_DMA

This section specifies the timeout parameter and the directory path for the Distributed Configuration Server (formerly known as the Distributed Manager Adapter, DMA), and enables you to specify values for these options.

Note: You must have the Distributed Configuration Server installed to enable these parameters.

MGR_DMA Settings

Setting	Description
DMA_TIMEOUT	Maximum interval (in seconds) that the Distributed Configuration Server will wait for tasks to complete before allowing a commit. A value of 0 means wait indefinitely.
DMA_STAGE_PATH	Path in which staging directories will be created.

Examples

Windows Example:

[MGR_DMA]

DMA_TIMEOUT	= 0

DMA_STAGE_PATH	= D:\MGR\

MGR_DMA Values

Setting	Value as Installed	Default Value
DMA_TIMEOUT	N/A	600
DMA_STAGE_PATH	N/A	ZTOPTASK Path

Performance and Usage Considerations

- To decrease the chance of the Distributed Configuration Server synchronization timing out without having committed database updates, increase the value of DMA_TIMEOUT.

MGR_ERROR_CONTROL

This section specifies error-handling parameters for the Configuration Server.

MGR_ERROR_CONTROL Settings

Setting	Description
DBERROR	Describes the type of error and the response. The format is: (<i>Error Type</i>) = (<i>Response</i>) Valid values for (Response) are: [SHUTDOWN, IGNORE], [NOEMAIL, EMAIL], [NOSNMP, SNMP]
UserEmailErrorsTo	E-mail address of the administrator to whom error messages should be sent.

Note: The actions taken by the settings in this section depend on the levels specified for VERIFY_DEPTH and the errors discovered during MGR_DB_VERIFY processing.

Example

[MGR_ERROR_CONTROL]

DBERROR	= IGNORE, EMAIL
UserEmailErrorsTo	= administrator@yourcompany.com

MGR_ERROR_CONTROL Values

Setting	Value as Installed	Default Value
DBERROR	N/A	[SHUTDOWN] [NOEMAIL] [NOSNMP]
UserEmailErrorsTo	N/A	N/A

Performance and Usage Considerations

- Currently, the only DBERROR supported is DBError. Examples of DBErrors are instances with 0 length, attempting to load a template that does not exist, and so forth.
- The UserEmailErrorsTo value must be a valid e-mail address.

MGR_LOG

This section specifies the logging directory and logging options for the Configuration Server logging facility. It also provides detailed information about reclaiming dormant Sentient agent-device licenses.

MGR_LOG Settings

Setting	Description
DIRECTORY	Fully qualified directory path where the Configuration Server log is written.
DISABLE_NT_EVENT_LOGGING	If YES is specified, the Configuration Server logger will disable its Windows Event logging support. This means that messages sent to the Configuration Server log will not be sent to the Windows Event log even if EVENTLOG is specified in the section MGR_MESSAGE_CONTROL. If NO is specified, such messages will be echoed to the Windows Event log if EVENTLOG is specified in the MGR_MESSAGE_CONTROL section. Note: This parameter affects only the event logging of Configuration Server log messages. Some Windows Event log records are written without any corresponding Configuration Server log messages. <i>Windows only.</i>
DISABLE_SNMP_TRAP_LOGGING	If YES is specified, the Configuration Server logger will disable its SNMP trapping support. This means that messages sent to the Configuration Server log will not be sent to the primary SNMP Manager as traps, even if SNMPTRAP is specified in the section MGR_MESSAGE_CONTROL. If NO is specified, such messages will be sent to the SNMP Manager as traps, if SNMPTRAP is specified in the MGR_MESSAGE_CONTROL section. Note: This parameter affects only the trapping of Configuration Server log messages. Some SNMP traps are issued without any corresponding Configuration Server log messages.
FLUSH_SIZE	The number of bytes between automatic flushes of operating system buffers for Configuration Server log file.
MESSAGE_DATE	Allows insertion of the date into every line in the log. Values are JULIAN (YYYYDDD), GREGORIAN (DDMMYYYY), and ISO (YYYYMMDD).
MESSAGE	The left and right delimiters that are used for enclosing the task name and

	task ID in the Configuration Server log messages. The format is MESSAGE_DELIMITER=xy, where x is the left delimiter and y is the right delimiter. The options are: [], (), < >, and { }. The default is [].
MESSAGE_PREFIX	The 3-digit, alphabetic, Configuration Server identifier that will precede Configuration Server log messages. Specify any 3-digit, alphabetic value. The default is NVD. Note: If you have log scrapers, verify that they work properly with the NVD setting. If not, change this to RAD.
MESSAGE_WIDTH	The maximum width (in bytes) of the messages in the Configuration Server log.
PIPE_SIZE	The maximum amount (in bytes) of log messages that can be queued by the Configuration Server logging facility while the log file is busy. When the value of PIPE_SIZE is reached, any task that issues a log message will freeze until the pipe starts emptying.
SWITCH_TOD	A time-of-day specification that will, each day, automatically trigger log switching (see "MGR_LOG" on the previous page). The value must be 5 characters, expressed in base-24 time (00:00–23:59), with a colon separator, as in HH:MM. Notes: This is a scheduled event that causes the log to switch; it is independent of, and runs regardless of, any previous log-switching activity. This setting can be used to enable license reclamation using ZLICUTIL.EXE. For more information, see "MGR_LOG" on the previous page . This setting is not part of the Configuration Server installation; it must be manually added. It is recommended that this be set to an off-peak, low-activity time.
THRESHOLD or THRESHHOLD (both accepted)	The maximum number of lines that will be written to the Configuration Server log before it is automatically switched to the next log. When the limit is reached, a new log file is created, regardless of SWITCH_TOD setting. Note: For a more detailed description of this setting and how it can be used for license reclamation, see "MGR_LOG" on the previous page.

Examples

Windows Example:

[MGR_LOG]

DIRECTORY	= D:\Program Files\PSL\RCA\ConfigurationServer\log
FLUSH_SIZE	= 100000
MESSAGE_DATE	= JULIAN

MESSAGE_DELIMITER	= []
MESSAGE_PREFIX	= NVD
MESSAGE_WIDTH	= 256
PIPE_SIZE	= 1000000
SWITCH_TOD	= 23:38
THRESHOLD	= -5000000

MGR_LOG Values

Setting	Value as Installed	Default Value	Minimum Value
DIRECTORY	Program Files\PSL\RCA \ConfigurationServer \log PSL\RCA /ConfigurationServer/lo g	<InstallDir> \ConfigurationServer\l og PSL\RCA/ ConfigurationServer/lo g	N/A
DISABLE_NT_EVENT_LOGGING	Varies across platforms.	NO	N/A
DISABLE_SNMP_TRAP_LOGGING	Varies across platforms.	NO	N/A
FLUSH_SIZE	1000 bytes	5000 bytes	1
MESSAGE_DATE	N/A	N/A	N/A
MESSAGE_DELIMITER	N/A	[]	N/A
MESSAGE_PREFIX	NVD	NVD	N/A
MESSAGE_WIDTH	256	90	80
PIPE_SIZE	1000000 bytes	1 MB	1 MB
SWITCH_TOD	N/A	N/A	N/A
THRESHOLD	-5000000 lines	100000 lines	1

Performance and Usage Considerations

- Increasing the FLUSH_SIZE will enhance performance, but will delay messages flushed to the log file.
- Increase MESSAGE_WIDTH if log messages are being truncated.
- When closely monitoring system status using the Configuration Server log, set THRESHOLD to a positive value to create and save successive portions of the Configuration Server log. If disk storage space is critical, set the value to a negative number to reuse the allocated log disk space.
- If numerous Configuration Server methods are being invoked, use the MGR_METHODS.LOG_LIMIT setting to control the size of the Configuration Server log for each method. Additionally, the MGR_TASK_LIMIT.TASK_LOG_LIM setting controls the number of messages printed by the execution of each task.
- When modifying parameters in this section as they relate to memory or disk utilization, take care not to exceed the maximum amount of memory or storage space available.

Log Switching

Log Switching

By default, the Configuration Server log will be switched according to the log size value of "[Log Switching](#)" above and the amount of Configuration Server activity. However, log switching can be configured to automatically occur on a daily basis, at a scheduled time. This is done by specifying the "[Log Switching](#)" above setting and modifying the THRESHOLD setting. THRESHOLD can also dictate what happens to the log that gets rolled over. These sections describe how to do this.

SWITCH_TOD

Specify the time-of-day for the Configuration Server log to automatically roll over. This setting is independent of THRESHOLD, but can be used with it to trigger either of two REXX methods (ZLOGSWCH.REX or ZLOGWRAP.REX) to launch the user-implemented license-reclamation utility, ZLICUTIL, as described in the section, "[SWITCH_TOD](#)" above.

THRESHOLD

This setting determines how large the Configuration Server log can be before it is switched to a new log. The value of this setting will dictate what happens to the log that is rolled out, as described in this section.

- If THRESHOLD $\geq$ 0, the old log will be renamed and saved, and the Configuration Server REXX method, ZLOGSWCH, will run.
The Configuration Server's log directory (ConfigurationServer\log) will have multiple log files whose names conform to the following operating system-specific conventions.

Note: ISO is the International Standards Organization.
Configuration Server ID is the value of the `edmprof` file's setting `MGR_STARTUP.MGR_NAME`.

Windows

- The *standard* log naming format is the Configuration Server log prefix (`nvd`), followed by the letter **r**, and the Configuration Server ID:
`nvdmr001.log`
- The *log switch* format is the **r** replaced by an **s**, followed by the Configuration Server ID, and the ISO-formatted date and time (each preceded by an underscore) appended:
`nvdms001_20050427_075835.log`

Note: This THRESHOLD setting will result in multiple Configuration Server log files. To differentiate between them, check the date on which they were modified.
It is recommended to use this setting because it results in saved, rather than deleted, log files.

- If `THRESHOLD < 0`, the old log will be overwritten (but not deleted), and the Configuration Server REXX method, `ZLOGWRAP`, will run.
The Configuration Server log directory will have just one old log file whose name conforms to the following operating system-specific conventions.

Note: ISO is the International Standards Organization.
Configuration Server ID is the value of the `edmprof` file's setting `MGR_STARTUP.MGR_NAME`.

Windows

- The *standard* log naming format is the Configuration Server log prefix (`nvd`), followed by the letter **r**, and the Configuration Server ID:
`nvdmr001.log`
- The *log wrap* (dump) format is the **r** replaced by a **d**:
`nvdmd001.log`

At the next log switch, the previous log (from the last wrap) will be replaced/overlaid by the newly switched log.

Delete this text and replace it with your own content.

ZLICUTIL

The Configuration Server REXX utility, `ZLICUTIL`, can be used with `ZLOGSWCH` and `ZLOGWRAP` for daily license reclamation.

For more information, see ["License Reclamation" on the next page](#).

License Reclamation

A Configuration Server license string supports a certain number of Sentient agent devices. To keep only active, current devices in a license count, the Configuration Server enables the reclamation of Sentient agent-device licenses. If an Sentient agent device hasn't connected to the Configuration Server for a specified number of days, its license becomes inactive and can be reclaimed in the license count by the Configuration Server utility, ZLICUTIL.

Considerations

- By default, license reclamation occurs only at Configuration Server shutdown.
- To make license reclamation occur daily:
 - Specify values for SWITCH_TOD and THRESHOLD (see ["Considerations" above](#) and ["Considerations" above](#) or in ["MGR_LOG" on page 521](#)).
 - Modify the appropriate REXX (either ZLOGSWCH or ZLOGWRAP) so that a log-switching REXX method will launch the license utility, ZLICUTIL, when it gets called.
- Licenses that are aged out via ZLICUTIL are not immediately available. They are reclaimed only at the next midnight.

Note: The exception to this condition is that aged-out licenses will be immediately available when the Configuration Server is restarted.

The following instructions detail how to implement license reclamation on a Configuration Server.

To Reclaim Licenses Daily

1. Add the log switch setting, SWITCH_TOD, to the MGR_LOG section of the `edmprof` file, and specify a valid value (see SWITCH_TOD in the table, ["To Reclaim Licenses Daily" above](#) such as:
`[MGR_LOG]SWITCH_TOD = 23:45`
2. Specify a value for THRESHOLD or, as described .
3. Navigate to the directory in which the Configuration Server methods reside, such as:
`<InstallDir>\ConfigurationServer\rexx\NOVADIGM`
and copy either ZLOGSWCH or ZLOGWRAP (or both) up one level to the `rexx` directory.

Caution: During the Configuration Server installation, the REXX methods are, by default, installed to the platform-specific directories that are listed in step 3. Before making changes to either ZLOGSWCH or ZLOGWRAP, be sure to copy it up one level to the `rexx` directory

4. Open the ZLOGSWCH (or ZLOGWRAP) file, and add the commands shown here:
`call edmget zcvt
ADDRESS CMD "ZLICUTIL" zcvt.dbpath zcvt.rptpath zcvt.uuid
zcvt.mgrid zcvt.syspath || license.nvd`

Note: See the section, "[Configuration Server as a Windows Service](#)" on page 496, for Windows-specific configuration information.

5. Save and close ZLOGSWCH (or ZLOGWRAP).
From this point on, the license reclamation feature will be started asynchronously when the log switch is invoked.

Configuration Server Running as a Windows Service

This section describes the changes that are necessary if the Configuration Server is running as a service on a Windows machine.

Note: This information is relevant *only* if the Configuration Server is configured as a Windows service; if it is not, this information is not applicable.

The directory in which Windows always first looks for its “service” programs is `Windows\system32`. Therefore, in order for a program to be run as a service, it must be located here; if it is not, it won’t be found by Windows—or Windows has to be instructed to look elsewhere.

Even though the Configuration Server might be set up to run as a Windows service, it probably isn’t in this directory—rather, it is likely in the directory that was specified during the installation (the default of which is `<InstallDir>\ConfigurationServer`). Therefore, Windows has to be told where to find the program files. Instructions for doing this are in the following section.

Directing Windows to the Configuration Server Program Files

To resolve the issue of Windows not being able to find the Configuration Server files, make the following changes to the ZLOGSWCH (or ZLOGWRAP) REXX method.

1. Specify the *fully qualified path* of the directory in which ZLICUTIL resides so that Windows will know where to look. (See the example that follows.)
2. Add a fifth argument—the *fully qualified path* of the directory in which the file `license.nvd` resides.

Examples

Original ZLOGSWCH (or ZLOGWRAP) REXX method command:

```
ADDRESS CMD "START ZLICUTIL" zcvt.dbpath  
"<InstallDir>\ConfigurationServer\log" "05c87c3e95194a9d9251fee5cbfddafb"  
zcvt.mgrid
```

Revised ZLOGSWCH (or ZLOGWRAP) REXX method command:

```
ADDRESS CMD "START FULLY_QUALIFIED_BIN_PATH\ZLICUTIL.EXE" zcvt.dbpath  
"<InstallDir>\ConfigurationServer\log" "05c87c3e95194a9d9251fee5cbfddafb"  
zcvt.mgrid "FULLY_QUALIFIED_BIN_PATH\LICENSE.NVD"
```

MGR_MESSAGE_CONTROL

This section specifies which log messages are to be sent and to where, or if they are to be suppressed.

MGR_MESSAGE_CONTROL Settings

Setting	Description
<i>nnnnn = (Destination)</i>	<i>(Message_Number) = (Message_Destination)</i>

Note: The left side of the equals sign (=) specifies which messages are to be affected by the command. There is an "ALL" directive that can be used to affect all messages (0001-9999). Also, if there is no destination after the equals sign, the message has no associated destination, so it is suppressed.

The following table presents a list of the six destinations for log messages.

MGR_MESSAGE_CONTROL Log Message Destinations

Destination	Description
EVENTLOG	Write log messages to the Windows Event log. (<i>Windows only</i>)
LOG	Write log messages to the Configuration Server log.
REXX	Write log messages to a pre-determined REXX (named MGRLOG). Important: MGRLOG is not an existing REXX, and therefore, must be created to use this facility. Note: The MGRLOG REXX will receive the message number and the message text as its first and second input parameters. You can then process the information in whatever manner you chose.
SNMPTRAP	Write log messages as traps to the current SNMP Manager.
USERLOG	Write log messages to the user log. Note: You must first enable the user log feature in the MGR_USERLOG section of the <code>edmprof</code> file, by specifying <code>ACTIVATE=YES</code> .

Example

[MGR_MESSAGE_CONTROL]

ALL	= SNMPTRAP
500	=
520-600	= LOG

225, 300	=
220-299, 400, 542-545, 803	= EVENTLOG, REXX, USERLOG

In this example:

- The first line will send all messages to the SNMP Manager as traps.

Note: Keep in mind that this means that no messages will be written to the Configuration Server log; all messages will go out as SNMP traps only, unless subsequent entries specifically override this.

- The second line will cause message 500 to be suppressed—that is, it will not be written to the Configuration Server log.
- The third line will cause messages 520 through 600 (inclusive) to be written to the Configuration Server log.
- The fourth line will suppress messages 225 and 300 only; all others will be written to the Configuration Server log.
- The fifth line will cause messages 220 through 299 (inclusive), 400, 542 through 545 (inclusive), and 803 to be written to the Windows Event log, the pre-determined REXX, and the user log.

MGR_MESSAGE_CONTROL Values

Setting	Value as Installed	Default Value
<i>nnnnn = (Destination)</i>	N/A	ALL=LOG

Performance and Usage Considerations

- SNMPTRAP should be used as a destination only if the address of the SNMP Manager has been configured in the SNMP section.
- Any line that does not contain an equals sign (=) is treated as a comment. In addition, lines that begin with an asterisk (*), double forward slashes (//), or a slash (/) are treated as comments. Blanks and tabs can occur anywhere on the line, even ahead of the comment specifications.
- ALL = will suppress all messages.
- Any errors encountered in parsing a line cause the entire line to be ignored and an error message to be written to STDERR.

MGR_METHODS

This section specifies options for method execution.

MGR_METHODS Settings

Setting	Description
LOG_LIMIT	Maximum number of messages that a method can issue to the Configuration Server log. When this limit is reached, a message will be written stating that the message limit has been reached and that all other messages from the method will be ignored.
TIMEOUT	The duration (in seconds) that a Configuration Server will wait for a response from a method that is running. After this interval, if no response is received, the Configuration Server will terminate the method.

Example

[MGR_METHODS]

LOG_LIMIT	= 0
TIMEOUT	= 300

MGR_METHODS Value

Setting	Value as Installed	Default Value	Minimum Value	Maximum Value
LOG_LIMIT	0	0 = No Limit	0	2,147,483,647
TIMEOUT	300 (seconds)	60 (seconds)	0	32000

Performance and Usage Considerations

- The number of messages generated by the execution of a method is also affected by the TASK_LOG_LIM setting in MGR_TASK_LIMIT.
- Tune system resource usage with the TIMEOUT setting. If system resources are critical, lower the TIMEOUT setting to free up unused processing cycles.
- When the TIMEOUT value is reached, the method is terminated and messages are written to the Configuration Server log.
- If the TIMEOUT=0, the method will continue in effect until its conclusion.

MGR_NOTIFY

This section specifies the defaults for the Configuration Server's Sentient agent notification.

MGR_NOTIFY Settings

Setting	Description
ISSUE_WAKE_ON_LAN	Enables support for TCP Wake-On-LAN.

NFY_RETRY	Number of times to attempt re-notification after initial, unsuccessful attempt.
NFYT_TIMEOUT	Notify timeout (in seconds) for TCP/IP Sentient agents.
RECOVERY_DOMAIN	The Domain that the Retry Manager will access to reinitiate notifies after the Configuration Server has prematurely shut down.
SUBNET_MASK	This value is used to convert a destination IP address into a subnet address for EDMWAKE. Note: Applicable only to version 4.3 CM Configuration Server, with Service Pack 2.
WAKE_ON_LAN_TTL	This value is the number of routers that the WOL broadcast is permitted to pass. The default is 99 . Note: Before establishing this value, consult your network administrator.

Example

[MGR_NOTIFY]

NFYT_TIMEOUT	= 120
NFY_RETRY	= 5
SUBNET_MASK	= 255.255.0.0
ISSUE_WAKE_ON_LAN	= YES
RECOVERY_DOMAIN	= ALL
WAKE_ON_LAN_TTL	= 99

Note: In some network addressing schemes, a class A IP address is used with a class B or class C subnet mask. This often results in problems, caused by parameters passed by a Configuration Server while trying to invoke EDMWAKE. If specified, the SUBNET_MASK parameter will cause the Notify Manager to use this mask from the `edmprof` file, rather than create the subnet mask according to the network type.

For example, if the notify for IP address, 192.168.5.5 failed, EDMWAKE will be issued for subnet address, 192.168.255.255.

MGR_NOTIFY Values

Setting	Value as Installed	Default Value	Value Range
ISSUE_WAKE_ON_LAN	N/A	NO	YES/NO
NFY_RETRY	N/A	0	32000
NFYT_TIMEOUT	120 seconds	0	32000
RECOVERY_DOMAIN	N/A	ALL	RETRY/ALL

SUBNET_MASK	N/A	N/A	N/A
WAKE_ON_LAN_TTL	99	99	99

Performance and Usage Considerations

- Establish MGR_NOTIFY settings based on network operations parameters.
- The MGR_NOTIFY settings should be coordinated with values in the MGR_RETRY and MGR_TASK_LIMIT sections.
- In order for the Wake-On-LAN Notify function to operate, and to enable the retrying of failed operations, zrtrymgr must be specified under MGR_ATTACH_LIST.

MGR_OBJECT_RESOLUTION

This section specifies the parameters to use during object resolution.

MGR_OBJECT_RESOLUTION Settings

Setting	Description
ALLOW_DUPPLICATE_INSTANCES	Allow or disallow duplicate instances to be used during object resolution. Values are YES and NO.
ALWAYS_CALL_ZADMIN	Force object resolution to call the ZADMIN method to process the ZADMIN object. Values are YES and NO.
ZERRORM_MAX_ERRORS	Maximum number of errors associated with a ZERRORM event.
ZERRORM_MAX_WARNINGS	Maximum number of warnings associated with a ZERRORM event.

Example

[MGR_OBJECT_RESOLUTION]

ALWAYS_CALL_ZADMIN	= YES
ZERRORM_MAX_WARNINGS	= 50
ZERRORM_MAX_ERRORS	= 50
ALLOW_DUPPLICATE_INSTANCES	= YES

MGR_OBJECT_RESOLUTION Values

Setting	Value as Installed	Default Value

ALWAYS_CALL_ZADMIN	YES	NO
ALLOW_DUPLICATE_INSTANCES	N/A	YES
ZERRORM_MAX_WARNINGS	N/A	50
ZERRORM_MAX_ERRORS	N/A	50

Performance and Usage Considerations

- ALLOW_DUPLICATE_INSTANCES=NO will cause the Configuration Server to eliminate duplicate services at resolution. This might result in the elimination of duplicate SOFTWARE.FILE instances on the Sentient agent. As the size of the SOFTWARE.FILE object expands, the resolution time will increase.
- It is recommended that you do not modify or remove ALWAYS_CALL_ZADMIN, as doing so might prohibit administrator type functions.

MGR_POLICY

This section specifies the IP address/name, port, and details of Policy Resolution process of the Sentient Enterprise Policy Server (Policy Server), if it has been enabled.

Note: This section is present only if the Policy Server option is selected during the Configuration Server installation.

MGR_POLICY Settings

Setting	Description
HTTP_HOST	The IP address of the Policy Server. If the Policy Server is co-located with the Configuration Server, specify <code>localhost</code> .
HTTP_PORT	The port of the Policy Server. The default is 3466 .
RADISH_DEBUG	Provides details of Policy Resolution process. If information messages on details of the Policy Resolution process are required, set RADISH_DEBUG=1. The default is 0 .

Example

[MGR_POLICY]

HTTP_HOST	= localhost
HTTP_PORT	= 3466
RADISH_DEBUG	= 1

MGR_POLICY Values

Setting	Value as Installed	Default Value	Minimum Value	Maximum Value
HTTP_HOST	None	N/A	N/A	N/A
HTTP_PORT	N/A	3466	N/A	N/A
RADISH_DEBUG	0	0	N/A	1

MGR_POOLS

Note: This section of the edmpof file is specific to Windows operating systems.

This section enables you to specify allocations (pools) of memory in different sizes, before Configuration Server startup. These allocations can be changed dynamically while the Configuration Server is running by using modify commands. In addition, you can establish percentage-of-waste tolerances (the amount of memory that is not used by a specific requirement) for each of the allocations.

You can establish pools of any size. The pool sizes specified, however, should be multiples of eight. If not, they will be rounded up to the next multiple of eight by the system. Likewise, you can specify any number of pools.

Note: The sizes and number of pools you establish, as well as the percentage of waste tolerated, should be based on the workload and operating requirements of your environment.

When the Configuration Server requires memory, the pools are searched from smallest to largest until a pool is found in which the memory requirement fits. When that pool is found, the first item on the free list is used to resolve the request—providing that the percent of space wasted is not above the value specified for that pool. If there is no item on the free list, then the memory allocation is redirected to the standard C heap. If the C heap is also exhausted, a host system native-storage request will be performed to satisfy the memory request.

The format for specific pool allocation is:

(Pool Size) = (number of elements for that pool size),
(specific percentage of waste tolerated),(expansion increments).

Note: The specific percentage of waste tolerated parameter is optional. The expansion increments default is 1.

MGR_POOLS Settings

Setting	Description
WASTE_TOLERATED	The general percentage of waste that will be tolerated to fit a specific requirement to an available pool. The default is 100 (%).
ALLOCATION_	Deny memory allocations above this size.

SIZE_ERROR_THRESHOLD	
ALLOCATION_SIZE_REPORTING_THRESHOLD	Report memory allocations above this size.
XXXXXX	The allocation for the XXXXXX byte pool. Any reasonable number of these can be specified.

By default, the MGR_POOLS section establishes the pool sizes shown in the example below.

Example

[MGR_POOLS]

WASTE_TOLERATED	= 100
ALLOCATION_SIZE_ERROR_THRESHOLD	= 4194304
ALLOCATION_SIZE_REPORTING_THRESHOLD	= 65536
168	= 150
296	= 50
552	= 20
1064	= 10
2100	= 5000
2700	= 100
4136	= 4
8232	= 2
12500	= 2

MGR_POOLS Values

Setting	Value as Installed	Default Value	Minimum Value	Maximum Value
WASTE_TOLERATED	100	100 (%)	0	100 (%)
ALLOCATION_SIZE_ERROR_THRESHOLD	4 MB	4 MB	0 (disabled)	2 GB

ALLOCATION_SIZE_REPORTING_THRESHOLD	64 KB	64 KB	0 (disabled)	2 GB
XXXXXX	N/A	(See "MGR_POOLS Settings" on page 534)	0	32767

Performance and Usage Considerations

- Provisions should be made for pools that are the product of the MGR_CLASS section of the STARTUP member, as requests for these memory elements will automatically occur at the initiation of resolution (generally, the receipt of the ZMASTER object). Special attention should be given to the pools that will be identified by the product of the third and fourth values specified in MGR_CLASS, as in Y,N,3072,500 for the MGR_CLASS entry for SOFTWARE.FILE.
- The product of the third and fourth values of this is a value exactly equal to 1.5 MB, and one of these will be allocated at the initiation of each resolution. Additional storage is required for storage management of these queues, so a storage pool of elements size $(3072) * (500) + 500 = 1536500$ should be created. The number of elements to assign to this pool is dependent on the MGR_TASK_LIMIT values specified (for example, one element for each concurrent task), and the number of SOFTWARE.FILE instances resolved for each Sentient agent.
- When using the Distributed Configuration Server, the default setting of ALLOCATION_SIZE_ERROR_THRESHOLD must be changed to prevent a destination CSDB being only partially updated. If the Distributed Configuration Server transfers a resource file that is larger than the default (4 MB), and the setting has not been changed, the 'commit' operation will halt, leaving the destination CSDB only partially updated. See the examples below for completing this:
 - If you use memory pooling, set ALLOCATION_SIZE_ERROR_THRESHOLD to zero, as below:
`[MGR_POOLS]ALLOCATION_SIZE_ERROR_THRESHOLD = 0`
 Here, 0 will disable the memory allocation limit.
 - If the MGR_POOLS section does not exist in the `edmprof` file, there are no issues and no changes are required.

Pool Values

Size	64	400	632	2048	4120	6504	8200	12000
Number	100	100	100	60	60	30	15	20
Waste %	100	100	100	100	100	89	65	75
InUse	21	2	12	0	0	0	1	0
HighWM	21	15	12	1	1	0	1	4
Allocs	30	656	19	1	5	0	1	218
Empty	0	0	0	0	0	0	0	0
Waste	0	0	0	0	0	0	0	0

- **Size** is the size of the elements of the pool.
- **Number** is the number of elements currently in the pool.
- **Waste %** is the percentage of waste tolerated at storage element allocation time.
- **InUse** is the number of elements that were in use when the stats call was made.
- **HighWM** is the high watermark that shows the maximum number of elements that have been allocated at any one time since startup or a clear command.
- **Allocs** is the total number of elements that were allocated since startup or since the last clear command on that pool.
- **Empty** is the number of failed attempts to get an element because there were no free elements available.
- **Waste** is the number of failed attempts to get an element because the waste was higher than the tolerated waste percent.

In the example in ["Pool Values" on the previous page](#), the pools of 64, 400, 632, 2048, and 4120 do not have a specified waste tolerance. This is because these pool sizes are so small that the waste in them will not be an issue. There is a **DISABLE** command that looks like: `F jobname, DISABLE, 2048` or just: `F jobname, D, 2048`. This would disable pool 2048, that is, it would prevent new allocations from using that pool. Allocations between 633 and 4120 bytes would then come from the 4120 pool, provided the tolerated waste percentage is met. A disabled pool will show up on the command with a "D" after the size. A disabled pool can be re-enabled with the **ENABLE** command, as follows: `F jobname, E, 2048`. The pool counts can be cleared with the **CLEAR** command: `F jobname, C, 2048`.

Note: The HighWM, Allocs, Empty, and Waste counters are cleared by the above command.

Lastly, there is an **ADJUST** command, which looks like: `F jobname, A, 2048, 80, 95`. This command would add 20 free elements to the 2048 pool ($60 + 20 = 80$) and would set its waste tolerated percentage to 95. The last parameter (percentage of waste tolerated) is optional. The "A" denoting the **ADJUST** subcommand can be replaced with **ADJUST**, **ADJUS**, **ADJU**, **ADJ**, or **AD**. The **Enable**, **Disable**, and **Clear** commands can also be specified in a longer form. If you use the **ADJUST** command to set the number of elements to 0, then the pool is completely deleted if all the elements can be freed. However, the pool count will not go to 0 as long as there are allocated elements. New pools can be added on the fly with the **ADJUST** command. Finally, storage trace (**STORAGE=YES**) can be used to produce a message in the log each time a storage allocation is made and freed. This can be used to tune the pool, and to understand the Configuration Server's use of dynamic memory.

MGR_RESOLUTION_FILTERS

This section defines filtering rules for resolution, based on connection type.

Note: This section is not included in the `edmprof` file at installation—it must be manually added.

MGR_RESOLUTION_FILTERS Settings

Setting	Description
DOMAIN.CLASS	Specify the domain and class of the CSDB for which to define filtering rules for resolution.

Valid values for MGR_RESOLUTION_FILTERS are:

- RCA – Allow Client Automation resolution only.
- ANY – Allow all resolutions. (This is the default.)

Example

```
[MGR_RESOLUTION_FILTERS]
```

SOFTWARE.ZSERVICE	= RADIA
POLICY.WORKGROUP	= RADIA
POLICY.*	= RADIA

MGR_RESOLUTION_FILTERS Values

Setting	Value as Installed	Default Value	Minimum Value	Maximum Value
DOMAIN.CLASS	N/A	N/A	N/A	N/A

Performance and Usage Considerations

- CLASS can be specified as a valid CSDB class, or with a wildcard (*) to specify all classes in a domain.

MGR_RETRY

This section specifies how soon (in minutes) an Sentient agent can attempt another connection to the Configuration Server, after being rejected due to an exceeded task limit or a disabled connection.

MGR_RETRY Settings

Setting	Description
BUSY_RETRY	Number of minutes for an Sentient agent to wait before reconnecting when the Configuration Server is at its task limit (TASK_LIMIT).
DISA_RETRY	Number of minutes for an Sentient agent to wait before reconnecting when the Configuration Server has logons halted.

Example

[MGR_RETRY]

BUSY_RETRY	= 1
DISA_RETRY	= 999

Note: It is recommended that the values of at least 1 for these settings to avoid unnecessarily tying up the Configuration Server.

MGR_RETRY Values

Setting	Value as Installed	Default Value	Minimum Value	Maximum Value
BUSY_RETRY	7 minutes	0 (allow connect now)	0 (allow connect now)	999 (do not retry)
DISA_RETRY	999 (do not retry)	999 (do not retry)	0 (allow connect now)	999 (do not retry)

Performance and Usage Considerations

- You can raise these values if processing resources are critical. Lowering these values will provide greater assurance that connections will be re-established if broken during Sentient agent connects.
- The MGR_RETRY settings should be coordinated with values in the MGR_NOTIFY and MGR_TASK_LIMIT sections.

MGR_RIM

This section specifies the IP address/name and port of the Sentient Enterprise Inventory Manager (Inventory Manager) if it has been enabled.

Note: This section will be present only if the Inventory Manager option was selected during the Configuration Server installation.

MGR_RIM Settings

Setting	Description
HTTP_HOST	The IP address of the Inventory Manager. If the Inventory Manager is co-located with the Configuration Server, specify <code>localhost</code> .
HTTP_PORT	The port of the Inventory Manager. The default is 3466 .

Example

```
[MGR_RIM]
HTTP_HOST = localhost
HTTP_PORT = 3466
```

MGR_RIM Values

Setting	Value as Installed	Default Value	Minimum Value	Maximum Value
HTTP_HOST	None	N/A	N/A	N/A
HTTP_PORT	N/A	3466	N/A	N/A

MGR_RMP

This section specifies the IP address/name and port of the Sentient Enterprise Portal (Portal), if it has been enabled.

Note: This section will be present only if the Portal option was selected during the Configuration Server installation.

MGR_RMP Settings

Setting	Description
HTTP_HOST	The IP address of the Portal. If the Portal is co-located with the Configuration Server, specify <code>localhost</code> .
HTTP_PORT	The port of the Portal. The default is 3466 .

Example

```
[MGR_RMP]
HTTP_HOST = localhost
HTTP_PORT = 3466
```

MGR_RMP Values

Setting	Value as Installed	Default Value	Minimum Value	Maximum Value
HTTP_HOST	None	N/A	N/A	N/A
HTTP_PORT	N/A	3466	N/A	N/A

MGR_ROM

Specifies information used for OS management.

Caution: Do not change any of these settings unless requested to do so by your Persistent Systems Software Support representative

MGR_ROM Settings

Setting	Description
LDAP_DIRECT_ENABLED	Enable the use of OpenLDAP authentication for OS Management operations (enabled by default)
LDAP_DIRECT_HOST	Host name or IP address of a server where OpenLDAP is active
LDAP_DIRECT_PORT	Port number to use for OpenLDAP authentication
LDAP_DIRECT_UID	User ID for OpenLDAP authentication
LDAP_DIRECT_PASS	Password for OpenLDAP authentication
PORTAL_HOST	Host name or IP address for the RCA Portal
PORTAL_PORT	Port number for the RCA Portal
PORTAL_ZONE	Zone name in the RCA Portal
DISPLAYNAME	Display name used in the RCA Portal for the device
PORTAL_UID	User ID of an RCA Portal user who can update a device or the ROM object
PORTAL_PASS	Password of an RCA Portal user who can update a device or the ROM object
PORTAL_USE_SSL	Enable use of SSL to connect to the RCA Portal (handled by the UI)
FORCE_ZONE	Used for migration from an CA Classic installation to a Core and Satellites installation. This setting ensures that non-standard zone names (different from cn=hp,cn=radia) are handled without cancelling the OS installation process.

Example

```
[MGR_ROM]

LDAP_DIRECT_ENABLED=1
```

```

LDAP_DIRECT_HOST = localhost

LDAP_DIRECT_PORT = 3474

LDAP_DIRECT_UID = (contains AES encrypted user ID value)

LDAP_DIRECT_PASS = (contains AES encrypted password value)

PORTAL_HOST = localhost

PORTAL_PORT = 3466

PORTAL_ZONE = cn=hp, cn=radia

FORCE_ZONE = true

DISPLAYNAME = compname

PORTAL_UID = (contains AES encrypted user ID value)

PORTAL_PASS = (contains AES encrypted password value)

PORTAL_USE_SSL = 0

```

MGR_SMTP_MAIL

This section specifies SMTP-related parameters, including the ID of the Configuration Server and the TCP port number of the Configuration Server to be used.

MGR_SMTP_MAIL Settings

Setting	Description
DNS_SERVER	The Domain Name Service (DNS) server that is used to query the mail server address. Note: Specify a value for this setting to ensure that mail is delivered.
MAIL_DIR	Directory to spool and queue outgoing mail from the SMTP send Manager.
MAIL_TIMEOUT	Timeout interval (in seconds) for establishing communications with the mail server.
MAX_TIME_IN_SPOOL	The interval (in minutes) to wait before deleting undelivered mail in the spool. The default is 4320 minutes (3 days).
MGR_MAIL_ID	The mail ID for the Configuration Server. This setting takes the form of a fully qualified address (for example, <code>ConfigServer@foobar.com</code>) and has a maximum length of 255 bytes. The mail-receiving Configuration Server will reject mail that is addressed to any other user ID.
RETRY_INTERVAL	The interval (in seconds) to wait before re-attempting to deliver mail in the spool. The default is 300 seconds (5 minutes).
SMTP_PORT	The port on which the Configuration Server's SNMP Manager <i>send</i> and <i>receive</i> tasks (ZSMTSMGR and ZSMTRMGR, respectively) will listen for incoming mail.

Examples

Windows Example:

[MGR_SMTP_MAIL]

DNS_SERVER	= 192.168.1.20
MAIL_DIR	= D:\MGR\MAIL
MAIL_TIMEOUT	= 60
MAX_TIME_IN_SPOOL	= 4320
MGR_MAIL_ID	= ConfigServer@foobar.com
RETRY_INTERVAL	= 300
SMTP_PORT	= 25

MGR_SMTP_MAIL Values

Setting	Value as Installed	Default Value
DNS_SERVER	N/A	NONE
MAIL_DIR	as specified during installation	<i>current_directory</i>
MAIL_TIMEOUT	N/A	60
MAX_TIME_IN_SPOOL	4320 minutes (3 days)	4320 minutes (3 days)
MGR_MAIL_ID	as specified during installation	<i>sending_address</i>
RETRY_INTERVAL	300 seconds (5 minutes)	300 seconds (5 minutes)
SMTP_PORT	as specified during installation	25

Performance and Usage Considerations

- If storage capacity is an issue, decrease the MAX_TIME_IN_SPOOL value. This will decrease the length of time messages are retained.
- Increase the RETRY_INTERVAL value to use fewer system processing resources.
- DNS_SERVER: For cross-platform compatibility, operating system network settings for DNS servers are not used by the Configuration Server mail delivery. Therefore, to ensure delivery of all e-mail (including license warning messages) that are sent by the Configuration Server, DNS_SERVER must be defined.

MGR_SNMP

This section contains SNMP-related parameters that include where SNMP traps are to be sent and how to control the behavior of the built-in SNMP agent.

MGR_SNMP Settings

Setting	Description
RUN_AS_EXTENSION	If YES, the Windows SNMP service is the primary SNMP agent, and RCA SNMP transactions are processed by a RCA SNMP extension DLL. If so, SNMP_PORT and SNMP_IP_ADDR are not used because SNMP port access is handled by the Windows SNMP service. The value of SNMP_COMMUNITY will be used for insertion into traps that are issued by the Configuration Server, but will not be used to authenticate GET and SET commands. If NO, the Configuration Server will act as the primary SNMP agent, and SNMP_COMMUNITY should be specified, while SNMP_IP_ADDR and SNMP_PORT can be specified to override their defaults.
SNMP_COMMUNITY	This is a password that incoming SNMP transactions must match. It should be set to a character string. The string will be used as the SNMP community name by the agent. The default is public . Note: This keyword is effective only if RUN_AS_EXTENSION=NO.
SNMP_IP_ADDR	The TCP/IP address of the local network adapter card on which the agent is to receive SNMP transactions. The default is 0.0.0.0 , meaning any adapter on the machine can be used. Note: This keyword is effective only if RUN_AS_EXTENSION=NO and there are several adapters on the machine, and a specific adapter is to receive SNMP transactions.
SNMP_MANAGER_IP_ADDR SNMP_MANAGER_IP_ADDR2 SNMP_MANAGER_IP_ADDR3	The SNMP Managers at the IP addresses specified here are authorized to issue GET and SET commands for variables supported by the agent. The SNMP Manager specified for SNMP_MANAGER_IP_ADDR is considered the primary SNMP Manager. This field is required because it specifies two things: 1) the receiving location of traps created by the Configuration Server, and 2) the address of the authorized source for SNMP commands.
SNMP_MANAGER_PORT	This parameter is used to specify the remote TCP/IP port to which the Configuration Server sends its traps. The default is port 162 .
SNMP_PORT	This parameter is used to specify the TCP/IP port on which the agent receives SNMP transactions. The default is port 161 .
	Note: This keyword is effective only if RUN_AS_EXTENSION=NO.
SNMP_SET_COMMUNITY	This parameter can be set to a character string. The agent will use this string as the SNMP community name when it is attempting to authorize SET commands. If this keyword is not specified, the community name given by SNMP_COMMUNITY is used for SET commands.

	Note: This keyword is effective only if <code>RUN_AS_EXTENSION=NO</code> .
SNMP_ZERROR_SEVERITY	This parameter is used to specify the severity of ZERROR instances to send as SNMP traps. The trap is sent when the Configuration Server adds an error instance to its ZERRORM for an error whose severity is greater than or equal to the value specified by this parameter. The parameter can be set to a positive value between 0 and 99; the default is 12 .

Example

[MGR_SNMP]

RUN_AS_EXTENSION	= NO
SNMP_COMMUNITY	= public
SNMP_IP_ADDR	= 0.0.0.0
SNMP_PORT	= 162
SNMP_MANAGER_IP_ADDR	= 183.235.246.32
SNMP_ZERROR_SEVERITY	= 12

MGR_SNMP Values

Setting	Value as Installed	Default Value
RUN_AS_EXTENSION	YES	NO
SNMP_COMMUNITY	public	public
SNMP_IP_ADDR	0.0.0.0	0.0.0.0
SNMP_MANAGER_IP_ADDR	N/A	N/A
SNMP_MANAGER_PORT	N/A	162
SNMP_PORT	161	161
SNMP_SET_COMMUNITY	N/A	N/A
SNMP_ZERROR_SEVERITY	12	12

Performance and Usage Considerations

There are no performance or usage issues with any of the SNMP parameters. If you do not start `zsnmpmgr`, you are not starting the Sentient agent, and are running one less task in the Configuration Server.

MGR_SSL

This section specifies the operational settings for an SSL Manager. For more information on configuring and using an SSL Manager, see *Sentient Enterprise Console Online Help*.

MGR_SSL Settings

Setting	Description
CA_FILE	Sets the Certificate Authority's certificate. The CA certificate is usually stored in a file in Privacy Enhanced Mail (PEM) format. The SSL Manager needs a CA certificate to start up. If it's expired or corrupt it prevents the SSL Manager from starting up.
CERTIFICATE_FILE	Sets the Configuration Server or the server certificate. The certificate is usually stored in a file in PEM format. This value must be a valid and existing certificate file. The SSL Manager needs a certificate to start up. An expired or corrupt certificate will prevent the SSL Manager from starting up.
KEY_FILE	Sets the private key. The private key is usually stored in a file in PEM format. This value must be a valid and existing key file. The private key is usually stored in the same file as the server certificate, in which case, you don't have to specify any value for the key file.
KEY_PASSWORD	The password that is used to encrypt the private key, the one specified in the KEY_FILE keyword. This is usually needed if the private key is encoded. If the private key is not encoded, you don't need to specify this parameter. Note: This setting is not automatically added to the <code>edmpconf</code> file by the installation; it must be manually added by an Sentient administrator.
SSL_CIPHERS	Specifies the preferred OpenSSL ciphers to use for SSL connections between the Configuration Server and other Sentient components. For example: <code>ALL:ADH:RC4+RSA:+HIGH:+MEDIUM:+EXP:+eNULL</code> If you do not specify SSL_CIPHERS, the default ciphers are used: <code>ALL:ADH:RC4+RSA:+HIGH:+MEDIUM:+LOW:+SSLv2:+EXP:+eNULL</code> Refer to the OpenSSL web site additional information: http://openssl.org/docs/apps/ciphers.html
SSL_PORT	The port on which the SSL Manager will listen for Sentient agent connects.
VERIFY_CLIENT	Specifies whether the Configuration Server should verify the Sentient agent by requesting a certificate from it. If the Sentient agent doesn't have a certificate, the connection will be dropped. Note: This setting is not automatically added to the <code>edmpconf</code> file by

the installation; it must be manually added by an Sentient administrator.

Example

[MGR_SSL]

CA_FILE	= w:\openssl\ms\cacert.pem
CERTIFICATE_FILE	= w:\openssl\ms\svrcert.pem
KEY_FILE	= w:\openssl\ms\srvprvk.pem
KEY_PASSWORD	= violin
SSL_PORT	= 3456
SSL_CIPHERS	= ALL:ADH:RC4+RSA:+HIGH:+MEDIUM:+EXP:+eNULL
VERIFY_CLIENT	= Y

Performance and Usage Considerations

- In order for an SSL Manager to function, `CMD_LINE=(zsslmgr) RESTART=YES` must be specified in the `MGR_ATTACH_LIST` section.
- The settings are placeholders and, as such, are not used until an SSL add-on is installed.

MGR_STARTUP

This section specifies startup information for the Configuration Server.

Caution: After a successful installation:
DO NOT change `MANAGER_TYPE`, `MEMORY_TYPE`, `MGR_ID`, `MGR_NAME`, or `TASK_TYPE`, unless advised to do so by Persistent Systems Technical Support.
DO NOT change or delete `MGR_UUID` unless instructed to do so by Persistent Systems Technical Support.

MGR_STARTUP Settings

Setting	Description
ALLOW_DUPPLICATE_IP_ADDRESS	NO will cause the Configuration Server to reject a second log on if one from the same IP address is already active. YES will allow multiple concurrent IP connections from the same Sentient agent IP address.
BYTE_LEVEL_DIFF	Turns on byte-level differencing during resolution.

MANAGER_ TYPE	Type of Configuration Server; valid values are DISTRIBUTED (the default), SERVER, and STANDALONE. For more information on this setting, see "MANAGER_TYPE Values" on page 550. Note: If the Configuration Server is going to participate in Distributed Configuration Server operations, its type must be DISTRIBUTED or SERVER. Important Note: Do not alter this setting unless instructed to do so by Persistent Systems Technical Support.
MEMORY_ TYPE	Reserved for future development. Important Note: Do not alter this setting unless instructed to do so by Persistent Systems Technical Support.
MGR_ID	A three-byte, hexadecimal identifier for the Configuration Server log file. This ID is used in the CSDB to identify Configuration Servers in a Distributed Configuration Server environment, and is passed to the Sentient agent as the ZOBJMID variable. Valid values are any combination of 001 to EFF. Important Note: Do not alter this setting unless instructed to do so by Persistent Systems Technical Support.
MGR_NAME	Configuration Server name. Important Note: Do not alter this setting unless instructed to do so by Persistent Systems Technical Support.
MGR_UUID	The Universal Unique Identification Number (UUID) of the Configuration Server. This 32-byte ID is created automatically during the installation and is used exclusively for licensing. Important Note: Do not change or delete this value.
OBJECTID_ FORMAT	This setting determines which format will be used for generating object IDs. Specify 1 to use the established algorithm. Specify 2 to use the new algorithm. If the value is absent or invalid, the default of 1 is assumed and the established algorithm is used. Note: For more information on this setting, see the section, "OBJECTID_FORMAT" on page 550.
SHOW_ VERINFO	Displays version information in the Configuration Server log at startup.
TASK_TYPE	Reserved for future development. Important Note: Do not alter this setting unless instructed to do so by Persistent Systems Technical Support.
TCP_PORT	Port on which to listen for TCP/IP Sentient agent connects. The default is 3464. Note: This setting will be overridden by specifying a TCP/IP port in the MGR_ATTACH_LIST section. See "MGR_ATTACH_LIST" on page 505 for more details.
VERBOSE	Sends messages to the screen (stderr) when the Configuration Server starts

	up, verifies the license, verifies the CSDB, loads cache, and readies IP Managers (TCP and SSL) for processing and when shutting down.

Example

[MGR_STARTUP]

BYTE_LEVEL_DIFF	= NO
MANAGER_TYPE	= DISTRIBUTED
MGR_ID	= 001
MGR_NAME	= RCS
MGR_UUID	= ssed111454d6kgh4eh7g3md90f94d6k8
OBJECTID_FORMAT	= 2
SHOW_VERINFO	= YES
TASK_TYPE	= THREAD
TCP_PORT	= 3464
VERBOSE	= NO

MGR_STARTUP Values

Setting	Value as Installed	Default Value	Minimum Value	Maximum Value
ALLOW_DUPLICATE_IP_ADDRESS	YES	NO	N/A	N/A
BYTE_LEVEL_DIFF	N/A	NO	N/A	N/A
MANAGER_TYPE	DISTRIBUTED	DISTRIBUTED	N/A	N/A
MEMORY_TYPE	SHARED	SHARED	N/A	N/A
MGR_ID	As specified	001	000	EFF (F00-FFF are reserved)
MGR_NAME	As specified	EDM	N/A	N/A
MGR_UUID	N/A	N/A	N/A	N/A
OBJECTID_FORMAT	2	1	N/A	N/A
SHOW_	N/A	YES	N/A	N/A

VERINFO				
TASK_TYPE	THREAD	THREAD	N/A	N/A
TCP_PORT	3464	1029	N/A	N/A
VERBOSE	N/A	NO	N/A	N/A

Performance and Usage Considerations

- If `MANAGER_TYPE` is set to `STANDALONE`, the Configuration Server will not be eligible for Distributed Configuration Server functionality. (See the section, "[MANAGER_TYPE Values](#)" below.)
- `MGR_ID` must be a three-character, alphanumeric, hexadecimal string. (Valid values are 001 to EFF.)
- If running the Configuration Server as a Windows Service, `VERBOSE` will be disabled.

OBJECTID_FORMAT

Note: A CSDB object's ID is a unique, 12-character, hexadecimal identifier that is automatically generated and assigned when a CSDB object is created.

The time-based format of object ID generation eliminates the possibility of randomly generating a duplicate object ID because each object ID begins with the letter A, B, C, E, or F—differing from the old format, which used only the letter D as a prefix.

The format of CSDB object IDs is *AmmmXXXXXXXX*, where:

A = the new prefix letter (A, B, C, E, or F)

mmm = the Configuration Server ID (see `MGR_ID` in "[MGR_STARTUP Settings](#)" on page 547.)

XXXXXXXX = the unique value in hexadecimal format

Note: Specifying the object ID generation format affects new object ID generation only; it has no effect on existing object IDs.

When the time-based generator counter wraps, this format will automatically start prefixing the object IDs with the next sequential letter.

MANAGER_TYPE Values

The values for the `MANAGER_TYPE` settings are described in more detail below.

- **DISTRIBUTED:** The CSDB is read- and write-enabled and can be updated by various means; this CSDB can participate in RCA-DCS operations in which it can function as Source, Destination, or both.
This is the `MANAGER_TYPE` **external default**.

- **SERVER:** The CSDB is read-only; it cannot be updated; to run, it requires that OBJECTID_FORMAT=2 and object ID cache be disabled; this CSDB can participate in RCA-DCS operations, but only as Destination or middle-tier.
- **STANDALONE:** The CSDB is read- and write-enabled for the Administrator and Configuration Server methods from Sentient agent connects; this CSDB cannot participate in DCS operations.
- This is the **MANAGER_TYPE internal default**.

Note: If the value of the MANAGER_TYPE setting is anything other than DISTRIBUTED or SERVER (including no value and invalid values), it will assume the MANAGER_TYPE internal default of STANDALONE.

MANAGER_TYPE=SERVER

If MANAGER_TYPE=SERVER then, by default, two subsequent conditions will exist:

- the OBJECTID_FORMAT=2 algorithm will be used for object ID generation, and
- object ID caching will be disabled.

MGR_TASK_LIMIT

An Sentient administrator can use this section of the edmpof file to specify the various settings that are related to Configuration Server tasks.

MGR_TASK_LIMIT Settings

Setting	Description
TASK_HEAP_SIZE	This setting controls the heap size of a task (used in conjunction with MGR_POOLS). Note: This setting is not applicable to the current release of the Configuration Server.
TASKLIM	The maximum number of Sentient agent tasks allowed to concurrently run on the Configuration Server.
TASK_LOG_LIM	The maximum number of lines, per Sentient agent, that can be written to the Configuration Server log.
TASK_RESO_LIM	The maximum number of resolutions allowed per Sentient agent.
TASK_STACK_SIZE	This setting defines how many variables can be used per task.

Example

```
[MGR_TASK_LIMIT]
```

TASK_HEAP_SIZE	= 0
TASKLIM	= 20
TASK_LOG_LIM	= 0
TASK_RESO_LIM	= 64000
TASK_STACK_SIZE	= 64000

MGR_TASK_LIMIT Values

Setting	Value as Installed	Default Value	Minimum Value	Maximum Value
TASK_HEAP_SIZE	0	0	0	4 GB
TASKLIM	50	0	0	32 KB
TASK_LOG_LIM	0	100000 Lines	0	N/A
TASK_RESO_LIM	64000	64000 Resolutions	1	64000
TASK_STACK_SIZE	64000	64000	16384	64000

Caution: Do not change the TASK_STACK_SIZE setting in this section unless advised to do so by a member of Persistent Systems Technical Support.

Performance and Usage Considerations

- The MGR_TASK_LIMIT settings should be coordinated with values in the MGR_NOTIFY and MGR_RETRY sections.
- TASK_STACK_SIZE will affect the depth of resolution. Higher values will result in deeper resolution.

MGR_TIMEOUT

This section specifies how long the Configuration Server will wait for a request from a connected Sentient agent before disconnecting that Sentient agent due to inactivity (no requests/responses from the Sentient agent).

MGR_TIMEOUT Settings

Setting	Description
ADMIN_TIMEOUT	Timeout (in seconds) for administrator functions.
SEND_THROTTLE	Timeout (in milliseconds) before each send.
TIMEOUT_COMM	Communications (receive) timeout (in seconds).

Example

```
[MGR_TIMEOUT]
ADMIN_TIMEOUT = 0
SEND_THROTTLE = 0
TIMEOUT_COMM = 1800
```

MGR_TIMEOUT Values

Setting	Value as Installed	Default Value	Minimum Value	Maximum Value
ADMIN_TIMEOUT	0 (never time out)	0 (never time out)	0	32767
SEND_THROTTLE	N/A	0	0	4 GB
TIMEOUT_COMM	1800 (seconds)	0 (never time out)	0	32767

Performance and Usage Considerations

- If processing resources are critical, increase these MGR_TIMEOUT values.
- The MGR_TIMEOUT settings should be coordinated with values in the MGR_RETRY section.
- The SEND_THROTTLE value can be overridden by bandwidth throttling variables sent up in an Sentient agent object.
- TIMEOUT_COMM is the Configuration Server analog to the Sentient agent ZTIMEO. If a connect is active and the Configuration Server has not received any data from a Sentient agent for the TIMEOUT_COMM value, the Configuration Server will terminate the session.

MGR_TPINIT

This section specifies packet sizes to send to Sentient agents.

MGR_TPINIT Settings

Setting	Description
BUFTCP	TCP buffer size used for send/receive.
GET_REMOTE_HOST_NAME	Controls the Configuration Server's attempt to obtain the <i>host_name</i> from the DNS server. The default is NO . Note: Do not set to YES if the Configuration Server is not in a dynamic TCP/IP environment (such as DNS and DHCP).
MAXREC	Maximum record size.

Caution: *Do not* change any settings in this section unless advised to do so by a member of Persistent Systems Technical Support.

Example

[MGR_TPINIT]

BUFTCP	= 12288
GET_REMOTE_HOST_NAME	= NO
MAXREC	= 6144

MGR_TPINIT Values

Setting	Value as Installed	Default Value
BUFTCP	12288	12288
GET_REMOTE_HOST_NAME	NO	NO
MAXREC	6144	6144

Performance and Usage Considerations

- The buffer size reflected in the MGR_TPINIT settings should be the same for the Configuration Server and Sentient agents.
- Any buffer size setting in the MGR_TPINIT section should only be changed in coordination with equivalent changes to Sentient agents and after having been directed to do so by a member of Persistent Systems Technical Support.
- Do not use GET_REMOTE_HOST_NAME if the Configuration Server is not in a dynamic TCP/IP environment. This will cause the Configuration Server to expend unnecessary processing time attempting to associate the remote host name.
If GET_REMOTE_HOST_NAME=YES, the Configuration Server obtains the remote host name using standard library calls. The IPNAME is received on the Configuration Server (via DNS) and stored in ZMASTER.ZIPNAME, which is stored in the appropriate domain of the PROFILE File.
The remote host name will appear in each associated line of the Configuration Server log in place of the IP address.

MGR_TRACE

This section contains a list of keywords (settings) that you can specify to control and influence diagnostic logging for the Configuration Server. All diagnostic output produced by TRACE settings is written to the active Configuration Server log. To enable a TRACE keyword, specify YES. To de-activate a TRACE keyword, specify NO.

TRACE keywords specified in this section are invoked at Configuration Server initialization, and remain in effect:

- until they are changed by altering the MGR_TRACE setting and restarting the Configuration Server.
- while the Configuration Server is running, using the Console.
- until a specified REXX overrides the setting.
- unless a ZCVT value overrides the setting.

The trace settings that are in effect at Configuration Server initialization are displayed at the beginning of the log.

Note: The value for each setting is evaluated in the order in which it is presented in the `edmpprof` file. The results can be non-intuitive. For example:

- If ALL=YES is the first setting specified, and each following settings are specified as NO, the effect is to turn off *tracing*.
- If ALL=YES is the last setting specified, all tracing will be active.
- If ALL=NO is the last setting, all tracing is turned off.

MGR_TRACE Settings

Setting	Description
ADMIN	Traces ADMIN transaction flow.
ADMPROM	Not used.
ALL	Turns on all other traces.
ALLOC	Traces file allocations.
AUDIT	Traces audit file activity.
BUFF	Traces data buffers (without transformation).
CMPR	Traces data compression.
COMM	Traces data stream buffers.
COMMCBS	Traces communications control block (CCB) activity.
COMMDATA	Traces data communications.
COMMRPLS	Traces communications control blocks (CCBS).
CONFIG	Traces configuration file activities.
DATA	Traces data buffers to or from the Sentient agent.
DES	This setting is no longer used.
DMA	Traces Distributed Configuration Server activity.

ENQDEQ	Traces serialization activity (enqueues/dequeues).
EXPL	Traces data transformation (explode).
FILE	Traces file I/O.
IMPL	Traces data transformation (implode).
LOOKASID	Traces cache activity for classes/instances.
METHOD	Traces Configuration Server method execution/return codes.
NOTIFY	Traces notify processing.
OBJCRC	Traces object CRC processing.
OBJRES	Traces object resolution (very detailed).
OBJRES1	Traces object resolution (medium detail).
OBJRESO	Traces high-level object resolution flow (light detail).
OBJXFER	Traces object transfer.
PASSWORD	Traces passwords.
POOLMISS	Traces memory pool allocation.
PROFILE	Traces profile database activity.
PROMOTE	Traces file promotion.
RESOURCE	Traces resource file activity.
REXX	Traces REXX environment.
REXXOFF	Suppresses all REXX activity.
STORAGE	Traces storage in conjunction with the MGR_LOG's STORAGE_INTERVAL setting.
STATS	Traces statistics.
SUBST	Traces variable substitution.
TCP	Traces TCP/IP activity.
TEST	Reserved.
VAR	Traces the variable references.
VARSTG	Traces variable processing storage usage.
VARSUB	Traces variable substitution activity.
YEAR2000	Traces a database's Year-2000 compliance.

Example

[MGR_TRACE]

ADMIN	= NO
ADMPROM	= NO
ALLOC	= NO
AUDIT	= NO
BUFF	= NO
CMPR	= NO
COMM	= NO
COMMCBS	= NO
COMMDATA	= NO
COMMRPLS	= NO
CONFIG	= NO
DATA	= NO
DMA	= NO
ENQDEQ	= NO
EXPL	= NO
FILE	= NO
IMPL	= NO
LOOKASID	= NO
METHOD	= NO
NOTIFY	= NO
OBJCRC	= NO
OBJRES	= NO
OBJRES0	= NO
OBJRES1	= NO
OBJXFER	= NO
PASSWORD	= NO

POOLMISS	= NO
PROFILE	= NO
PROMOTE	= NO
RESOURCE	= NO
REXX	= NO
REXXOFF	= NO
STATS	= YES
STORAGE	= NO
SUBST	= NO
TCP	= NO
TEST	= NO
VAR	= NO
VARSTG	= NO
YEAR2000	= NO
ALL	= YES

Performance and Usage Considerations

- Turning on trace flags creates a large number of Configuration Server log messages. This will degrade the performance of the Configuration Server due to the disk I/O load. However, this might be necessary at times for problem resolution. Ensure that the Configuration Server log is properly configured.
- Tracing can be clustered to troubleshoot a particular aspect of Configuration Server operations, while simultaneously preserving an appropriate level of logging activity. For example, turning on flags, such as OBJCRC, OBJRES, OBJRES1, OBJRES0, and OBJXFER, can help identify problems that might be occurring during object resolution. Likewise, CPIC, DATA, and TCP focus on communications activities. (Note that some of these traces pertain to specific protocols.) Special purpose trace flags include DMA, NOTIFY, REXX, and METHOD.
- STORAGE=YES can be used to produce a message in the log each time a storage allocation is made and freed. This can be used to tune the pool and to understand the Configuration Server's use of dynamic memory.
- ALL=YES does not affect the REXXOFF trace settings.

MGR_USERLOG

This section specifies the logging directory and logging options for the user logging facility.

MGR_USERLOG Settings

Setting	Description
ACTIVATE	Activate user log at Configuration Server startup. Values are YES and NO.
DIRECTORY	Fully qualified directory path where the user log is written.
FLUSH_SIZE	The number of bytes between automatic flushes of operating system buffers for Configuration Server log file.
MESSAGE_WIDTH	The maximum width (in bytes) of the messages in the user log.
PIPE_SIZE	The maximum amount (in bytes) of log messages that can be queued by the Configuration Server logging facility while the log file is busy. When this value is reached, any task that issues a log message will freeze until the pipe starts emptying.
THRESHOLD THRESHHOLD (both spellings accepted)	Maximum number of messages that will be written to a log before automatically switching to the next log. When the limit is reached, new log files are created. Specify a negative number to overwrite the log file when the limit is reached.

Windows Example

[MGR_USERLOG]

ACTIVATE	= NO
DIRECTORY	= C:\Program Files\PSL\RCA\ConfigurationServer\log
FLUSH_SIZE	= 256
MESSAGE_WIDTH	= 256
PIPE_SIZE	= 1000000
THRESHOLD	= 5000000

MGR_USERLOG Values

Setting	Value as Installed	Default Value	Minimum Value	Maximum Value
ACTIVATE	NO	NO	N/A	N/A
DIRECTORY	/edmmgr/log	<i>current_directory</i>	N/A	N/A
FLUSH_SIZE	256 bytes	100000 bytes	1	N/A
MESSAGE_WIDTH	256 bytes	90	80	N/A
PIPE_SIZE	1000000 bytes	65535 bytes	1	N/A

THRESHHOLD	-5000000 bytes	5000 bytes	1	N/A

Performance and Usage Considerations

- Increasing the FLUSH_SIZE will enhance performance, but will delay messages flushed to the log file.
- Increase MESSAGE_WIDTH if log messages are being truncated.
- When modifying parameters in this section as they relate to memory or disk use, be sure that the maximum amount of memory or storage space is available.

User Log Naming Conventions

The user log names conform to the following operating system specific conventions.

Note: ISO is the International Standards Organization.
Configuration Server ID is the value of the `edmprof` file's setting `MGR_STARTUP.MGR_NAME`.

Windows

- The *standard* log naming format is the user log prefix (`nvd`), followed the user designation (`ur`), and the Configuration Server ID:
`nvdur001.log`
- The *log switch* format is similar to the standard but an **`us`** (rather than `ur`) is added to the prefix, and the ISO-formatted date and time (each preceded by an underscore) appended:
`nvdus001_20050427_083357.log`
- The *log wrap* (dump) format is similar to the standard but sees the `r` being replaced by a `d`:
`nvdud001.log`

OBJECT_SIZES

Note: The `OBJECT_SIZES` section must be manually added to the `edmprof` file. Additionally, this section must be added to the `edmprof` file in order for the Configuration Server self-tuning tool to operate properly.

This section accommodates specifying the number of heaps and the heap size for CSDB objects that are being created on the Configuration Server as in-storage CSDB objects.

- These values affect only the Configuration Server self-tuning tool; they have no effect on other Configuration Server processing.
- The format for specifying these values is:
`OBJECT_NAME = heap_size,number_of_heaps`

For example:

```
ZCONTROL = 512,100000
```

In this example, whenever an object named ZCONTROL is created in the Configuration Server as an in-storage object, its initial allocation will be for 100,000 heaps of 512 bytes each.

```
ZERROR = 400,1000
```

In this example, whenever an object named ZERROR is created in the Configuration Server as an in-storage object, its initial allocation will be for 1000 heaps of 400 bytes each.

Note: For a detailed description of how these settings affect the Configuration Server self-tuning tool, see ["Configuration Server Self-Tuning Tool" on page 569](#).

Example

```
[OBJECT_SIZES]

FILE = 1536,2000
RELEASE = 420,1536
WBEMAUDT = 6000,100
MSIFEATS = 512,100
ZOBJSTAT = 512,270
PRODUCT = 1024,200
ZREQDATA = 200,100
ASERVICE = 3000,60
ZSERVICE = 4000,60
ZREQNEWI = 150,15
MSIPROPS = 1024,30
ZERROR = 512,10
UMFLTCRI = 1536,23
WBEM = 1024,20
APPEVENT = 1024,5
SAPSTATS = 1024,10
SAP = 1024,10
CLISTATS = 512,5
UMFLTRUL = 600,4
DESKTOP = 2000,3
MSI = 2000,5
REGISTRY = 1536,5
PATH = 1024,3
ZCONFIG = 4000,1
ZMASTER = 3000,1
TIMER = 1536,2
```

RCS_TUNING_CONTROL

Note: The RCS_TUNING_CONTROL section must be manually added to the `edmprof` file. Additionally, this section must be added to the `edmprof` file in order for the Configuration Server self-tuning tool to operate properly.

This section provides a mechanism to override the default values that are specified in the Configuration Server self-tuning tool and these are described in the following table. (For a look at the Configuration Server self-tuning tool, see ["Configuration Server Self-Tuning Tool" on page 569.](#))

RCS_TUNING_CONTROL Settings

Setting	Description
MAXIMUM_MEG_CACHE	A four-digit integer value that specifies the maximum number of MBs that are to be allocated to CONTENT CACHE. Note: This setting provides protection when the size of a class (or classes) would exceed the virtual storage of the process space. In the case that one class exceeds this value, that class is forced to be cached on first reference (=Y,N).
MINIMUM_MEG_CACHE	A four-digit integer value that specifies the minimum number of MBs that are to be allocated for CONTENT CACHE.
MAXIMUM_SHARED_MEMORY_SEGMENTS	A two-digit integer value that specifies the <i>maximum_number_of_segments</i> times the <i>largest_shared_memory_size</i>. Note: This value must be less than 15 because the value cannot exceed the size of the process space (generally, 2 GB [2*1024*1024*1024]).
MAXIMUM_SHARED_MEMORY_SIZE	A four-digit integer value that specifies the maximum number of MBs that are to be allocated to any single CONTENT CACHE segment. Note: If the aggregate size of CONTENT CACHE exceeds this value, it should be allocated as multiple segments—each less than this value in size.
MAXIMUM_CLASS_INSTANCES	A ten-digit integer value that specifies the maximum number of instance elements that can exist in a class that might be cached at startup. Note: Any class with more instances will be marked (=Y,N) as “cache on first reference.”
MINIMUM_INSTANCES	A five-digit integer value that specifies the minimum number of instance elements for which ICACHE and CONTENT CACHE are intended to be sized.
UPDATE_RCS_STARTUP	A YES-NO toggle for updating the startup of the Configuration Server. The default (NO) leaves the <code>edmprof</code> file unchanged and creates the file <code>EDMPROF_TUNED.DAT_</code>.

Example

```
[RCS_TUNING_CONTROL]

MAXIMUM_MEG_CACHE = 1500
MINIMUM_MEG_CACHE = 50
MAXIMUM_SHARED_MEMORY_SEGMENTS = 6
MAXIMUM_SHARED_MEMORY_SIZE = 32
MAXIMUM_CLASS_INSTANCES = 100000
MINIMUM_INSTANCES = 10000
UPDATE_RCS_STARTUP = YES
```

RCS_TUNING_CONTROL Values

Setting	Value as Installed	Default Value	Minimum Value	Maximum Value
MAXIMUM_MEG_CACHE	None	1500	None	None
MINIMUM_MEG_CACHE	None	200	None	None
MAXIMUM_SHARED_MEMORY_SEGMENTS	None	6	None	None
MAXIMUM_SHARED_MEMORY_SIZE	None	384	None	None
MAXIMUM_CLASS_INSTANCES	None	100000	None	None
MINIMUM_INSTANCES	None	50000	None	None
UPDATE_RCS_STARTUP	None	NO	None	None

SECTION_DELIMITERS

This section is not a true section of the Configuration Server `edmprof` file. It is used only to specify the symbols that will be used to delimit section names in the `edmprof` file.

Caution: If used, SECTION_DELIMITERS must be the very first entry and the first non-blank line in the `edmprof` file. If it is not, the Configuration Server will not be able to read-in the license string and will not start up.

SECTION_DELIMITERS Settings

Setting	Description

SECTION_DELIMITERS	The character set that is to be used to enclose the section headings of the <code>edmp_{prof}</code> file. The format is SECTION_DELIMITERS = <i>xy</i> , where <i>x</i> and <i>y</i> are the left and right delimiters. The options are: <code>[]</code> , <code>()</code> , <code><></code> , and <code>{}</code> .

Examples

```
SECTION_DELIMITERS = <>

<MGR_LICENSE>LICENSE_STRING = FFCDAB

SECTION_DELIMITERS = []

[MGR_LICENSE]LICENSE_STRING = FFCDAB
```

SECTION_DELIMITERS Values

Setting	Value as Installed	Default Value
SECTION_DELIMITERS	N/A	[]

Managing Configuration Server Processing

This section describes about managing the configuration server processing.

Configuration Server Operations

Configuration Server operations has three basic phases:

- Startup
- Processing Requests
- Shutdown

Configuration Server startup is initiated by icon, command line, console, or control panel, depending on the platform and installation. In the Startup phase, the Configuration Server initializes ZTOPTASK. Using the Configuration Server `edmpprof` file to provide the working parameters for configuring the Configuration Server, ZTOPTASK then starts the Task Manager. After the various tasks specified in MGR_ATTACH_LIST are enabled, the Configuration Server is ready to process requests.

A request can be sent to the Configuration Server as a system command, an Sentient agent connect, an administrative transaction, or a Distributed Configuration Server command. During the Processing Requests phase, the Configuration Server performs the requests (tasks) that are submitted to it. There are four types of tasks.

- **System**
These tasks pertain to Configuration Server functions.
- **Client**
These tasks pertain to Sentient agent requests.
- **Admin**
These tasks pertain to Sentient CSDB Editor and Publisher operations.

- **Distributed Configuration Server**

These tasks pertain to Distributed Configuration Server functions.

The type of task is shown in each line of the Configuration Server log, as shown in the following example.

```
NVD1069I 13:49:52 [208.244.225.166 /16B] Radia ClientMax size of local  
memory allocated : 230805
```

```
NVD8115I 13:50:30 [ztoptask/17B] System TaskREXX Method  
<D:\DEV\MGR\REXX\ZSHUTDWN> with parms <%s> started at <13:50:30:574>
```

Like startup, Configuration Server shutdown can be initiated in a number of ways. In the Shutdown phase, the Configuration Server performs some basic housekeeping, then essentially reverses the startup flow. The Configuration Server is now down, enabling you to run a backup, use the database utilities, apply maintenance, or perform other operating system tasks.

Customizing Configuration Server Processing

The values in your Configuration Server `edmpconf` file enable you to customize its overall configuration. There are two main ways of customizing the flow of Configuration Server processing:

- Configuration Server REXX programs and
- Configuration Server methods.

The primary difference is that REXX programs are preconfigured, while the methods can be inserted anywhere in the processing flow.

This chapter describes the Configuration Server REXX programs and Configuration Server methods. An overview of the methods is presented in ["Overview" on page 593](#). Also, ["Configuration Server Methods" on page 593](#) details each method, providing an example of its use, a description, its associated parameters, and possible return codes.

Configuration Server REXX Programs

REXX Directories

The Configuration Server installation creates two REXX-related directories,

- `<InstallDir>\ConfigurationServer\rexx`
- `<InstallDir>\ConfigurationServer\rexx\NOVADIGM`

The `rexx` directory is empty, and the `rexx/NOVADIGM` (`rexx\NOVADIGM`) directory contains all the Sentient-related REXX programs.

Note: The `ConfigurationServer/rexx` (`ConfigurationServer\rexx`) directory can be renamed to further distinguish it from the RCA REXX directory.

Customizing the Sentient REXX Programs

The RCA REXX programs can be customized to adapt to, and enhance, various computing environments. To customize any of these REXX programs, copy them from the `rexx\NOVADIGM` (`rexx/NOVADIGM`) directory to the `rexx` directory, then modify them as needed.

Caution: Do not make any changes to the REXX programs in the `rexx\NOVADIGM` (`rexx/NOVADIGM`) directory. Doing so will adversely affect the performance of the CSDB.

There are two reasons that RCA REXX programs have to be copied to the `rexx` directory before being modified:

- If a REXX is customized and left in the `\rexx\NOVADIGM` directory, the customizations could be lost (overwritten) if a database update is applied, thereby affecting the behavior and execution of the database operations.
- During processing, the database reads the `\rexx` directory first. Therefore, place any customized REXXs in that directory.

Event Points

There are eight event points at which the Configuration Server issues calls to ten major REXX programs. The table below lists these REXX programs and the points at which they are called.

Configuration Server REXX Programs

REXX Name	When Called (Event Point)
ZSTARTUP	Configuration Server Startup
ZPCACHE	Configuration Server Startup
ZINIT	Configuration Server Startup
ZTASKSTA	Task Start
ZTASKEND	Task End
ZNFYxSTA	Notify Start
ZNFYxEND	Notify End
ZLOGSWCH	Log Switch
ZLOGWRAP	Log Wrap
ZSHUTDOWN	Configuration Server Shutdown

REXX Programs

ZSTARTUP

This is called just before the Configuration Server is enabled to accept and process Sentient agent connections. ZSTARTUP does not pass any parameters to the REXX, nor does it accept any information from this REXX. It can be used to perform user-defined specific processing before enabling connections to be initiated.

ZPCACHE

This can be called after cache is loaded during Configuration Server startup.

ZINIT

This is called during Configuration Server startup. It runs REXXs and tests for certain conditions. If the conditions are not met (return code = 16), Configuration Server startup will be halted.

Caution: This is not configurable. Consult Persistent Systems Technical Support before using this REXX.

ZTASKSTA

This is called when each connection is first accepted. It is passed a single parameter that contains the protocol level identifier for the Sentient agent.

ZTASKEND

This is called when each connection has ended, while storage and objects associated with the connection are still available. It is passed a single parameter that contains multiple subparameters that can be parsed and that are position dependent.

- Connect termination reason.
- User ID of the current connection.
- Total number of object instances, of any type, processed during this connection.
- Total number of object instances, of any type, resulting from resolution.
- The maximum depth of transient objects processed in any single instance resolution.
- Count of communications protocol sends and receives originating from this connection.
- Total number of bytes transmitted from the Configuration Server to the Sentient agent during this connection (non-compressed count).
- Total number of files transferred from the Configuration Server to the Sentient agent during this connection.

ZNFYxSTA

This is called at the beginning of Notify processing for each Sentient agent that is being notified. The x defines the type of Notify (T for TCP/IP and D for Dial-up). The set of parameters passed includes:

- UINF=<value1>
user info passed via EDMMPUSH method. If no information is provided, COMMON will be set as a value.
- RETRS=<value2>
number of retries for this destination.
- STATUS=<value3>
RETRY, SUCCESS, FAILURE.
- MSG=<value4>
message describing the result of notification.

ZNFYxSTA can generate a return code that controls the execution of the Notify, RC value 1956 (RC_SKIP_NOTIFY). If this is set, it will cause the Notify request to be written for retry without execution of current notification.

ZNFYxEND

This is called at the end of Notify processing for each Sentient agent being notified. The x defines the type of Notify (T for TCP/IP and D for Dial-up). The set of parameters passed includes:

UINF=<value1>
user info passed via EDMMPUSH method. If no information is provided, COMMON will be set as a value.

RETRS=<value2>
number of retries for this destination.

STATUS=<value3>
RETRY, SUCCESS, FAILURE.

MSG=<value4>
message describing the result of notification.

ZNFYxEND can generate a return code that controls the execution of the Notify, RC value 1955 (RC_NEVER_RETRY). If this is set, it will prevent the Notify request from being rescheduled for retry.

ZLOGSWCH

This can be called when log switch occurs (a new log file is created) to insert a user-defined command (for example, zip the old log file and save it).

ZLOGWRAP

This can be called when log wrap occurs (the log file is reused) to insert a user-defined command (for example, zip the old log file and save it).

ZSHUTDOWN

This is called just before the Configuration Server shuts down.

Configuration Server Self-Tuning Tool

The Configuration Server self-tuning tool enables Configuration Server instances to automate the tuning of the `edmpconf` file's `MGR_CACHE` and `MGR_CLASS` sections.

Note: Two `edmpconf` file sections, "`OBJECT_SIZES`" on page 560 and "`RCS_TUNING_CONTROL`" on page 561, are needed in order for this tool to function properly.

This tool does not dynamically alter the `MGR_CACHE` and `MGR_CLASS` sections while the Configuration Server is active; rather, while the Configuration Server is shutdown, it examines the database and creates an updated dataset that can be compared to the existing (master) `edmpconf` file, and possibly used to automatically replace it.

The self-tuning tool can be configured to rename the master `edmpconf` file and replace it with the newly created one, while this is not the default behavior (which is to overwrite the master `edmpconf` file). If the non-default (rename-and-replace) option is selected, the only active `edmpconf` file settings that should differ between the two `edmpconf` files are those in the `MGR_CACHE` and `MGR_CLASS` sections (because these settings are generated at shutdown and are based on the size of the database and the number of instances).

The MGR_CLASS Section

If the master `edmpconf` file does not have a `MGR_CLASS` section, the self-tuning tool attempts to content cache all the database instances. It does this by calculating the amount of space that will be necessary to support content caching all the instances, and by generating the caching directives for each class that is to be set for caching in its entirety at Configuration Server initiation and/or cache refresh.

If the `MGR_CLASS` section in the master `edmpconf` file has caching directives for individual classes, they will be preserved. However, if these caching directives are intended for internally created objects, or are received by the Configuration Server from a session partner, they will not be preserved in the automatically generated values because the new `MGR_CLASS` section is based on the CSDB.

OBJECT_SIZES

The `OBJECT_SIZES` section will be preserved (as it was in the master `edmpconf` file) in the new `edmpconf` file, and will result in additional `MGR_CLASS` section entries, in the form of:

```
"NONDB." OBJECT_NAME = N,N,heap_size,number_of_heaps
```

These entries will be listed first in the updated `MGR_CLASS` section.

They are followed by the MGR_CLASS entries from the master `edmpprof` file, which are listed based on the Sentient Enterprise processing preference in which priority is placed on processing the ZSERVICE and PACKAGE instances to facilitate the initial catalog resolution, followed by component instances, and then non-component instances.

Revise and Overwrite

The default behavior for the Configuration Server self-tuning tool is to create a new `edmpprof` file in the directory that contains the master (or original) `edmpprof` file. That format for naming the file is:

```
EDMPROF_TUNED.DAT_REVISED_YYYYMMDD_HH_MM_SS_uuuuuu
```

Where:

- `EDMPROF_TUNED.DAT_REVISED_` is a literal value,
- `YYYYMMDD` is the year, month, and date of the revise action, and
- `HH_MM_SS_uuuuuu` is the local hour, minute, second, and microsecond at which the new file is established.

Only one file matching the filter `EDMPROF_TUNED.DAT_REVISED_*` will be retained in the directory—that file will be the most recent one that was created by this tool.

Rename and Replace

If the non-default behavior of replacing the master `edmpprof` file with the newly created one is selected, the `edmpprof` file that existed when the Configuration Server was started will be renamed to:

```
EDMPROF.DAT_REPLACED_YYYYMMDD_HH_MM_SS_uuuuuu
```

Where:

- `YYYYMMDD` is the year, month, and date of the replace action, and
- `HH_MM_SS_uuuuuu` is the local hour, minute, second, and microsecond at which the file is replaced.

Up to 15 files matching the filter `EDMPROF.DAT_REPLACED_*` will be retained in the directory that contains the `edmpprof` file.

Note: The 15 files are the 14 most recent datasets, and the oldest `edmpprof` file (so that an original `edmpprof` file is retained for reference).

All of these files will be in the directory in which the original `edmpprof` file was found. Therefore, the account under which the Configuration Server is executing must have update and create authority for that directory.

The MGR_CACHE Section

In addition to the updated MGR_CLASS section, a new MGR_CACHE section gets created and replaces the section in the input `edmpprof` file.

Note: While processing the original input `edmpprof` file, the existing `MGR_CLASS` and `MGR_CACHE` sections are ignored, while all other sections are copied in their entirety. The newly created `MGR_CACHE` and `MGR_CLASS` sections will be appended to the end of the copied input `edmpprof` file.

If, in the original `edmpprof` file, there are comments preceding the `MGR_CLASS` and/or `MGR_CACHE` sections, then, in the newly created `edmpprof` file, the comments might appear to be dislocated. However, since these are comments, their position does not effect the Configuration Server startup.

Reporting Files

A reporting file (`DB_EDMPPROF_REPORT.TXT`) will be created in the Configuration Server log directory. It documents the `MGR_CACHE` and `MGR_CLASS` settings and the fact that the newly created values represent the actual amount of required storage. Only the most recent copy of each file will be retained, and each execution of the self-tuning tool will delete prior copies.

The self-tuning tool will attempt to allocate sufficient `ICACHE` and `CONTENT CACHE` to accommodate a database expansion of 30%. However, the results might change due to default sizes for minimum number of database instances, maximum amount of virtual storage to commit to content cache, and the effect of very large classes.

- A file named `EDMPPROF_SECTION.DAT` is also created. It contains a replica of the `MGR_CLASS` and `MGR_CACHE` sections that were merged into the master `edmpprof` file and replaced the existing `MGR_CACHE` and `MGR_CLASS` sections when the self-tuning tool started.
- A file (`LIST_PREFIX.CSV`) is created, but currently not used. When implemented, this will contain a snapshot of database domain and class information, including the size and number of instances in each class at shutdown.

To implement the Configuration Server self-tuning tool:

1. Install the new `ZSHUTDWN` REXX method into the `rexx` directory.

Note: If a previous, customized version of `ZSHUTDWN` REXX exists in the `rexx` directory, be sure to copy any customizations from it to the new one. Then delete the previous one, or move it down to the `rexx\NOVADIGM` directory.

2. Shutdown the Configuration Server.
(After the Configuration Server has shutdown, in the location that housed the `edmpprof` file when the Configuration Server was started, there should be either: an updated `edmpprof` file along with the original (but now renamed) `edmpprof` file, or the revised `edmpprof` file that overwrote the original.)
3. Restart the Configuration Server.

Sentient REXX Functions

Ten RCA REXX functions perform specific actions. These actions encompass the getting and setting of objects and variables; object resolution; retrieving object names and properties; and

UTF-8 and local code page strings.

Knowledge of REXX operators and word lists is assumed.

The REXX extensions are:

- EDMGET (objects)
- EDMGETV
- EDMSET (objects)
- EDMSETV
- EDMRESO
- NvdCurrentObjects
- NvdObjectInfo
- NvdObjectInfoEX
- NvdL2U
- NvdU2L

EDMGET

(object,heapnumber)

- *object* is the Configuration Server in-storage object that is to be processed. Object names are usually specified in uppercase; EDMGET will uppercase the value that is specified for *object*.
- *heapnumber* is the heap—of the above-referenced object—that is to be processed. If *heapnumber* is not specified, it defaults to zero (**0**), meaning the current heap. The *heapnumber* can range from one (**1**) to *the number heaps in the object*.

Note: On the Configuration Server, the numbering of heaps starts at **1**, whereas on an Sentient agent the numbering of heaps starts at **0**.

EDMGET is used to retrieve the contents of a CSDB object, one heap at a time, into a REXX's variable pool. In addition to returning a return code, EDMGET will set the REXX variables to contain the object's heap and variable count, the attribute names that are saved in the object, and their "as-is" and "resolved" ("indirect") values. The clause

```
rcode = EDMGET("zmaster")
```

will copy the contents of the first heap of the ZMASTER object into REXX variables. The REXX variable names that are created are in the following form.

```
object.attribute
```

```
object
```

```
objectvars
```

```
object.resolved.attribute
```

```
object1 ... objectn
```


For example, to check the value of the ZUSER attribute in the ZMASTER object, use the REXX name

```
zmaster.zuser.
```

Note: For additional information on using REXX stem variables that are in this form, refer to the section on *Symbols* in the *Sentient Enterprise Application Manager and Configuration Server REXX Programming Guide*.

For example, the clause

```
Say zmaster.zuserid /* might return Fred */
```

- If the value of `zmaster.zuserid` was `&(object.attribute)`, its syntax is known as substitution, which is a form of value indirection.
- If the value of `zmaster.zuserid` was `&(SESSION.USERID)`, the USERID attribute in the SESSION object can be checked for the value. However, the SESSION object does not have to be opened to check the value because it is saved in the REXX variable `zmaster.resolved.zuser.`

If the value of `zmaster.zuserid` was not indirected, the value of `zmaster.resolved.zuserid` is the same.

In addition to the attributes in the object, two other REXX variables are created. The first has the same name as the object that is being processed; the second is the *objectname* with `vars` appended to the end of it.

Therefore, if the object that is being processed is ZMASTER, then:

```
say zmaster /* is the number of heaps in the object */
```

```
say zmastervars /* is the count of attributes in the object */
```

The variables `<object>1` ... `<object>n` contain the name of the attribute that is in the object. For example,

```
say zmaster1 /* might be ZUSERID */
```

To get all the attributes in an object, code the following.

```
object = "zmaster"
heaps = value(object)
vars = value(object"vars")
vnames = ""
do vv = 1 to vars
  vnames = vnames value(object|| vv)
end vv
```

If *heapnumber* is not specified, it defaults to zero (0), meaning the current heap. The heapnumber can range from one (1) to *the number heaps in the object*.

Note: On the Configuration Server, the numbering of heaps starts at 1, whereas on a Sentient agent the numbering of heaps starts at 0.

The return value is:

- 0 if the object was copied to REXX storage
- Ø if it failed; the Configuration Server log will indicate the exact error.

EDMGET has a high cost to process a CSDB object, because each call to EDMGET will return all of the attributes in the object, thus creating all of the REXX variables that are noted above. If the intent (in the REXX code) is to get the value of some of the attributes that are defined in the object, then use EDMGETV, which is discussed in the next section

EDMGETV

(object,attribute,heapnumber,resolve)

- *object* is the CSDB object that is to be processed.
The value that is specified will be automatically uppercased. Valid syntax for object names is 1 to 8 bytes; invalid names will generate a REXX syntax error.
- *attribute* is the attribute—of the above-referenced object—that is to be read.
The value that is specified will be automatically uppercased. Valid syntax for attribute names is 1 to 8 bytes; invalid names will generate a REXX syntax error.
- *heapnumber* is the heap—of the above-referenced object—that is to be processed.
The *heapnumber* can range from one (1) to the number heaps in the object.
 - If *heapnumber* is not specified, it defaults to zero (0), meaning the current heap.
 - If the value that is specified is out of range, a REXX syntax error is generated.
- *resolve* is a flag that can accept the values of 0, 1, Y, and N.
 - If *resolve* is not specified, it defaults to 0 (N), and the value that is returned is the contents of the object and attribute that are specified in arguments 1 and 2.
 - If 1 (Y) and the contents of the object and attribute that are specified in arguments 1 and 2 are in the form of &(object.attribute), the value that is returned is an “indirect” value. If the contents are not in the form of &(object.attribute), the contents are returned “as-is.”

Note: For additional information on REXX operators, refer to the *Sentient Enterprise Application Manager and Configuration Server REXX Programming Guide*.

EDMGETV will return the null string ("") if either the object or the attribute does not exist.

Use the REXX “strictly equal” operator, consecutive equals signs (==), to check for the null string ("").

Note: For additional information on signal instruction, refer to the *Sentient Enterprise Application Manager and Configuration Server REXX Programming Guide*.

For example:

```
thisuser = edmgetv("zmaster", "userid") /* such as FRED */  
  
this_data = edmgetv("zmaster", "mydata")  
  
if "" == this_data  
  
then say "The attribute mydata does not exist"  
  
else say "The attribute mydata is" this_data
```

EDMSET

(object,heapnumber)

EDMSET is used to migrate the value of REXX variables into the specified object at the specified heap. Attributes that are not defined in the object are added; existing attributes are updated. To add a new heap to an existing object, set the value of *heapnumber* to one more than the current number of heaps in the object (that is, *the current number heaps in the object + 1*). The variables that are migrated to the object are in the following form.

object.attribute

The process scans all of the currently allocated (REXX) variables, looking for names that start with the value that is specified for *object*, such as **zmaster**. When a match is found, it is checked to determine whether the second part of the argument (*attribute*) conforms to an attribute name (1 to 8 bytes in length).

If so, it is checked to determine whether it fits the size restrictions (no more than 255 bytes) of an attribute.

If it passes this check, the REXX variable is migrated to the object.

- If either of these format checks fails, the REXX variable is skipped.
The return value is:
 - 0 if no errors were encountered during the migration.
 - Ø if errors were generated; the Configuration Server log will indicate the exact error.

As noted with EDMGET, EDMSET is a costly process to save one or two attributes. For this processing, it would be better to use EDMSETV, which is discussed in the next section.

EDMSETV

(object,attribute,value,heapnumber)

- *object* is the CSDB object that is to be processed.
The value that is specified must be from 1 to 8 bytes in length; this value will be automatically uppercased.
- *attribute* is the attribute—of the above-referenced object—that is to be written.
The value that is specified must be from 1 to 8 bytes in length; this value will be automatically uppercased.
- *value* is the text to be saved to the attribute.
- *heapnumber* is the heap—of the above-referenced object—that is to be processed.
If *heapnumber* is not specified, it defaults to zero (0), meaning the current heap. The

heapnumber can range from one (1) to the number heaps in the object.

If the specified arguments do not conform to the syntax, a REXX syntax error is generated with the error text being written to the Configuration Server log.

If EDMSETV completes (without a syntax error), a return value of:

- 0 indicates that the value was saved in the specified object/attribute.
- Ø indicates that it failed; the Configuration Server log will indicate the exact error.

EDMRESO

EDMRESO runs a “secondary” resolution on the instance name that is specified by

```
rcode = EDMRESO(f.d.c.i).
```

Note: The `f.d.c.i` indicates the CSDB tree structure, FILE.DOMAIN.CLASS.INSTANCE.

EDMRESO status is indicated by return codes of:

- 0 indicates that the process started
- Ø indicates that the process did not start; check the Configuration Server log for details.

Configuration Server Database Object and non-ASCII Object Names

For REXX variables to be created by EDMGET, their names need to consist of characters that are valid REXX variable names. (See the *Sentient Enterprise Application Manager and Configuration Server REXX Programming Guide*) Such is not the case for CSDB object names that consist of multi-byte UTF-8 sequences.

Objects can still be processed via EDMGETV and EDMSETV because all of the arguments to these functions are (REXX) strings, as in:

```
say EDMGETV("zmaster", "zos")
```

All object processing can be done with EDMGETV and EDMSETV, but the numbers of heaps and attributes, which are set by the EDMGET call, would not be known.

Additional Functions

Three additional REXX functions retrieve object names and properties.

- "[NvdCurrentObjects](#)" on the next page
returns information pertaining to the current Configuration Server session.
- "[NvdObjectInfo](#)" on the next page
returns information pertaining to heap counts and attributes.
- "[NvdObjectInfoEX](#)" on page 578
is similar to `NvdObjectInfo` but includes attribute-property information.

These functions are valid only on Windows operating systems.

NvdCurrentObjects

```
Objectlist = NvdCurrentObjects() /* no arguments */
```

NvdCurrentObjects returns a list of words that contains the names of the objects that exist in the current Configuration Server session. So,

```
AllObjects = NvdCurrentObjects()
```

returns a list of all object names that are allocated in the current Configuration Server-Sentient agent connection. Use this information to check the availability of, for example, ZMASTER, as in:

```
Objects = NvdCurrentObjects()
```

```
If wordpos("ZMASTER", Objects) > 0 /* Note that object names are case sensitive */
```

```
then say "zmaster exists"
```

```
else say "Can't find ZMASTER"
```

NvdObjectInfo

```
Infolist = NvdObjectInfo(object)
```

object is the CSDB object that is to be processed.

The value that is specified will be automatically uppercased. The name that is specified has to be from 1 to 8 bytes; invalid names will generate a REXX syntax error.

Returns

NvdObjectInfo returns a list of words in which

- the first word is the heap count for the specified object,
- the second word is the total size of all the attributes in the specified object, and
- the rest of the list contains the names of the attributes that are in the specified object.

Note: Encrypted attributes are not returned.

So, the above-mentioned information can be determined by using the following REXX code.

```
Object = "zmaster"
```

```
Parse value NvdObjectInfo(Object) With Heaps TotalSize AttributeNames
```

```
AttributeCount = words(AttributeNames)
```

If there are any errors, only the heap count is returned, and is so as a negative number.

```
if heaps > 0
```

```
then nop
```

```
else exit 16
```

To get the attribute names:

```

Do nn = 1 to AttributeCount
  say word(AttributeNames, nn)
End nn

```

NvdObjectInfoEX

```
Infolist = NvdObjectInfoEX(object)
```

- *object* is the CSDB object that is to be processed.
The value that is specified will be automatically uppcased. The name that is specified has to be from 1 to 8 bytes; invalid names will generate a REXX syntax error.

Returns

NvdObjectInfoEX returns “property” information for each of the attributes that is returned—including encrypted attributes.

```
ObjectInfo = NvdObjectInfoEX(objectname)
```

The property information is returned in the following form, with the last four values returned as numerals.

attributename:size:mflags:cflags:vtype

- *size*
“size” is a decimal numeric value from 1 to 255
- *mflags*
“manager flags” is a decimal numeric value from 0 to 255.
- *cflags*
“client flags” is a decimal numeric value from 0 to 255.
- *vtype*
“attribute type” is a single ASCII character that describes the attribute “type.” Valid values are **C** (connection), **V** (variable), and **M** (method).
This is a decimal numeric value, so **V** would be specified as **86**.

To access this information, use the following REXX code.

```

parse value nvdobjectinfoex("zmaster") with heaps totalsize attributes
do aa = 1 to words(attributes)
  parse value word(attributes, aa) with attribute ":", maxlen ":" mflags
  ":" cflags ":" vtype .
  wstr = d2c(vtype)
  if datatype(wstr, 'm')
  then vtype = wstr
  else vtype = d2x(vtype)
  mflags = x2b( d2x(mflags, 2))
  cflags = x2b( d2x(mflags, 2))

```

```
say attribute mflags cflags vtype
end aa
```

Lastly, two utility functions to convert between local code page (LCP) and UTF-8 strings.

NvdL2U

```
UTF8Value = NvdL2U(LCPVALUE)
```

NvdU2L

```
LCPValue = NvdU2L(UTF8Value)
```

The REXX functions **stream**, **linein**, **lineout**, **charin**, **charout**, and **lines** reference external files. The filename and path that are specified must be in LCP. These functions can be used to convert the value. For example, to read a file that has been saved in the Configuration Server's method path, use:

```
wstr = edmgetv("ZCVT", "MTHPATH")
if right(wstr, 1) = rxxospathseparator()
then nop
else wstr = wstr || rxxospathseparator()
mydata = nvdu2l(wstr || "mydata.file")
info = stream(mydata, 'c', 'query exists')
```

In this case, *mydata* will contain LCP characters that are the value needed for the REXX stream function.

Delete this text and replace it with your own content.

ZCVT and ZTCBG Objects

These two objects assist you in determining current values for CSDB tasks and connects, whether they are running. ZCVT performs a global function—determining values of tasks and connects for the entire CSDB, whereas ZTCBG determines the values that are relevant to the specified task or connect.

The ZCVT and ZTCBG objects (*objname*) work with the EDMGET, EDMGETV, EDMSET, and EDMSETV REXX extensions, as described in the previous section.

"[ZCVT Table of Variables](#)" below and "[ZTCBG Table of Variables](#)" on page 589 present the variables associated with ZCVT and ZTCBG, respectively.

ZCVT Table of Variables

ZCVT Table of Variables

ZCVT Variables

Variable Name	Variable Type	Description
VERMAJ	USHORT	Configuration Server major version number
VERMIN	USHORT	Configuration Server minor version number
VERREVNO	USHORT	Configuration Server revision number
VERREVLE	UCHAR	Configuration Server revision letter
VERBLDNO	ULONG	Configuration Server build number
OSNAME	STR	Configuration Server's operating system
TOLOGONS	LONG	Total number of logons to the Configuration Server
TRCOMMCB	FLAG	COMM Trace value
TRCOMDAT	FLAG	COMM trace value
TRDSCOMP	FLAG	DSCOMP trace value
TRTEST	FLAG	TEST trace value
TRDYNALO	FLAG	ALLOC trace value
TRVARSTG	FLAG	VARSTG trace value
TRDBCBC	FLAG	ODBCBS trace value
TRDBDATA	FLAG	ODBDATA trace value
TRAUDIT	FLAG	AUDIT trace value
TRPROFIL	FLAG	PROFLE trace value
TRRESRCE	FLAG	RESOURCE trace value
TRPROMOT	FLAG	PROMOTE trace value
TRCONFIG	FLAG	CONFIG trace value
TRMETHOD	FLAG	METHOD trace value
TRCPIC	FLAG	CPIC trace value
TRADMIN	FLAG	ADMIN trace value
TRRESLV0	FLAG	OBJRES0 trace value
TRBINDFL	FLAG	DATA trace value
TRDAXFRM	FLAG	TRAN trace value
TR3270BU	FLAG	BUFF trace value

TRCOMM	FLAG	COMM trace value
TRFILPRO	FLAG	FILE trace value
TROBJXFR	FLAG	OBJXFER trace value
TROBJCRC	FLAG	OBJCRC trace value
TRREXX	FLAG	REXX trace value
TRVARS	FLAG	VAR trace value
TRSUBST	FLAG	SUBST trace value
TRDESENC	FLAG	DES trace value
TRCOMPR	FLAG	CMPR trace value
TROBJRES	FLAG	OBJRES trace value
TRIMPLOD	FLAG	IMPL trace value
TREXPLOD	FLAG	EXPL trace value
TRLASIDE	FLAG	LOOKASID trace value
TRENQUE	FLAG	ENQDEQ trace value
TRSTATS	FLAG	STATS trace value
TRRESLV1	FLAG	OBJRES1 trace value
TRTCPIP	FLAG	TCP trace value
TRADMPRM	FLAG	ADMPROM trace value
TRNOTIFY	FLAG	NOTIFY trace value
TRSESBK	FLAG	SESSBLK trace value
TRSTORAG	FLAG	Storage trace value
TRY2K	FLAG	YEAR2000 trace value
TRDMA	FLAG	DMA trace value
TRVSAPI	FLAG	VSAM trace value
TRVSCB	FLAG	VSAMRPL trace value
TRVSDATA	FLAG	VSAMDATA trace value
TRREXOFF	FLAG	REXXOFF trace value
SHTINDIC	UCHAR	Shutdown indicator
SHTLGMGR	UCHAR	Log Configuration Server shutdown indicator

REXALLOC	LONG	REXX allocate count
TOPTSKID	LONG	ZTOPTASK ID
LOGMGRID	LONG	ZLOGMGR ID
ULGMGRID	LONG	ZULOGMGR ID
CLKMGRID	LONG	ZCLKMGR ID
TSKMGRID	LONG	ZTASKMGR ID
MGRTYPE	UCHAR	MANAGER_TYPE value
TASKTYPE	UCHAR	TASK_TYPE value
MEMTYPE	UCHAR	MEMORY_TYPE value
DBLUPDATE	STR	Date of last CSDB update
DBLUTIME	STR	Time of last CSDB update
DBLCKCNT	USHORT	CSDB lock count
DBSTATUS	UCHAR	CSDB status
BYTELEVD	UCHAR	Byte level differencing on value
DNYDUPIP	FLAG	ALLOW_DUPLICATE_IP_ADDRESS value
TRUSTEDP	FLAG	Authorized trusted partner value
TSOSRVCE	FLAG	TSO service value
LICERROR	FLAG	License violation error value
ACALLADM	FLAG	ALWAYS_CALL_ZADMIN value
OKDUPINS	FLAG	ALLOW_DUPLICATE_INSTANCES value
QUITASKS	FLAG	Quiesce task
QUITRANS	FLAG	Quiesce transaction
LALLTCP	FLAG	TCP/IP logons allowed value
REXDISAB	FLAG	REXX off
MODNAMLO	FLAG	SHOW_MODULE value
MODVERLO	FLAG	SHOW_VERINFO value
TCPRHNAM	FLAG	GET_REMOTE_HOST_NAME value
HISTORY	UCHAR	History file value
ACCESSA	UCHAR	MGR_ACCESS ADMIN value

ACCESSC	UCHAR	MGR_ACCESS CONSOLE value
ZTIME	STR	Time in HH:MM
ZTIME24	STR	Time in HH:MM on 24-hour clock
ZAMPM	STR	AM or PM value
ZDATE	STR	MM/DD/YYYY date form
ZDATEDMY	STR	DD/MM/YYYY date form
ZMONTH	STR	Short value for month (e.g., Jan)
ZMONTHLNG	STR	Long value for month (e.g., January)
ZDATEJUL	STR	Julian date
ZDATEYMD	STR	YYYY/MM/DD form for sorting
ZDAT2YMD	STR	YYYY/MM/DD form for sorting – full MM, DD value
MGRID	STR	Configuration Server ID (MGR_ID)
MGRNAME	STR	Configuration Server name (MGR_NAME)
DOMAIN	STR	Configuration Server domain name (DOMAIN)
STDOMANR	STR	Sentient agent start domain (START_DOMAIN)
STCLASSR	STR	Sentient agent start class (START_CLASS)
STDOMANA	STR	ATM start domain (START_DOMAIN)
STCLASSA	STR	ATM start class (START_CLASS)
DBASE	STR	CSDB used at startup (DBASE)
AGENT	STR	Agent ACB (AGENT)
TCPPORT	STR	TCP PORT value (TCP_PORT)
TCPUSRID	STR	USERID of TCP/IP address space
TORESO	LONG	Number of object resolutions
DEEPRESO	LONG	Deepest object resolution
TOOBJI	LONG	Number of inbound objects
TOOBJO	LONG	Number of outbound objects
ZERMXWRN	SHORT	Max number of warnings heaps
ZERMXERR	SHORT	Max number of error heaps
TOCOMP	LONG	Number of times compression done

TOCOMPI	LONG	Compression total bytes in
TOCOMPO	LONG	Compression total bytes out
TODCMPI	LONG	Decompression total bytes in
TODCMPO	LONG	Decompression total bytes out
TODCMP	LONG	Number to times decompression done
COMPSEED	LONG	Compression seed
TODBGETS	LONG	Number of GET operations
TODBPUTS	LONG	Number of PUT operations
TODBADDS	LONG	Number of ADD operations
TODBDELE	LONG	Number of DELETE operations
TOFILEIO	LONG	Global file i/o counts
TOFALLOC	LONG	Global file allocations
TSKLIMAX	SHORT	Max TASKLIM
TSKLIMIT	SHORT	TASKLIM (TASKLIM)
TSKLHARD	SHORT	HARD TASKLIM (TASKLIM_HARD) calculated as ALSLOTS + 5 + TSKLIMIT + (TSKLIMIT/3)
TSKLSTFT	SHORT	SOFT TASKLIM calculated as TSKLIMIT + ALSLOTS
TSKLDLTA	SHORT	TASKLIM Delta calculated as (TSKLSTFT - TSKLIMIT) + 1
MAXRESCL	LONG	Max number of resource per Sentient agent
MAXRESAL	LONG	Max number of resource all Sentient agents
TOMETBIN	LONG	Number of methods run – ASM and C
TOMETREX	LONG	Number of methods run – REXX
TOXNRCVD	LONG	Number of transaction received
TOXNRJCT	LONG	Number of transaction rejected
PLSTATUS	PTR	Pointer to disable Pools heap
PLGLBHEP	FLAG	Global/local pools in used flag
PLCONTIG	FLAG	Pools are allocated contiguous flag
PLEXPSIZ	ULONG	Pool expansion size
PLCSCRED	ULONG	Current storage credit
PLPSGUAR	ULONG	Percent Storage Guard

PLCSCUSH	ULONG	Current storage cushion
PLMSCUSH	ULONG	Minimum storage cushion
PLPOLHWM	ULONG	Largest polar bytes used
TSKPRIV	ULONG	Task private size
TSKSTSZ	ULONG	Task stack size
TSKSTHWM	ULONG	Task stack size overuse HWM
TSKHPSIZ	ULONG	Task heap size
TSKHPHWM	ULONG	Task heap size overuse HWM
TIMEOCOM	LONG	TIMEOUT_COMM
TIMEONCM	LONG	TIMEOUT_NCOMM
TIMEOADM	LONG	ADMIN_TIMEOUT
TIMEODMA	LONG	DMA_TIMEOUT
RETRYBUS	LONG	BUSY_RETRY
RETRYDIS	LONG	DISABLE_RETRY
DSCOMPI	LONG	Data stream compression total bytes in
DSCOMPO	LONG	Data stream compression total bytes out
DSCOMPT	LONG	Number of times data stream compression done
SNDTHRTL	ULONG	SEND_THROTTLE
TIMEONFT	SHORT	NFYT_TIMEOUT
TIMEONFS	SHORT	NFYS_TIMEOUT
TIMEONFD	SHORT	NFYD_TIMEOUT
OBJNMASK	STR	Object name mask for traces
VARNMASK	STR	Variable name mask for trace
LOGLNTSK	ULONG	TASK_LOG_LIM
MAXRSTSK	ULONG	TASK_RESO_LIM
MAXREC	LONG	MAXREC
BUFTCP	LONG	BUFTCP
LOGDIR	STR	DIRECTORY
LOGTHRES	LONG	THRESHOLD

LOGLNCNT	LONG	Log file line count
LOGSTINT	LONG	STORAGE_INTERVAL
LOGFSIZE	LONG	FLUSH_SIZE
LOGMWIDT	USHORT	MESSAGE_WIDTH
LOGMPREF	STR	MESSAGE_PREFIX
LOGMDATE	UCHAR	MESSAGE_DATE
LOGMLDEL	UCHAR	MESSAGE_DELIMITER
LOGMRDEL	UCHAR	MESSAGE_DELIMITER
LOGPSIZE	LONG	PIPE_SIZE
LOGFLUSH	FLAG	Log flush
LOGSWITC	FLAG	Log switch
LOGELOFF	FLAG	DISABLE_NT_EVENT_LOGGING
LOGSLOFF	FLAG	DISABLE_SNMP_TRAP_LOGGING
LOGSFREQ	STR	SWITCH_TOD
LOGFFREQ	STR	FLUSH_INTERVAL
LOGBPIPE	LONG	Log bytes in pipe
ULGDIR	STR	USERLOG DIRECTORY
ULGTHRES	LONG	USERLOG THRESHOLD
ULGLNCNT	LONG	USERLOG LINECOUNT
ULGFSIZE	LONG	USERLOG FLUSH_SIZE
ULGMWIDT	USHORT	USERLOG MESSAGE_WIDTH
ULGACTIV	FLAG	USERLOG ACTIVATE
ULGPSIZE	LONG	USERLOG PIPE_SIZE
ULGFLUSH	FLAG	USERLOG flush
ULGSWITC	FLAG	USERLOG switch
SIMTSKPC	LONG	Simulation TASKS_PER_CONNECT
STATPATH	STR	Stats path
STATINTV	LONG	Stats interval
MTHPATH	STR	METHOD_PATH

MTHMLIMI	LONG	LOG_LIMIT
MTHTIMEO	LONG	TIMEOUT
DBPATH	STR	DBPATH
DBPPRIM	STR	PRIMARY DB path
DBPSECO	STR	SECONDARY DB path
DBPPROF	STR	PROFILE DB path
DBPRESO	STR	RESOURCE DB path
DBPHIST	STR	HISTORY DB path
DBPNOTI	STR	NOTIFY DB path
REXXPATH	STR	REXX_PATH
SYSPATH	STR	System path
DMASPATH	STR	DMA_STAGE_PATH
MGRSETFI	STR	PROFILE path
EXPTPATH	STR	EXPORT_PATH
USRPATH1	STR	USER_PATH1
USRPATH2	STR	USER_PATH2
USRPATH3	STR	USER_PATH3
USRPATH4	STR	USER_PATH4
USRPATH5	STR	USER_PATH5
USZTCBGS	SHORT	Number of used ZTCBGs
CACSEGS	USHORT	CACHE_SEGMENTS
CACSIZE	ULONG	CACHE_SIZE
CACMAXE	ULONG	Max cache entries
CACSTATS	USHORT	CACHE_STATS
CACFULL	USHORT	CACHE full
CACCLOSE	USHORT	CACHE closed
ICASIZE	ULONG	ICACHE_SIZE
ICACLOSE	UCHAR	ICACHE closed
SMMAILDR	STR	MAIL_DIR

SMMGRMID	STR	MGR_MAIL_ID
SMLOCHST	STR	Local host NAME
SMDNSSRV	STR	DNS_SERVER
SMSMTPRT	USHORT	SMTP_PORT
SMSPLCNT	ULONG	Spooled mail count
SMTIMEO	USHORT	MAIL_TIMEOUT
SMMAXSPL	USHORT	MAX_TIME_IN_SPOOL
SMRETRYI	USHORT	RETRY_INTERVAL
SNPORT	USHORT	SNMP_PORT
SNLOGPRT	USHORT	SNMP_LOGGER_PORT
SNMGRPRT	USHORT	SNMP_MANAGER_PORT
SNRUNEXT	FLAG	RUN_AS_EXTENSION
SNIPADD	STR	SNMP_IP_ADDR
SNMIPAD	STR	SNMP_MANAGER_IP_ADDR
SNMIPAD2	STR	SNMP_MANAGER_IP_ADDR2
SNMIPAD3	STR	SNMP_MANAGER_IP_ADDR3
SNCMNT	STR	SNMP_COMMUNITY
SNSTCMNT	STR	SNMP_SET_COMMUNITY
SNZERSEV	SHORT	SNMP_ZERROR_SEVERITY
ALSLOTS	LONG	ATTACH_LIST_SLOTS
ALVINTVL	LONG	VERIFY_INTERVAL
ALRLIMIT	LONG	RESTART_LIMIT
DMSECMTH	STR	SECURITY_METHOD
DIINTVL	ULONG	DIAGNOSTIC_INTERVAL
DIADBYTE	ULONG	DIAGNOSTIC_MIN_DB_BYTES
DIALBYTE	ULONG	DIAGNOSTIC_MIN_LOG_BYTES
DBVERIFY	STR	VERIFY_DEPTH
DBAUTOFIX	FLAG	DB_AUTOFIX
METHDLLS	FLAG	Run methods as DLLS

ACTSKMON	ULONG	Number of monitors
ACTSKCON	ULONG	Number of consoles
ERREMAIL	STR	UserEmailErrorsTo
DBEEMAIL	FLAG	DBERROR e-mail
DBESHTDN	FLAG	DBERROR SHUTDOWN
DBESNMP	FLAG	DBERROR SNMP
POLCYSVR	STR	Status of the Policy Server (enabled/disabled)
RIM	STR	Status of the Inventory Manager (enabled/disabled)
RMP	STR	Status of the Portal (enabled/disabled)

ZTCBG Table of Variables

ZTCBG Table of Variables

ZTCBG Variables

Variable Name	Variable Type	Description
AUDFLAG	FLAG	Audit trace flag
TSKNAME	STR	Name of this task or Sentient agent
TSKSTTIM	STR	Time when task started
TSKSTDAT	STR	Date when task started
TSKLSTCO	STR	Time of last communication transaction
FREEMAIN	FLAG	Return storage on last free call
SHORTSTO	FLAG	Retry due to Short on storage
ZTERMINI	FLAG	Terminal task initiated
STOCCUR	ULONG	Current storage credit
STOCRHEP	ULONG	Heap credit
STOCRSTK	ULONG	Stack credit
STOCRPUT	UONG	Private credit
STOCRPOO	ULONG	Zpools credit
STOCCUS	ULONG	Credit cushion
MTHNAME	STR	Child name
MTHLIBNA	STR	Method library name

MTHLIBHA	ULONG	Method library handle
MTHLIBEN	ULONG	Method library entry point
MTHTHPRM	PTR	Method thread parameter pointer
TASKID	ULONG	Unique ID of this task
TASKPAR	ULONG	Unique ID of the parent task
USERID	STR	Task's user ID
TASKTYPE	STR	TASK TYPE
DEEPRESO	LONG	DEEPSET OBJECT RESOLUTION
OBJSRESO	LONG	NUMBER OF OBJECTS RECEIVED
OBJSRECV	LONG	NUMBER OF OBJECTS RECEIVED
OBJSSENT	LONG	NUMBER OF OBJECTS SENT
TRCOMDAT	FLAG	COMM TRACE FLAG
TRDSCOMP	FLAG	DSCOMP TRACE FLAG
TRTEST	FLAG	TEST TRACE FLAG
TRDYNALO	FLAG	ALLOC TRACE FLAG
TRVARSTG	FLAG	VARSTG TRACE FLAG
TRAUDIT	FLAG	AUDIT TRACE FLAG
TRPROFIL	FLAG	PROFILE TRACE FLAG
TRRESRCE	FLAG	RESOURCE TRACE FLAG
TRPROMOT	FLAG	PROMOTE TRACE FLAG
TRCONFIG	FLAG	CONFIG TRACE FLAG
TRMETHOD	FLAG	METHOD TRACE FLAG
TRCPIC	FLAG	CPIC TRACE FLAG
TRADMIN	FLAG	ADMIN TRACE FLAG
TRRESLVL	FLAG	OBJRES0 TRACE FLAG
TRBINDFL	FLAG	DATA TRACE FLAG
TRDAXFRM	FLAG	TRAN TRACE FLAG
TR3270BU	FLAG	BUFF TRACE FLAG
TRCOMM	FLAG	COMM TRACE FLAG

TRFILPRO	FLAG	FILE TRACE FLAG
TROBJXFR	FLAG	OBJXFER TRACE FLAG
TROBJCRC	FLAG	OBJCRC TRACE FLAG
TRREXX	FLAG	REXX TRACE FLAG
TRVARS	FLAG	VAR TRACE FLAG
TRSUBST	FLAG	SUBST TRACE FLAG
TRDESENC	FLAG	DES TRACE FLAG
TRCOMP	FLAG	CMPR TRACE FLAG
TROBJRES	FLAG	OBJRES TRACE FLAG
TRIMPLD	FLAG	IMPL TRACE FLAG
TREXPLOD	FLAG	EXPL TRACE FLAG
TRLASIDE	FLAG	LOOKASID TRACE FLAG
TRENQUE	FLAG	ENQDEQ TRACE FLAG
TRSTATS	FLAG	STATS FLAG
TRRESOL1	FLAG	OJBRES1 TRACE FLAG
TRTCPIP	FLAG	TCP TRACE FLAG
TRADMPRM	FLAG	ADMPROM TRACE FLAG
TRNOTIFY	FLAG	NOTIFY TRACE FLAG
TRSESBK	FLAG	SESSBLK TRACE FLAG
TRSTORAG	FLAG	STORAGE TRACE FLAG
TRY2K	FLAG	YEAR2000 TRACE FLAG
TRDMA	FLAG	DMA TRACE FLAG
TRVSAPI	FLAG	VSAM TRACE FLAG
TRVSCB	FLAG	VSAMRPLS TRACE FLAG
TRVSDATA	FLAG	VSAMDATA TRACE FLAG
TRREXOFF	FLAG	REXXOFF TRACE FLAG
STBBSNT	FLAG	BB SENT FLAG
STNOSNAP	FLAG	NO SNAP FLAG
STCOWAIT	FLAG	WAIT FOR COMM OP FLAG

STPWDVER	FLAG	EDATS SF ORDERS OK FLAG
STTIMOUT	FLAG	TIMEOUT IN PROGRESS FLAG
STFORTER	FLAG	FORCED TREMINATION FLAG
STPARSES	FLAG	PAR SESS PARTNER FLAG
STSESEST	FLAG	SESSION ESTABLISHED FLAG
STSESTER	FLAG	SESSION TERMINATED FLAG
STSESLST	FLAG	SESSION LOST FLAG
STTSKABN	FLAG	TASK ABENDED FLAG
STSESSND	FLAG	SEND FLAG
STNOPDS	FLAG	NO PARSE R/S DATA STREAM FLAG
STTSKINA	FLAG	TASK BEING INACTIVATED FLAG
STTIMSND	FLAG	TIME SENT FLAG
STNODSCO	FLAG	NO DS COMPRESSION FLAG
STABTRES	FLAG	ABORT OBJECT RESOLUTION FLAG
STABTLEG	FLAG	ABORT OBJECT RESOLUTION FLAG
STSTRLOG	FLAG	IN LOGGER FLAG
STEOT	FLAG	EOT FLAG
STNOSUB	FLAG	NO SUBSTITUTION FLAG
STDRAINS	FLAG	DRAIN FLAG
STCONSOL	FLAG	CONSOLE IS RUNNING
STHRDLCK	FLAG	SYSTEM HARDLOCKED FLAG
STSSRESO	FLAG	SINGEL SERVICE RESOLUTION FLAG
STUSEMET	FLAG	USER METHOD RUNNING FLAG
STMSG LIM	FLAG	LOG MSG LIMIT REACHED FLAG
STMETMES	FLAG	METHOD MSG LIMIT REACHED FLAG
STREXMET	FLAG	REXX METHOD RUNNING FLAG
TOCOMP	LONG	NUMBER OF TIMES COMPRESSION DONE
TOCOMPI	LONG	COMPRESSION TOTAL BYTES IN
TOCOMPO	LONG	COMPRESSION TOTAL BYTES OUT

TODCOMPI	LONG	DECOMPRESSION TOTAL BYTES IN
TODCOMPO	LONG	DECOMPRESSION TOTAL BYTES OUT
TODCOMP	LONG	NUMBER OF TIME DECOMPRESSION DONE
TODBGETS	LONG	NUMBER OF GETS
TODBPUTS	LONG	NUMBER OF PUTS
TODBADDS	LONG	NUMBER OF ADDS
TODBDELE	LONG	NUMBER OF DELETES
TOFILEIO	LONG	FILE I/O COUNT
TOFALLOC	LONG	FILE ALLOCATION COUNT
TOMTHBIN	LONG	NUMBER OF METHODS RUN – ASM AND C
TOMTHREX	LONG	NUMBER OF METHOS RUN – REXX
REMIPNAM	STR	IP NAME OF REMOTE CLIENT

Configuration Server Methods

Overview

A method is a program or procedure that can be packaged and exchanged as an object, specifically as an instance of the METHOD Class. By connecting an instance of this class to another class instance, an Sentient administrator can specify where and when that program/procedure will run. An Sentient administrator can also run a method from a REXX script, thereby enabling the execution of methods outside of the object resolution process. The following is an example of the format that is used to execute a method in this way.

```
ADDRESS EDMLINK ZOBJCMR 'ZTEST'
```

EDMLINK is a method that enables an Sentient administrator to process other methods. It returns the return code of the invoked method. The format for EDMLINK is:

```
ADDRESS EDMLINK methodname 'Parameter associated with Method'
```

Configuration Server methods allow an Sentient administrator to manipulate in-storage objects and database entities (database components) at the system (Configuration Server) level as opposed to the Sentient agent or workstation objects.

- *Configuration Server Database components* are the entities (files, domains, classes, instances, and variables) that reside in the CSDB.
- *In-storage objects* are used or created during object resolution.

The Affects of Configuration Server Methods

The table "Methods affecting in-storage objects or CSDB entities" lists the Configuration Server methods that affect in-storage objects and database entities.

Note: The following table "Methods affecting in-storage objects or CSDB entities" lists only those Configuration Server methods that affect in-storage objects and database entities. Configuration Server methods that affect neither are not listed in this table.

Methods affecting in-storage objects or CSDB entities

Method	Affects
EDMMAILQ	In-storage objects
EDMMCACH	In-storage objects
EDMMDB	CSDB entities
EDMMRPRO	CSDB entities
ZDCLASS	CSDB entities
ZDELINS	CSDB entities
ZDELOBS	In-storage objects
ZDELPROF	CSDB entities
ZEXIST	CSDB entities
ZGETPROF	CSDB entities
ZOBJCMPR	In-storage objects
ZOBJCOPY	In-storage objects
ZOBJDELI	In-storage objects
ZOBJDELV	In-storage objects
ZOBJSORT	In-storage objects
ZPROMANY	CSDB entities
ZPUTHIST	In-storage objects
ZPUTPROF	In-storage objects
ZSIMRESO	In-storage objects
ZTOUCH	CSDB entities
ZVARDEL	In-storage objects

ZVARGBL	In-storage objects
ZVARLOG	In-storage objects
ZXREF	In-storage objects

Methods are often used in conjunction with one another to achieve a purpose. For example, you can use the ZDELOBSJ method to delete an in-storage object, and then execute ZOBJCOPY to copy an object, giving it the original object name.

Methods must be connected to other class instances at an appropriate point to achieve a desired result. For example, you do not want to delete an instance before it is used in object resolution, or create an instance if it will be immediately overwritten.

The default file and domain used by some methods can be specified in the DBASE and DOMAIN values of the MGR_STARTUP setting of the Configuration Server `edmprof` file.

Method Naming Standard

The standard that is used to name the methods is dissectible, enabling you to ascertain the method's use. All method names are structured as detailed in "[Configuration Server Methods Naming Standard](#)" below.

Configuration Server Methods Naming Standard

Symbol	Definition
EDM	Identifies the method as a Sentient method.
M	Identifies the method as a Configuration Server method.
DOBJ	Represents an abbreviation of the function for which the method can be used, in this case, Delete Object.

"Must Run" Methods

When you configure methods to run during the resolution process, you expect specific outcomes. If one method is intended to work in conjunction with another, or have a direct effect on the correct outcome of the resolution, the entire resolution process might depend on, first the existence of, then the successful launching of, this method.

You can designate a method as "must run," which means that, before continuing with the resolution process, the Configuration Server will determine if the method exists and can be run. If it is not found or cannot be launched, the return code for the method will be set to 16 (Abort Resolution). The resolution will then be halted.

If you do not designate a method as "must run," the Configuration Server does not recognize it as being essential to the outcome of the resolution and will continue processing based on the resulting return code. The only indications that the method was not processed are the return codes (as shown in messages in the log) and the result of the resolution path.

To configure a method as "must run," insert the ZMUSTRUN variable in the METHOD instance and set the value to YES. (The default value for ZMUSTRUN is **NO**.) You can also establish a

specific message to be returned by inserting the MSGONERR variable in the ZMETHOD instance.

Note: If no value is specified for MSGONERR, the following message will appear:
"CONFIGURATION UNCHANGED! UNABLE TO DETERMINE NEW CONFIGURATION."

Configuration Server Methods

This appendix is a reference for Configuration Server methods. For information on configuring and using Configuration Server methods, see ["ZTCBG Table of Variables" on page 589](#)

The following section describe each Configuration Server method, providing examples of use, a description, its parameters, and the associated possible return codes.

Note: All arguments are expected to be in the format, KEYWORD=VALUE, and delimited by commas.
Quotation marks (" ") are required when a value contains commas and/or embedded blanks.

EDMAILQ

This method deposits e-mail in the mail queue (outbox). For this method to execute correctly, the MGR_SMTP_MAIL section must be added to the Configuration Server `edmprof` file.

EDMAILQ Parameters

attach	Specifies attachment files. Multiple attachments can be listed by using a semicolon (;) delimiter between each attachment. For example, <code>c:\config.sys;c:\autoexec.bat</code> . Attachments are sent using MIME, and can be in binary. This parameter is optional.
from	The sender's address. This parameter is required.
mesgfile	Specifies the file that contains the message. Used in place of the parameter, <code>message</code> , if the message is greater than 255 bytes. This parameter is optional.
message	Specifies a brief message (limited to 255 bytes). This parameter is required.
subject	Specifies the subject of the e-mail. This parameter is optional.
to	Specifies the e-mail recipients. Multiple users can be listed by using a semicolon (;) delimiter between each recipient. This parameter is required.

Note: Parameters are used like keywords and are not case-sensitive.

Example

In the example below, an e-mail with a brief message is sent from **user1@company1.com** to **user2@company2.com**.


```
EDMMAILQ from=user1@company1.com,to=user2@company2.com,Message="This  
is a brief message"
```

Example

In the example below, a text file (c:\report.txt) with a subject (My report) is sent between the same users.

```
EDMMAILQ  
from=user1@company1.com,to=user2@company2.com,Mesgfile=c:\report.txt,S  
ubject="My report"
```

Return Codes

0	The method was successful.
8	An error was detected, the method failed.

EDMMCACH

This method refreshes or disables caching.

EDMMCACH Parameters

Parameter	Description
option	Caching option values are ENABLE or DISABLE.

Example

```
ADDRESS EDMLINK EDMMCACH 'option=ENABLE' ;
```

Return Codes

0	The method was successful.
8	An error was detected, the method failed.

EDMMDB

This method locks and unlocks the CSDB to all tasks except the Distributed Configuration Server.

EDMMDB Parameter

option	LOCK locks the CSDB to any incoming tasks except Distributed Configuration Server. UNLOCK makes the CSDB accessible to all incoming tasks.

Example

```
EDMLINK EDMMDB "OPTION=LOCK"
```

Return Codes

0	The method was successful.

8	An error was detected, the method failed.

EDMMGNUG

This method retrieves a list of local and global groups to which a specified user belongs.

Note: This method is functional in a Windows network environment only.

Usage

This method is used to issue a function call to a specific server to collect information about the Windows group membership of a user. It does not provide authentication of a particular user ID. These calls are issued under security provisions of the user used to start the Configuration Server when it runs as a normal task; or under a system account when the Configuration Server runs as a service. See the Security Requirements below for security limitations as defined by Microsoft.

Return Codes

0	The method was successful.
>0	An error was detected, the method failed.

The instance indicates which CSDB instance will be altered by specifying five control variables:

ZOBJCLAS	Target class, for example, ZADMIN.
ZOBJDOMN	Target domain, for example, SYSTEM.
ZOBJFILE	Target file, for example, PRIMARY.
ZOBJNAME	Target instance. This can be any valid instance name.
ZOBJFILE	Target file, for example, PRIMARY.

- On a **delete** request, only the control variables are used to identify the instance to be deleted, the remaining variables are ignored.
- On **add** and **update** requests, the variables in each instance contain the values used to update the instance in the CSDB.
- The fields that can be updated are variables, class connections, expressions, and methods.
- There are specification differences for the three field types.
However, regardless of field type, the target instances' field lengths determine the amount of data moved, and length adjustment is performed, including blank padding and truncation.

Method Input Parameters

The only parameter passed to the method is the name of the object containing the request. The following table details the input parameters and defaults for the method.

ZUSERID	Required variable used as user name.

NTSRVNAME	Name of the remote server on which the function is to execute. If this parameter is NULL, the local computer is used. The default is the local server.
ZOBJREQ	Name of the response object that will contain information about local and global (network) groups to which the user specified in ZUSERID belongs. The default is NTGROUPS .

Method Return Values

The EDMMGNUG method returns group membership information about a user in the object specified in ZOBJREQ (usually NTGROUPS). The following are the variables delivered by the method.

NTGRPLCT	Number of local groups on the server to which the user belongs.
NTGRPGCT	Number of global (network) groups on the server to which the user belongs.
NTGRPSCT	Total number of groups on the server to which the user belongs. (NTGRPLCT + NTGRPGCT)
NTGRPLxx	There are as many of these variables as there are local groups that a user belongs to on the specified server. xx = {1, NTGRPLCT}
NTGRPGxx	There are as many of these variables as there are global groups that a user belongs to on the specified server. xx = {1, NTGRPGCT}
MSGGRPLE	An error message, returned by the Network Management Functions, for request for the local group list to which the user belongs.
MSGGRPGE	An error message, returned by the Network Management Functions, for request for the global group list to which the user belongs.
NTUSER	Name of the user for which the function was executed. (This is the same as ZUSERID)
NTSRVNAME	Name of the remote server on which the function was executed.
ZMRC	Return code (set in in-bound and response objects).

EDMMPUSH

This method receives input requests, gets the required parameters, and then puts the requests to the right queues for processing by a specific Notify Manager. An in-bound object, or even a dynamic object, created because of the object resolution can be used to deliver requests to EDMMPUSH. The return code associated with the in-bound object might initiate further action. (For more information, see ["Configuration Server Methods" on page 596.](#))

EDMMPUSH Parameters

nfydelay	The interval for delay before trying to re-notify an Sentient agent. The default is the value specified in the NFYT_TIMEOUT setting of the <code>edmprof</code> file.

nfyhndl	The domain name of the NOTIFY File where the results of notifications will be stored. The heap number of the request object will become the instance name.
nfymrtry	The maximum number of retries. The default is the value specified in the NFY_RETRY setting of the section of the <code>edmprof</code> file.
ntfyrtim	Sentient timestamp defining the time after which the notification should occur.
nfyproc	Controls the processing of the current heap request. If Y, the heap will be processed. If N, the request for the current heap is ignored. The default is Y.
nfytype	Defines the type of the notify requested. Valid values are: TCP and EMAIL. Note: Only the first three bytes of the type are used for the identification. Therefore, EMAIL and EMA will be treated as the same. There is no default for this variable. If it is not defined, the current heap of the object will be ignored.
nfyuinfo	Allows you to enter user information.

Example

```
EDMLINK EDMMPUSH ZNOTIFY
```

Return Codes

0	The method was successful.
8	An error was detected, the method failed.

EDMMPUTD

This method is called when EDMSENDP is used to send the ZTRANSF object to the Configuration Server. It closes a security loophole. EDMSENDP will not work with the version 4.4 CM Configuration Server, therefore, it is necessary to modify your CSDB by adding a new instance in ZPROCESS and linking it to a ZMETHOD object that invokes the EDMMPUTD method. This will have to be done for each object that is sent to the Configuration Server using EDMSENDP.

EDMMPUTD enables you to receive multiple data types sent by the Inventory Manager, and enables you to specify where on the Configuration Server to store this data.

- With EDMMPUTD, you can inject a REXX method before EDMMPUTD gets dispatched to alter the location of the data based on appropriate criteria or to do security validation.
- The EDMMPUTD method handles the object and data sent from the Sentient agent's EDMSENDP method. Thus, the specifications for the inbound object are the same.
- PROCESS class instances must be configured for each inbound object needing EDMMPUTD to receive the appended inbound data and store it in a file.

The following table lists the attributes that EDMMPUTD expects.

Attributes Associated with EDMMPUTD

ZRSCMFIL	ZRSCMLOC	ZRSCMMEM

ZRSCRASH	ZRSCSTYP	ZOBJCLAS
ZOBJDOMN	ZOBJFILE	ZOBJID
ZOBJNAME	ZPERUID	ZPERGID
ZEDMTYPE		

Note: The ZRSCDATE and ZRSCTIME fields are not referenced, nor are they used to update the date/time of the file received. ZRSCSIZE and ZCMPSIZE are ignored also. However, these attributes might be used in the future.

EDMMRPRO

This method allows the adding, updating, and deleting of heaps in the PRIMARY database based on the contents of the parameter object that is passed. Each heap in the object specifies an instance to be added, updated, or deleted.

EDMMRPRO Parameter

object	The name of the in-storage object. The object can have multiple heaps where each heap in the object represents a CSDB instance to be altered.

Example

```
EDMLINK EDMMRPRO 'ANYOBJECT'
```

Return Codes

0	The method was successful.
>0	An error was detected, the method failed.

The instance indicates which CSDB instance will be altered by specifying five control variables:

ZOBJCLAS	Target class, for example, ZADMIN.
ZOBJDOMN	Target domain, for example, SYSTEM.
ZOBJFILE	Target file, for example, PRIMARY.
ZOBJNAME	Target instance. This can be any valid instance name.
ZOBJFILE	Target file, for example, PRIMARY.

- On a **delete** request, only the control variables are used to identify the instance to be deleted, the remaining variables are ignored.
- On **add** and **update** requests, the variables in each instance contain the values used to update the instance in the CSDB.
- The fields that can be updated are variables, class connections, expressions, and methods.

- There are specification differences for the three field types.
However, regardless of field type, the target instances' field lengths determine the amount of data moved, and length adjustment is performed, including blank padding and truncation.

Variables

Any variable name found in the parameter object and found in the target instance will be used to update the target instance. Any variable not found in the target instance will be ignored.

Method Fields

Method fields found in the parameter object and found in the target instance will be used to update the target instance. Any method not found in the target instance will be ignored.

Note: The string before the equals sign (=) must be eight bytes long.

The methods are indicated by variables in the parameter object that are named MTHDnnnn, where nnnn is 0001 to 9999. The value of the variable in the object should contain

methodfieldname = xxxxxxxxxx

where *methodfieldname* is used to identify the target method field. For example:

`_ALWAYS_=SYSTEM.METHOD.SIGNON_METHOD`

OR

`EDMSETUP=SYSTEM.METHOD.CHECK_APPL_STATUS.***`

Connection Fields

Class fields found in the parameter object and found in the target instance will be used to update the target instance. Any variable not found in the target instance will be ignored. The connections are indicated by variables in the parameter object that are named CONNnnnn, where nnnn is 0001 to 9999. Types of connection fields include CONNnnnn (connection), INCLnnnn (Includes), ALWAnnnn (Always), and REQUnnnn (Requires). The value of the variable in the object should contain

connectionfieldname = xxxxxxxxxx

where *connectionfieldname* is used to identify the target class field. For example,

`_ALWAYS_=SOFTWARE.ZSERVICE.MY_SERVICE.`

These variable names are the same format as object resolution with a message type = `_NONE_`. This is designed to enable the output of these resolutions (sometimes referred to as reporting resolutions) to be used unchanged as input to EDMMRPRO.

You can update specific target instances while not overwriting some existing values (for example, EDMSETUP=) via EDMMRPRO in one of two ways:

- Each class named by the CONNnnnn, the value of the variable in the object *connectionfieldname*=xxxxxxx, needs to be specified even if it is not the target of change and will be updated with the specified content. Each CONNnnnn needs to be specified for each variable in the sequence to provide a placeholder for updating the variables. For example, CONN0001 EDMSETUP=COUNTRY.USA_EAST_COAST

CONN0002 EDMSETUP=ZSERVICE.XYZ

CONN0003 EDMSETUP=ZSERVICE.ABC

- Another way of updating specific target instances while not overwriting some existing values is to specify a CSV (comma-separated variable) string for the instance. Empty values specified before a comma indicates that the connection should skip over the existing value and not update it. For example, the format for skipping over the first two variables and updating the third would appear as:
CONN0001 "EDMSETUP=, ,ZSERVICE.ABC"

Notes on EDMMRPRO Usage

- The file, domain, and class must already exist; EDMMRPRO will not add any of these levels dynamically. In addition, any fields being processed must already be defined in the target class; ZPROMANY will not modify classes.
- Different class instances can be altered during one execution of EDMMRPRO. However, it is not advisable to do so as certain field names might overlap (particularly methods and connections).
- Different databases cannot be altered in one execution of EDMMRPRO.

EDMMULOG

This method writes a message returned from the execution of a REXX method to a user log file. For this method to work, the MGR_USERLOG section must be added to the `edmprof` file, and `ACTIVATE=` must be YES.

EDMMULOG Parameter

msg	The message that will be written to the user log file. This message is returned by a method after its execution.

Example

```
ADDRESS EDMLINK "EDMMULOG" MSG
```

Return Codes

0	The method was successful.
8	An error was detected, the method failed.

Configuration Server EDMPROF File Sample

```
MGR_USERLOG]
ACTIVATE = YES
COLUMN_WIDTH = 128
DIRECTORY =
FLUSH_SIZE = 128
PIPE_SIZE = 100000
THRESHOLD = 500000
```

EDMSIGN

This method enables the Configuration Server to authenticate Sentient agent and Sentient administrator sessions against the CSDB. The password that is stored in the ZPWD variable in the specified object on the Sentient agent/administrator is compared to that which is stored in the user's profile in the CSDB.

- If the passwords match, the session continues.
- If the passwords do not match, the message "PASSWORD INVALID" is returned.

Changing Passwords

Passwords can be changed by either of the following methods.

- In the Sentient Admin Agent Explorer: specifying a new password in ZNEWPWD and the old password in ZPWD.
- On the Sentient CSDB Editor login panel: selecting the **Change Password** option, and specifying the password information.

Note: For information about Sentient agent and Sentient Administrator password authentication, see the *Sentient Enterprise Console Online Help*.

EDMSIGN Parameter

object	Specifies the name of the object from which the ZPWD variable is extracted. If no object name is specified, the ZMASTER object is used by default.

Example

```
REXX
```

```
EDMSIGN
```

```
(Uses the ZMASTER Object)
```

```
EDMSIGN & (ZCURRENT>ZCUROBJ)
```

Return Codes

0	The method was successful.
4	New user.
8	An error was detected, the method failed.

EDMSIGNR

This method enables the Configuration Server to authenticate sessions with Sentient agent and Sentient administrator against an external security system such as Windows security. The password that is stored in the ZPWD variable in the specified object on the Sentient

agent/administrator is compared to that which is stored in the native security system for that user ID.

- If the passwords match, the session continues.
- If the passwords do not match, the message “PASSWORD INVALID” is returned.

Changing Passwords

Passwords can be changed by either of the following methods.

- In the Sentient Admin Agent Explorer: specifying a new password in ZNEWPWD and the old password in ZPWD.
- On the Sentient CSDB Editor login panel: selecting the **Change Password** option, and specifying the password information.

Note: For information about Sentient agent and Sentient Administrator password authentication, see the *Sentient Enterprise Console Online Help*.

EDMSIGNR Parameter

object	Specifies the name of the object from which the ZPWD variable is extracted. If no object name is specified, the ZMASTER object is used by default.

Example

```
/*****REXX*****/  
  
EDMSIGNR  
  
(Uses the ZMASTER Object)  
  
EDMSIGNR & (ZCURRENT>ZCUROBJ)
```

Return Codes

0	The method was successful.
8	An error was detected, the method failed.

ZDCLASS

This method deletes a class and its associated instances from the database.

Note: ZDCLASS will not delete data-bearing instances.

ZDCLASS Parameters

domain	The 32-byte (maximum) name of the domain that houses the class to be deleted.
class	The eight-byte (maximum) name of the class to be deleted.
file	The file name that contains the class to be deleted. This parameter is optional.

Example

```
/***** REXX *****/  
  
DOMAIN = 'SOFTWARE';  
  
CLASS = 'TESTCLAS';  
  
PARM = SUBSTR(DOMAIN,1,8) || SUBSTR(CLASS,1,8);  
  
SAY 'PARM STRING IS ' PARM;  
  
ADDRESS EDMLINK ZDCLASS PARM;
```

Return Codes

0	The method was successful.
8	An error was detected, the method failed.

ZDELINS

This method displays or deletes (from the CSDB) an instance, or range of instances, within a class. It permits the use of wildcards (*).

Note: Displayed instances will be written to the Configuration Server log even if all other TRACE settings are OFF.

ZDELINS Parameters

Parameter	Description
file	The file that contains the instances to be displayed or deleted.
domain	The domain that contains the instances to be displayed or deleted.
class	The class that contains the instances to be displayed or deleted.
option	DISPLAY if instances are to be displayed. DELETE if instances are to be deleted.
frominst	The name or starting name of the instance to be deleted or displayed.
toinst	The instance to be deleted or displayed. Blanks in this field indicate that it is a single instance to display or delete, not a range.

Example

```
/***** REXX *****/  
  
FILE = 'PRIMARY'  
  
DOMAIN = 'SOFTWARE'  
  
CLASS = 'PACKAGE';  
  
FROMIN = 'TSO_ ';
```

```
TOINS = ' ' ;  
OPTION = 'DISPLAY';  
PARM = FILE||DOMAIN||CLASS||OPTION||FROMIN||TOINS;  
SAY 'PARM STRING IS 'PARM;  
ADDRESS EDMLINK ZDELINS PARM;
```

Return Codes

0	The method was successful.
8	An error was detected, the method failed.

ZDELOBS

This method deletes an in-storage object.

Note: This method supports the deletion of multiple objects in one call, without spawning additional processes.

ZDELOBS Parameters

object	The name of the in-storage object to be deleted.

Example

```
ADDRESS EDMLINK ZDELOBS 'ZTEST';
```

Return Codes

0	The method was successful.
8	An error was detected, the method failed.

ZDELPROF

This method deletes a class in the PROFILE File of the CSDB.

ZDELPROF Parameters

domain	The domain that contains the object to be deleted.
class	The class that contains the object to be deleted.

Example

```
EDMLINK ZDELPROF 'TESTP1,TESTPROF'
```

Return Codes

0	The method was successful.
8	An error was detected, the method failed.

ZEXIST

This method verifies the existence of classes and instances in the CSDB.

ZEXIST Parameters

file	The file that contains the class or instance to be verified.
domain	The domain that contains the class or instance to be verified.
type	The type of file.
class	The class that contains the instance or class record to be verified.
instance	The instance to be verified.

Example

```
FILE = 'PRIMARY'
DOMAIN = 'POLICY' ;
CLASS = 'USER' ;
INST = 'USER1' ;
PARM = FILE || '.' || DOMAIN || '.' || CLASS || '.' || INST ;

ADDRESS EDMLINK ZEXIST PARM ;

IF RC = 0 THEN

    SAY 'QAREXX ***** OBJECT ' INST ' EXISTS;
```

Return Codes

0	The method was successful.
8	An error was detected, the method failed.

ZGETPROF

This method accesses the PROFILE File of the CSDB, gets the *dbobject* object and puts it in storage creating an in-storage object, *inobject*. The domain in the PROFILE database is the USERID, ZUSERID, which is found in the current object or in the ZMASTER object. If ZUSERID is not found, the method will return an error message, “Profile error: user ID not found” in the log.

ZGETPROF Parameters

dbobject	The PROFILE File object name.
inobject	The name of the in-storage object to be created.

domain name	The name of the domain in the PROFILE File (usually the ZUSERID of the ZMASTER object).
instance	The name of the instance. This parameter is <i>optional</i> .

Example

```

/***** REXX *****/

  PARM='ZSTATUS,ZSTATUS,'ZMASTER.ZUSERID;

  ADDRESS EDMLINK ZGETPROF PARM;

/* GET OLD PROFILE.?.ZSTATUS */

```

Return Codes

0	The method was successful.
8	An error was detected, the method failed.

ZNFYT

This method enables you to initiate a PUSH notification (the execution of a program or programs on an Sentient agent from another location). To execute notify successfully, EDMEXECD must be running on the Sentient agents on which you are executing a PUSH.

ZNFYT Parameters

“process to run”	The application that you are forcing the Sentient agent to execute.
domain	The name of the domain where the notification results are stored. If not specified, this will be automatically generated as a function of date/time.
instance	The instance name containing the results of the single notification. If not specified, this will be automatically generated as an eight-digit number. The default is 00000001 . Note: Only numeric names 00000000 to 99999999 are valid. Specifying anything else will result in 00000000 replacing the erroneous name.
password	The ZNFYPWD for the target terminal's ZMASTER object.
port	The port on which the Sentient agent notify daemon is listening. This should be the same as the Sentient agent's ZMASTER.ZNTFPORT port number.
target IP address	The IP address of the Sentient agent device on which you are executing a PUSH.
user ID	The Sentient agent user ID.

Note: If the **domain** and **instance** parameters are omitted, the resulting instance will be written as:

```
NOTIFY.mmddyymmss.NOTIFY.00000001
```

This does not guarantee the uniqueness of the domain name. In addition, the instance name does not represent anything significant, other than sequence.

Example

```
/*trace i*/  
  
RC = EDMGET("ZNFYT",0)  
  
NHEAPS = ZNFYT  
  
DO CURRHEAP = 1 TO NHEAPS BY 1  
  
    RC = EDMGET("ZNFYT",CURRHEAP)  
  
    NIPADDR = ZNFYT.IPADDR  
  
    NPORT = ZNFYT.PORT  
  
    NUSER = ZNFYT.USER  
  
    NPASSW = ZNFYT.PASSW  
  
    NCMDLINE = strip(ZNFYT.CMDLINE)  
  
    NHANDLE = ZNFYT.HANDLE  
  
/*** CALL ZNFYT TO ISSUE THE NOTIFY ***/  
  
    ADDRESS EDMLINK "ZNFYT" NIPADDR || ',' || NPORT || ',' || NUSER ||  
' ,' || NPASSW || ',' || NCMDLINE || '"' || ',' NHANDLE || ',' ||  
CURRHEAP;  
  
END
```

Return Codes

0	The method was successful.
8	An error was detected, the method failed.

ZOBJCMPR

This method compresses an in-storage object.

ZOBJCMPR Parameter

object	The name of the in-storage object to be compressed.

Example

```
ADDRESS EDMLINK ZOBJCMPR 'ZTEST' ;
```

Return Codes

0	The method was successful.
8	An error was detected, the method failed.

ZOBJCOPY

This method copies an in-storage object. The resulting object has the same variables and number of heaps as the original object.

ZOBJCOPY Parameters

fromobject	The name of the existing in-storage object to be copied.
toobject	The name of the in-storage object to be created.

Example

```
ADDRESS EDMLINK ZOBJCOPY 'OBJECT1,OBJECT2' ;
```

Return Codes

0	The method was successful.
8	An error was detected, the method failed.

ZOBJDELI

This method deletes a heap of an in-storage object.

ZOBJDELI Parameters

object	The name of the in-storage object from which to delete a heap.
instance#	The heap number to delete.

Example

```
/***** REXX *****/  
DPARM = 'TESTOBJ, '||1|| ' ' ;  
ADDRESS EDMLINK ZOBJDELI DPARM;
```

Return Codes

0	The method was successful.
8	An error was detected, the method failed.

ZOBJDELV

This method deletes a variable from all heaps of an in-storage object. It verifies the existence of the specified object and finds the specified variable in that in-storage object. The variable value is then removed from each heap of the in-storage object.

ZOBJDELV Parameters

object	The object that contains the variable to be deleted.
variable	The name of the variable to be deleted.

Example

```
/***** REXX *****/  
ADDRESS EDMLINK ZOBJDELV 'TESTOBJ,VAR00001' ;  
SAY 'QAREXX ***** VAR00001 DELETED FROM OBJECT TESTOBJ' ;
```

Return Codes

0	The method was successful.
8	An error was detected, the method failed.

ZOBSORT

This method sorts the heaps of an in-storage object by the values of specified variables and according to the desired collating sequence.

ZOBSORT Parameters

sort sequence	Ascending (SORT) or descending (SORTD).
object	The name of the object to be sorted.
variable	Up to three variable names can be specified.

Example

```
PARM1 = 'SORT, '  
PARM2 = 'ZSERVICE'  
PARM3 = ', ZOBJNAME, ZOBJDATE, ZOBJTIME ' ;  
PARM = PARM1 || PARM2 || PARM3 ;  
ADDRESS EDMLINK ZOBSORT PARM ;
```

Return Codes

0	The method was successful.
8	An error was detected, the method failed.

ZPROMANY

This method allows the addition and updating of instances in the PRIMARY database, based on the contents of the parameter object that was passed. Each heap in the object specifies an instance to be added or updated.

Note: ZPROMANY requires that spaces be entered after commas to blank out existing services in fields on the CSDB.

ZPROMANY Parameter

object	The name of the in-storage object.

Example

```
EDMLINK ZPROMANY 'ZANYOBJECT'
```

ZPUTHIST

This method puts an in-storage object into the HISTORY file. It takes the *inobject* that is in storage and puts it in the HISTORY file as *dbobject*. The domain used in the HISTORY file is the DATE/TIME stamp found in current object, or in ZMASTER object.

ZPUTHIST Parameters

dbobject	The object name that will be put in the HISTORY file. This parameter is optional.
inobject	The object name of the in-storage object.

Example

```
ADDRESS EDMLINK ZPUTHIST 'ZCOMPARE,ZCOMPARE';
```

```
ADDRESS EDMLINK ZPUTHIST 'ZSTATUS';
```

Return Codes

0	The method was successful.
8	An error was detected, the method failed.

ZPUTPROF

This method puts an in-storage object (inobject) into the PROFILE File of the CSDB, as dbobject. The domain in the PROFILE File is the ZUSERID found in the inobject, or in the ZMASTER object. If ZUSERID is not found, the dbobject is put in the domain, _UNKNOWN.

Note: The ZPUTPROF method allows you to specify multiple objects simultaneously.

ZPUTPROF Parameters

dbobject	The object name that will be put into the PROFILE database. The default is the object name.
domainid	The value of an optional third operand can be used to specify a domain other than ZUSERID or to eliminate the search for a ZUSERID value.
inobject	The name of the in-storage object.

Example

```
/***** REXX *****/  
ADDRESS EDMLINK ZPUTPROF 'OBJECTS=ZMASTER,ZCONFIG,ZUSERID'
```

Return Codes

0	The method was successful.
8	An error was detected, the method failed.

ZSIMRESO

This method resolves the CSDB instance specified by the parameter string, and the resulting objects are left in storage. Any prerequisite objects needed for a successful resolution must already be built and in storage. For example, to resolve:

USER.&ZUSERID

an object containing the ZUSERID variable might have to be constructed. Otherwise, the resolution might not be completely successful.

ZSIMRESO Parameters

file	The file that contains the instance to resolve.
domain	The domain that contains the instance to resolve.
class	The class that contains the instance to resolve.
instance	The instance to resolve.
message	This specifies the message for conditional resolution paths.

Example

```
ADDRESS EDMLINK ZSIMRESO PRIMARY,POLICY,USER,USER1,EDMSETUP
```

Return Codes

0	The method was successful.
8	An error was detected, the method failed.

ZTOUCH

This method updates the date/time stamp of an instance in the CSDB.

ZTOUCH Parameters

class	The name of the class that contains the instance to be updated.
domain	The name of the domain that contains the instance to be updated.
instance	The name of the instance to be updated.

Example

```
DOMAIN = "SOFTWARE";  
CLASS = "ZSERVICE";  
INST = "TEST_OBJECT";  
PARM = SUBSTR(DOMAIN,1,8) || SUBSTR(CLASS,1,8) || SUBSTR(INST,1,32);  
  
ADDRESS EDMLINK ZTOUCH PARM ;
```

Return Codes

0	The method was successful.
8	An error was detected, the method failed.

ZVARDEL

This method deletes all in-storage objects. There are no parameters associated with ZVARDEL.

Example

```
ADDRESS EDMLINK "ZVARDEL"
```

Return Codes

0	The method was successful.
8	An error was detected, the method failed.

ZVARGBL

This method migrates values from one in-storage object to another, and then deletes the source object.

ZVARGBL Parameters

destination	Object to which the values are being migrated.
source	Object from which the values are being migrated.

Note: Only variables with appropriate flag settings will be migrated.

Example

```
EDMLINK ZVARGBL 'TESTSORT,TESTVGBL'
```

Return Codes

0	The method was successful.
8	An error was detected, the method failed.

ZVARLOG

This method writes the contents of an in-storage object to the Configuration Server log.

Note: EDMMOLOG will write to the Configuration Server log even if all other TRACE settings are OFF.

ZVARLOG Parameter

object	The name of the in-storage object to be displayed.

Example

```
ADDRESS EDMLINK ZVARLOG 'ZMASTER';
```

Return Codes

0	The method was successful.
8	An error was detected, the method failed.

ZUPDPROF

This method only updates profile information. It does not perform any type of deletion.

- If toobject is not specified, the fromobject (source) name will be used.
- If user ID is not specified, the current user ID will be used.

ZUPDPROF Parameters

fromobject	The source object name.
toobject	The destination object name.
user ID	The user ID.

Example

```
ADDRESS ZUPDPROF OBJECT1,OBJECT
```

Return Codes

0	The method was successful.
8	An error was detected, the method failed.

ZXREF

ZXREF cross references class and instance usage during object resolution, and collects information on the cross-referenced objects. It will generate objects that enable administrators to cross reference users with any, and all, Departments, Workgroups, and Services with which they are affiliated (connected).

To implement the ZXREF method:

1. Add new method instances to the METHOD class. Some are methods to create the objects from the PRIMARY database and others are to write these objects to the PROFILE database.
2. Update the Configuration Server process (ZMASTER) used during the Sentient agent connect process by adding new methods to SYSTEM.PROCESS.ZMASTER.
3. Update your class templates, if necessary, to add new method attributes.
4. Update the `_BASE_INSTANCE_` to specify the method instances to be executed.

ZXREF Parameter

object	The name of the object containing the cross referenced information.

Return Codes

0	The method was successful.
8	An error was detected, the method failed.

EDM Access Method Services (EDMAMS)

Caution: It is recommended to shut down the Configuration Server to ensure that the CSDB contents are not changing during the execution of the EDMAMS utilities. However, ZEDMAMS can be run as a method, in which case the Configuration Server must be running. In addition, it is recommended to back up the CSDB before running any of the EDMAMS utilities that will update it.

Overview

EDMAMS is a core set of utilities that can be used to create, delete, copy, change, and list objects in the Configuration Server Database. The functionality of these utilities is invoked using one of the EDMAMS verbs, which are called by the module, ZEDMAMS. The ZEDMAMS module is located in the directory in which the Configuration Server was installed.

- On a Windows system, the default is:
`<InstallDir>\ConfigurationServer\bin`

Terminology

The following table lists the terms that will be encountered in this chapter.

EDMAMS Verb Terminology

Term	Definition
Verb	The action to be performed on the database object. For example: <code>DELETE_INSTANCE</code> , <code>EXPORT_CLASS</code> , and <code>SYNC_CLASS</code> .
Keyword	The (required and optional) predefined database location or object designators, such as: <code>FILE</code> , <code>DOMAIN</code> , <code>FROMDOMA</code> , and <code>TOINST</code> . Also, a predefined instruction to be considered during the action. For example: <code>KEEPDATE</code> , <code>PREVIEW</code> , and <code>HEADER</code> .
Value	The user-specified database location or object on which the verb will act, such as: <code>POLICY</code> , <code>USER</code> , <code>RAD*</code> , and <code>EXPC.DAT</code> . Also, a user-specified consideration that accompanies the action. For example: Retaining an object's creation date and time (<code>KEEPDATE=YES</code>), Applying a comment to a copied object (<code>COMMENT=copied_object</code>), and Replacing all existing data in the target object (<code>REPLACE=YES</code>)
Unmated Instance	An instance that does not have a resource.
Orphaned Resource	A resource that does not have a parent instance.
Component Orphans	A component instance that does not have a parent (for example, a package instance).

Invoking the EDMAMS Verbs

For a list of the EDMAMS verbs that are available with the version of the CSDB running in an environment, on the command line, type:

```
ZEDMAMS VERB=
```

Note: The ZEDMAMS module is subject to continuous development. Due to this ongoing development, some of the verbs detailed in this chapter might not be applicable to the CSDB version that is running in your environment. Subsequent printings of this guide will document enhancements to the EDMAMS functionality.

To display a verb's (required and optional) keywords, type:

```
ZEDMAMS VERB=VERB_NAME
```

For example, typing,

```
ZEDMAMS VERB=DELETE_INSTANCE
```

on the command line shows the results of invoking the verb, DELETE_INSTANCE. The keywords are listed, as well as an explanation of each, with examples and applicable defaults. The keywords in parentheses are optional.

For more information on using the verbs, see the sections, [""Using the EDMAMS Verbs" below"](#) and [""Usage Considerations" on page 651](#)

Using the EDMAMS Verbs

All of the EDMAMS verbs are specified in the following format:

```
ZEDMAMS VERB=VERB_NAME, KEYWORD=VALUE, KEYWORD=VALUE
```

For example,

```
ZEDMAMS VERB=DELETE_INSTANCE, CLASS=USER, INSTANCE=SALES
```

Note: There are no rules governing the order in which the keyword-value combinations are specified.

Usage Considerations

This section addresses some of the more noticeable and critical effects that might result from using this verb, as well as some of the new features.

- After the data from an import deck has been written to the CSDB, the changes are considered permanent. Therefore, it is imperative that a CSDB administrator using this verb be certain of the changes that are being considered.

Note: It is recommended to
Shut down the Configuration Server to ensure that the CSDB contents are not changing during the processing.
Back up the CSDB before running this verb.
Specify PREVIEW=YES and check the resulting log before committing any changes to the database.

- Executing this verb might result in the creation of new decks (**MPI/MPR**). The circumstances under which this might happen are:
 - There exist duplicate object IDs (ZOBJID) in the database instances and the import deck instances, and DUPLICATES=MANAGE.
 - There is a domain name change for the imported data (using the MAP_DOMAIN keyword).
 - The ZOBJRCRC (the object resource CRC) is NULL or empty and the value can be calculated and assigned in the process.

Note: All of these occur in conjunction with the existence of the XPR deck, it being specified on the command line, and the values of the CONTINUE and REPLACE allowing the processing to continue. The XPR deck is necessary so that if changes are required, it is available to be updated at that time.

- This version of IMPORT_INSTANCE has more consistency checks that must be passed before the import deck is considered valid for import.
- Combining instances and resources on the same command line allows the import deck to be more thoroughly examined.
- Use the CONTINUE option to prevent data loss during import.
- In this version, an administrator:
 - Can manage the processing behavior if an import deck OID collides with a database OID.
 - Can specify the filename prefix if a new deck is generated.
 - Has a single keyword to facilitate changing domains.
- A new time-based format of object ID generation has been introduced, thereby eliminating the chance of randomly generating a duplicate OID. For more information on this feature, see the ["MGR_STARTUP" on page 547](#).

Input Files

Another way to run the EDMAMS utilities is to contain the commands in an input file—a file that has been edited by a text editor. This file can then be run to successively execute several EDMAMS verb functions.

- In an input file, the keywords can be specified on one or more lines for a single function.
- Like the command line method, a comma must follow each keyword-value combination, with the last keyword-value combination followed by a space.

EXPORT Verbs

Input files are very useful when using the exporting EDMAMS verbs (EXPORT_CLASS, EXPORT_INSTANCE, and EXPORT_RESOURCE) because they enable multiple domain-class combinations to be exported during a single export function.

The section, ["Internationalization Consideration for Exporting/Importing Database Decks" on page 622](#) contains important information regarding the use of the EXPORT verbs.

Multiple Verbs

More than one verb can be specified in an input file. For example, the COPY_DOMAIN verb can be followed by the verbs, DELETE_CLASS and LIST_CONS_VARS. Any combination of verbs can be included in one run, as long as the data sets being accessed do not conflict. If an asterisk (*) is placed in the first column, it is considered a comment and no action is taken.

To execute commands that are contained in an input file, enter the following on the command line:

```
ZEDMAMS ZFILE "DRIVE:\FILE_PATH\FILE_NAME"
```

Note: ZEDMAMS is the executable. ZFILE is the second argument and must be in uppercase.

Example of multiple VERBS executed from one file:

```
VERB=COPY_DOMAIN, FROMDOMA=POLICY, TODOMAIN=SYSTEMA, REPLACE=NO
```

*

```
VERB=DELETE_CLASS, DOMAIN=SYSTEMA, CLASS=TESTCLAS
```

*

```
VERB=LIST_CONS_VARS, DOMAIN=SOFTWARE, CLASS=FILE, INSTANCE=*
```

Wildcards

EDMAMS supports two types of wildcards: *implicit* and *explicit*.

- **Implicit wildcards**

are available for the COPY_INSTANCE, DELETE_INSTANCE, and LIST_INSTANCE verbs. Specify any portion of the value to select all occurrences that contain that portion of the value. Implicit wildcards do not require an asterisk, and are expressed as follows:

KEYWORD=<wildcard_string>.

For example, specify FROMINST=RAD to include all the fields that contain RAD as any part of the string.

- **Explicit wildcards**

are available for the CHANGE_INS_FIELD, COPY_NEW_SUFFIX, COPY_RESOURCE, LIST_CONS_VARS, LIST_INST_DATA, LIST_ZRSC_FIELDS, and SEARCH_INSTANCES verbs. Explicit wildcards require an asterisk, and are expressed as follows:

KEYWORD=<wildcard_string>*.

For example, specify FROMINST=RAD* to select all the instances that contain RAD as the first part of the string.

LOGFILE

This keyword can be used with all of the EDMAMS verbs. Specify the fully qualified path to, and name of, the log file to which the information is to be reported. The default log file name is

ZEDMAMS.LOG

If this keyword is not specified, or specified without a value, the default location on windows is assumed the `bin` folder in which the `zedmams.exe` utility is running.

To send the information to a location\file other than the default, specify:

`LOGFILE=FILE_PATH\FILE_NAME`

Note: It is recommended that if LOGFILE is specified as a path other than the default, that its existence be verified. If the directory does not exist, the log creation process will fail, and the command will not execute.

Internationalization Consideration for Exporting/Importing Database Decks

This section details the conditions under which database decks can be exported/imported using the ZEDMAMS EXPORT and IMPORT verbs. "Codepage and Locale Defaults" on the next page, in the section, "Codepage and Locale Defaults" on the next page, is a supplement to this information; it lists the defaults of the new keywords for the EXPORT and IMPORT verbs.

EXPORT_CLASS, EXPORT_INSTANCE, and EXPORT_RESOURCE

Source Database Format	Acceptable Target Database Formats
UTF-8	UTF-8 and Basic ASCII
Other Locale or Codepage	UTF-8
Basic ASCII	UTF-8 and Basic ASCII
Extended ASCII	UTF-8 and Extended ASCII

EXPORT Verb Considerations

- If the deck is being exported from a database with a *locale* or *codepage* that differs from the database into which it is being imported, the appropriate keyword (FROM_LOCALE or FROM_CODEPAGE) must be specified.
If the source database's *locale* or *codepage* is the same as the database into which it is being imported, neither keyword needs to be specified.
- If the deck is being exported from a database with a *locale* or *codepage* that differs from the database into which it is being imported, the appropriate keyword (TO_LOCALE or TO_CODEPAGE) must be specified.
If the source database's *locale* or *codepage* is the same as the database into which it is being imported, neither keyword needs to be specified.
- If TO_LOCALE=LEGACY is specified, XPR headers will be automatically converted to EBCDIC.
- If TO_LOCALE=UTF8 is specified, an internationalized ZEDMAMS UTF-8 export deck is produced.

IMPORT_CLASS, IMPORT_INSTANCE, and IMPORT_RESOURCE

Target Database Format	Source Database Format Must Be

UTF-8	UTF-8 or Basic ASCII or Extended ASCII
Other Locale or Codepage	UTF-8
Basic ASCII	Basic ASCII
Extended ASCII	Extended ASCII

IMPORT Verb Considerations

- If the deck is being imported into a *locale* or *codepage* that differs from the database from which it was exported, the appropriate keyword (TO_LOCALE or TO_CODEPAGE) must be specified.
If the target *locale* or *codepage* is the same as the database from which the deck was exported, neither keyword needs to be specified.
- If the deck is being imported into a *locale* or *codepage* that differs from the database from which it was exported, the appropriate keyword (FROM_LOCALE or FROM_CODEPAGE) must be specified.
If the target *locale* or *codepage* is the same as the database from which the deck was exported, neither keyword needs to be specified.
- Translated data must fit the field size limits.
- Legacy XPR headers in EBCDIC are automatically converted to ASCII.

Codepage and Locale Defaults

The following table lists the defaults for the IMPORT and EXPORT *codepage* and *locale* keywords.

Codepage and Locale Defaults

Keywords	Default
FROM_CODEPAGE	If the deck is encoded in UTF8, the default is UTF8 codepage 65001 ; otherwise, the default is the current system codepage.
TO_CODEPAGE	If the database is encoded in UTF8, the default is UTF8 codepage 65001 ; otherwise, the default is the current system codepage.
FROM_LOCALE	If the deck is encoded in UTF8, the default is Locale country_region.UTF8 ; otherwise, the default is the current process locale.
TO_LOCALE	If the database is encoded in UTF8, the default is Locale country_region.UTF8 ; otherwise, the default is the current process locale.

Note: If the command line options are omitted, ZEDMAMS will open the import decks, look for the indicator and, if not found, will assume locale code page; it will then examine the database and look for the indicator and, if found, will treat as UTF8, but if not found, will treat as locale code page.

ZEDMAMS Utility

"ZEDMAMS Utility" [above](#) lists and describes the verbs for the ZEDMAMS utility.

ZEDMAMS Verbs

Verb	Description
ADD_FIELD	Adds a variable at the end of a template, including an automatic README file.
BUILD_PATCH	Builds a PATCH file in a CSDB, or a file from two Configuration Server Databases.
BUILD_STAGING_POINT	Creates a staging point, as needed, on the hard drive, and enables the resources (represented as an export deck) to be converted into a tree of files located in a stager style directory.
CHANGE_FIELDNAME	Changes variable names in class templates.
CHANGE_FLD_VALUE	Changes a template's variable length, type, Configuration Server, and Sentient agent flags.
CHANGE_INST_DATA	Globally changes data in instance records, by class.
CHANGE_INS_FIELD	Changes one field in each instance of a class and verifies connects.
CHECK_RESOURCES	Verifies the size of resources against ZRSCSIZE and ZCMPSIZE.
CLONE_INSTANCE	Clones an instance with a four-digit suffix (max. 2000).
COPY_CLASS	Copies a class, its instances, and, if it exists, its resource data.
COPY_DATA	Copies only resource data from one <i>domain.class.instance</i> to another.
COPY_DOMAIN	Copies domains.
COPY_FIELD	Copies attribute data to a new attribute or to an existing attribute.
COPY_INSTANCE	Copies an instance or a range of instances and associated resource records.
COPY_NEW_SUFFIX	Copies and renames the suffixes for the FILE class and its mated instances.
COPY_RESOURCE	See COPY_INSTANCE.

CREATE_ INSTANCES	Writes and populates instances from data in an edited text file.
DELETE_ CLASS	Deletes a class and its instances.
DELETE_ COMP_ORPHS	Deletes component orphans from the PACKAGE class.
DELETE_ DOMAIN	Deletes one, or a range of, domains.
DELETE_ FIELD	Deletes an attribute from a template.
DELETE_ INSTANCE	Deletes/displays a range of instances within a class and, if it exists, its resource data.
DELETE_ ORPHANS	Deletes unmated resource records (orphans) from the RESOURCE file.
EDIT_CLASS_ PREFIX	Updates the fields in the first 60 bytes of the prefix.
EXPORT_ CLASS	Exports classes to an output file or data set for import to another file or data set.
EXPORT_ INSTANCE	Exports instances to an output file or data set for import to another file or data set for reporting.
EXPORT_ RESOURCE	Exports resource data to an output file or data set for import to another file or data set.
IMPORT_ CLASS	Imports classes from an exported output file or data set to a PRIMARY file specified in the <code>edmp_{prof}</code> file.
IMPORT_ INSTANCE	Imports instances from an exported output file or data set to a PRIMARY file specified in the <code>edmp_{prof}</code> file.
IMPORT_ RESOURCE	Imports resource data from an exported output file or data set to a RESOURCE file specified in the <code>edmp_{prof}</code> file.
LIST_ CLASSES	Displays a list of classes with the class names, object IDs, ZOBJDATE, ZOBJTIME, persistence flag, sequence sensitive flag, Distributed Configuration Server flag, database type, and count totals. Note: If DOMAIN=* is specified, all classes in the domain will be listed.
LIST_CONS_ VARS	Displays connect and variable field (types C and V) values from instance records and stores output in <code>zedmams.log</code> .
LIST_ DOMAINS	Display a list of domains in the log of a specified file.
LIST_FLAGS	Lists Configuration Server and Sentient agent flags in selected class record.

LIST_INST_DATA	Lists instance data including variable names to <code>zedmams.log</code> .
LIST_INSTANCE	Lists instance names by domain, class, and instance (prefix).
LIST_PACKAGE	List PACKAGE class instances and all mated components.
LIST_PREFIX	Lists the Distributed Configuration Servers prefix by file, domain, and class.
LIST_RESOURCES	Lists resource prefix information (promote date, instance, and so forth).
LIST_ZRSC_FIELDS	Lists field values of fields beginning with ZRSC.
MATCH_RESOURCES	Matches resource records against data-bearing instances for the specified class.
PACKAGE_UNMATES	List all PACKAGE class instances without mated components.
REFRESH_DMA	Recounts the instances and classes in a file and updates the Distributed Configuration Server prefix.
RENAME_INSTANCE	Renames or displays instances by full name or prefix.
SEARCH_INSTANCES	Searches selected instances by domain and class for specified string
SORT_OBJECT_ID	Sorts object IDs in ascending or descending order by file or domain.
SYNC_CLASS	Synchronizes an existing class with a newly formatted class, and re-organizes all (existing class) instances according to the mapping in the newly formatted class. All existing class attributes that have a match in the new template will adopt the characteristics of the new template attribute, whereas any existing class attributes that do not have a match in the new template will be deleted.
UPDATE_INSTANCES	Updates the existing instance fields from data in an edited text file.
UPDATE_MGRIDS	Updates the Configuration Server ID and name by file, domain, and class.
VERIFY_CLASS	Verifies class templates for gaps, overlaps, and other anomalies.
ZRSOURCE_UNMATES	Matches data-bearing instances with the appropriate resources for the specified class.

The sections that follow describe each ZEDMAMS verb.

Note: In the **Syntax** for each verb, the default values are presented in **bold**.

ADD_FIELD

This verb adds a variable (attribute) to the end of a class template and unconditionally includes a README attribute.

- The README attribute is 35 bytes long.
- The LENGTH of the specified attribute cannot be greater than 255 bytes, or less than one byte.
- The Configuration Server and Sentient agent flags (MFLAGS and CFLAGS) are optional and will default according to the attribute TYPE. The default flags are presented when requesting a usage display.
- The FLDNAME can be any one- to eight-byte name and can be changed with the CHANGE_FIELDNAME function.
- INDEX adds the field 'before' the *n*th (INDEX=*n*) occurrence of the same attribute name. For example, on an _ALWAYS_ connection, if 10 _ALWAYS_ connections are present, an INDEX =1 adds the attribute to the beginning, an INDEX = 5 adds the attribute at the 5th spot, and an INDEX >= 11 adds the attribute as the last _ALWAYS_. When an attribute is added to the end, it is placed below the last attribute of the same type. However, if no INDEX is specified, the attribute is added at the end of all attributes.

Syntax:	(FILE=,) DOMAIN=, CLASS=, FLDNAME=, LENGTH=, TYPE=(, MFLAGS=) (, CFLAGS=) (, KEEPDATE=YES/ NO) (, README=) (, DEFAULT=) (, INDEX= <i>n</i>)
Example:	Add an M type attribute, NEWFIELD, to the specified class, giving it a length of 165 bytes, assigning default Configuration Server and Sentient agent flags, and updating the object date and time: DOMAIN=SOFTWARE, CLASS=USER, FLDNAME=NEWFIELD, LENGTH=165, TYPE=M, INDEX=5
Tip:	Run LIST_FLAGS against the class before and after running this to see the old and new template.

BUILD_PATCH

The functionality of this verb has been replaced with the **Service Optimization** feature of the Sentient CSDB Editor. For more information, refer to the *Sentient Enterprise Console Online Help*.

BUILD_STAGING_POINT

This verb enables resources (represented as an export deck) to be converted into a tree of files located in a stager style directory. It creates, on the hard drive, a staging point at a location

designated by OUTFILE, then the data can be transferred to a CD using standard CD-writing software. This verb checks for the staging point and, if it does not exist, creates it.

- Specify PREVIEW=YES to see the expected results of running the verb with specific parameters.
- INFILE is the fully qualified path and filename of an exported resource (XPR) file.
- OUTFILE is the destination directory location of the staging files.
OUTFILE defaults to the *location_of_edmprof\staging_point*. Therefore, if edmprof is located in C:\Temp\configserver\bin, the default staging point is C:\Temp\configserver\bin\staging_point.
If PREVIEW=YES, OUTFILE is ignored.
- XPI is an exported instance deck that is used to timestamp the staged resources.
This setting's value must be specified as the fully qualified path and filename of an exported instance deck.
If XPI is not specified, or if the specified deck does not contain instance data for the staged resource, the timestamp will be determined using the date and time from the input deck resource header.
- REPLACE=YES replaces the existing same-named file at the specified staging point.
- MULTICAST=YES specifies that the OUTFILE will contain the name of a directory where resource data will be stored in File.Domain.Class.Instance format, and each file will contain an embedded 60-byte prefix, identical to the prefix saved in the RESOURCE file.

Syntax:	(PREVIEW=YES/NO,) INFILE=(, OUTFILE=(, XPI=(, REPLACE=YES/NO) (, MULTICAST=YES/NO)
Example:	Accept the default staging point on the hard drive as the destination for files from C:\NovaFiles\NewCD: INFILE=C:\NovaFiles\NewCD
Tip:	To see what would be the result of running the verb (without actually doing so), specify PREVIEW=YES.

Resource Naming

BUILD_STAGING_POINT will remove the 60-byte prefix from each resource that is delivered to the staging point, and rename it according to the following format:

1. The first byte of the object ID is added to the **000** string, and used as a *branch root*.
2. Bytes 2 – 9 will be used as a *name*, and followed by a period.
3. The last three bytes will be used as an *extension*.
For example, a resource with an object ID of
DABC12345678
will be renamed:
000D\ABC12345.678.

In accordance with step 1, the first byte (**D**) was added to the string **000**. The back slash (\) was automatically inserted to make **000D** a branch root. A period was placed in front of the final three bytes (**678**), making them the extension, as described in step 3. The bytes in between the branch root and the extension (**ABC12345**) become the name.

CHANGE_FIELDNAME

This verb changes the specified class template field name (FROMNAME) to a new field name (TONAME).

- If there are multiple occurrences of a field name (such as, README), all of them will be changed.
However, README names should not be changed; only user-created variable field names should be changed.

Syntax:	(FILE=,) DOMAIN=, CLASS=, FROMNAME=, TONAME= (, KEEPDATE=YES/NO)
Example:	Change the template field name in the specified class from GROUPID to GROUP: DOMAIN=SOFTWARE , CLASS=USER , FROMNAME=GROUPID , TONAME=GROUP
Tip:	Run LIST_INST_DATA to display template field names with INSTANCE=_BASE.

CHANGE_FLD_VALUE

This verb changes a variable's length, type, and Configuration Server and Sentient agent flags by FLDNAME.

- The LENGTH must be in the range of 1 to 255 bytes.
- The Configuration Server (Manager) and Sentient agent flags (MFLAGS and CFLAGS) are optional, and will retain their current settings if omitted.
- The keyword DESC changes the DESCRIPTION field in the template for a maximum of 20 bytes. The DESCRIPTION field can be changed either alone or in addition to other fields. If blanks are included in the text, it must be enclosed in quotation marks (" ").
- The keyword TYPE is required except when DESC is the only field being changed. The values are:
 - For connect types – the characters **C**, **A**, **I**, **R**, and **O**.
 - For method types – the characters **M**, **T**, **H**, and **D**.
 - For variable types – the characters **V**, **U**, and **W**.
- INDEX changes the nth occurrence of variable FLDNAME. When omitted, the first variable with a matching fieldname will be changed.
- KEEPDATE=YES will prevent the OBJDATE and OBJTIME from being updated.
- README is the 1- to 35-byte description of the changed field that was placed in the _BASE_INSTANCE_. If blanks are included in the text, it must be enclosed in quotation marks (" ").
- DEFAULT the length of the default data must be less than or equal to the length specified by LENGTH.
This value populates FLDNAME in the _BASE_INSTANCE_. If blanks are included in the text, it must be enclosed in quotation marks (" ").

Syntax:	(FILE=,) DOMAIN=, CLASS=(, DESC=,) FLDNAME=(, LENGTH=,) TYPE=(, MFLAGS=) (, CFLAGS=) (, INDEX=) (, KEEPDATE=YES/ NO) (, README=) (, DEFAULT=)
Example:	In the template specified, change the TYPE value of the third attribute (named EDMSETUP) from V to C. (The length, and Configuration Server and Sentient agent flags retain their current value.): DOMAIN=SOFTWARE, CLASS=ZSERVICE, FLDNAME=EDMSETUP, TYPE=C, INDEX=3
Tip:	Run LIST_FLAGS against the class before and after running this to see the old and new template.

CHANGE_INST_DATA

This verb changes instance data.

- A match occurs only when the value of FROMDATA begins the instance field and is not embedded, or is not part of other data in the field.
- If PREVIEW=YES, only the instances to be changed will be displayed.
- If FIELD is specified, this verb will change all instances whose FROMDATA criteria is equal to the data portion of the heap specified by FIELD.
- FROMDATA is the data, in the heap, that is to be replaced.
All instances that meet the criteria of FIELD and FROMDATA will be changed with TODATA.
If FROMDATA=BLANKS, the change criteria will be all blank bytes.
- Omitting TODATA will set the field to blanks.
- FROMDATA and TODATA can be entered in any case. Keep in mind that the comparison made against FROMDATA is based on the case entered.
- If either FROMDATA or TODATA contain embedded spaces, the string must be enclosed in quotation marks.
- If KEEPDATE=NO, a new ZOBJDATE and ZOBJTIME are generated.

Syntax:	(FILE=,) DOMAIN=, CLASS=(, FIELD=,) FROMDATA=(, TODATA=) (, PREVIEW= YES/NO) (, KEEPDATE=YES/NO)
Example:	Change all instance data in the specified class from <i>JohnPublic</i> to <i>John Q. Doe</i> : DOMAIN=SOFTWARE, CLASS=USER, FROMDATA=JohnPublic, TODATA="John Q. Doe"
Tips:	Run LIST_INST_DATA to get a display of instance data and the class field names to which the data belongs. Run first with PREVIEW=YES to display the current variable data values.

CHANGE_INS_FIELD

This verb changes the instance data of the specified field in specific instances within the same domain.

- INDEX is the relative number (1–99) of multiple same named fields (such as, EDMSETUP or README).
- The keyword PREFIX:
 - Can be specified wildcards (*). For example, DIFF* and DIFF*SOL*.
 - If specified with just an asterisk (*), all instances in the specified class are changed.
 - To change only one instance, the entire name must be specified.
- The keyword FIELD is the name of the attribute to be changed.
- VERIFY=YES verifies that a connection value in TODATA exists. If the verify fails, the program ends.
- KEEPDATE=YES will prevent the OBJDATE and OBJTIME from being updated.
- TODATA can be entered in any case; the data is placed in the field as entered. If TODATA contains embedded spaces, the string must be enclosed in quotation marks. Specify only CLASS.INSTANCE, not the domain.
- PREVIEW=YES displays instance names and the current data before the change and PREVIEW=NO displays instance names and the current data after the change.

Syntax:	(FILE=,) DOMAIN=, CLASS= (, PREVIEW= YES /NO,) PREFIX=, FIELD=, TODATA= (, INDEX= (, VERIFY=YES/ NO) (, KEEPDATE=YES/ NO)
Example:	Change the third EDMSETUP field (INDEX=3) in the specified instances to ZLOCMGR.RAD_RESOURCE_FILE. Verify the existence of the connection, and update the OBJDATE and OBJTIME: DOMAIN=SOFTWARE, CLASS=USER, PREFIX=TSO, FIELD=EDMSETUP, TODATA=ZLOCMGR.RAD_RESOURCE_FILE, VERIFY=YES, INDEX=3
Tip:	This verb enables wildcards for the PREFIX parameter. Therefore, you can specify PREFIX=RAD* to see all the prefixes that contain RAD as the first part of the string.

CHECK_RESOURCES

This verb verifies the actual size of the resource data against ZRSCSIZE or ZCMPSIZE.

The entire PRIMARY file is checked for the presence of the variable field names ZRSCSIZE and ZCMPSIZE. If these fields exist, and if ZCMPSIZE contains a value, it is compared against the actual size. If ZCMPSIZE is zero or blank, the value in ZRSCSIZE is used. If there is a mismatch,

the name of the resource and the appropriate sizes are listed to the log and a return code of 8 is passed.

- If LISTALL=YES, all objects checked are listed, and objects with resources are listed with their respective sizes.
- If UPDATE=YES, and if the appropriate of either ZRSCSIZE or ZCMPSIZE does not contain the correct value, the field will be updated with the correct value. In addition, the 8-byte, printable hex field in the instance prefix will be updated if in error.
- CRC is a toggle to enable/disable the Cyclical Redundancy Check.
If CRC=YES, a CRC is calculated for each resource and if UPDATE=YES, incorrect CRCs are updated to their correct values in the OBJRCRC field.
- DOMAIN is a 1- to 32-byte (Domain) name that is to have its resources checked.
- CLASS is a 1- to 8-byte (class) name that is to have its resources checked.
If specified, DOMAIN must be specified.
- INSTANCE is a 1- to 32-byte (Instance) name that is to have its resources checked.
If specified, DOMAIN and CLASS must be specified.

Syntax:	LISTALL=YES/NO (, UPDATE=YES/NO) (, CRC=YES/NO ,) DOMAIN= , CLASS= , INSTANCE=
Example:	List all the resources that have a mismatch: CHECK_ RESOURCES , LISTALL=YES
Tip:	N/A

CLONE_INSTANCE

This verb clones the instance a specified number (nnnn) of times and each new instance name is suffixed with one to four digits: 0000 through nnnn-1.

The cloned instance name, plus its suffix, cannot exceed 32 bytes: 1 to 28 digits for the instance name + 1 to 4 digits for the suffix. Therefore, instance name length = 32, the length of the suffix.

- A new OBJID, OBJDATE, and OBJTIME are generated for the cloned objects.
- COUNT is the number of clones to spawn, and can range from 1 to 2000.

Syntax:	DOMAIN= , CLASS= , INSTANCE= , COUNT=nnnn
Example:	Clone 100 instances in the specified class, naming the cloned instances DIFF80 through DIFF899 type: DOMAIN=SOFTWARE , CLASS=USER , INSTANCE=DIFF8 , COUNT=100
Tip:	Run LIST_INSTANCE to display the instance names before and after the cloning.

COPY_CLASS

This verb copies a class template, its component instances, and resource data from one domain to another.

Note: While this verb is supported, it is recommended to use the verbs `EXPORT_CLASS` and `IMPORT_CLASS` to copy a class from the CSDB.

As of version 4.4 of the CSDB, this verb will copy the component instances and resource data also.

- TODB specifies the path to a destination file other than the one in the `edmpprof` file. If omitted, it defaults to the DBPATH specified in the `edmpprof` file.
- If TOCLASS is omitted, the destination class will be assumed to be the same as the FROMCLAS. In this case, TODOMAIN and FROMDOMA must be different.
- A new object ID, ODJDATE, and OBJTIME are generated for the copied objects.
- If TODOMAIN does not exist, the ZBASE class and base instance will be copied from the source domain (FROMDOMA) to create a valid domain.
- If REPLACE=YES, all existing data is replaced.

Syntax:	(TODB=) (, FILE=,) FROMDOMA=, FROMCLAS=, TODOMAIN=(, TOCLASS=) (, REPLACE=YES/ NO)
Example:	Copy the specified class template and only the base instance from SOFTWARE to SYSTEM: FROMDOMA=SOFTWARE, FROMCLAS=USER, TODOMAIN=SYSTEM
Tip:	Run once with REPLACE=NO to determine the preexistence of the class in the destination domain.

COPY_DOMAIN

This verb copies only the class template and `_BASE_INSTANCE_` of a domain within a database, and optionally, to a different destination database. To copy the entire contents of a domain, see the section, "[Copying a Domain and its Contents](#)" on the next page.

- A new object ID, OBJDATE, and OBJTIME are generated for the copied objects.
- If REPLACE=YES, all existing data is replaced.
- TODB specifies the path to a destination file other than the one in the `edmpprof` file. If omitted, it defaults to the DBPATH specified in the `edmpprof` file.

Syntax:	(FILE=,) FROMDOMA=, TODOMAIN=(, REPLACE=YES/NO) (, TODB=)
Example:	Copy the class template and <code>_BASE_INSTANCE_</code> of the SOFTWARE Domain

	to the SYSTEM domain: FROMDOMA=SOFTWARE, TODOMAIN=SYSTEM
Tip:	Run once with REPLACE=NO to determine the preexistence of the destination domain.

Copying a Domain and its Contents

To copy a domain, run this verb with FROMDOMA and TODOMAIN specified, then use the export and import verbs as detailed below.

To copy a domain with contents:

1. Run this verb with FROMDOMA and TODOMAIN specified.
This will copy just the _BASE_INSTANCE_ and class template.
2. Verify that the new domain has been created by checking the ZEDMAMS log's return code or physically querying the CSDB using an explorer tool.
3. Use the EDMAMS verbs, "[EXPORT_INSTANCE](#)" on page 642 and "[EXPORT_RESOURCE](#)" on page 644 with **DOMAIN=<old_domain>** (the value of FROMDOMA).
4. Import the domain using the EDMAMS verb, "[IMPORT_INSTANCE](#)" on page 646. Be sure to specify:
 - the exported instance (**FILE=**),
 - the resource files (**XPR=**),
 - **MAP_DOMAIN=old_domain/new_domain**.

COPY_FIELD

This verb copies an attribute (FROMFLD) and its instance data to a new attribute (TOFLD). The length, type, and flags of the new attribute will be inherited from the existing attribute.

- INDEX is the nth occurrence of a FROMFLD multiple-named attribute.
- PREVIEW will display the value of the FROMFLD attribute before changing an existing TOFLD attribute.
- REPLACE=YES will overlay existing data in the TOFLD attribute with the values in the FROMFLD attribute.
- If KEEPDATE=NO, a new zobjdate and zobjtime are generated.

Syntax:	(FILE=,) DOMAIN=, CLASS=, FROMFLD=, TOFLD= (, INDEX=) (, PREVIEW=YES/NO) (, REPLACE=YES/NO) (, KEEPDATE=YES/NO)
Example:	Copy an attribute named OLDFLD (and the associated data in the specified class) to a new attribute named NEWFLD: DOMAIN=SOFTWARE, CLASS=USER, FROMFLD=OLDFLD, TOFLD=NEWFLD
Tip:	Run LIST_INST_DATA against the class to see the contents of each attribute.

COPY_INSTANCE (COPY_RESOURCE)

These verbs copy a range of specified instances and resource data from one domain-class pair to another, in the PRIMARY and RESOURCE files, and optionally, to a different destination database. The function assumes that the destination class name is the same as the source, and that the templates are identical.

Note: While this verb is supported, it is recommended to use the verbs EXPORT_INSTANCE and IMPORT_INSTANCE to copy an instance from the CSDB. This verb can be specified as either COPY_INSTANCE or COPY_RESOURCE, as these verbs have been combined and perform identical functions.

As of version 4.4 of the Configuration Server Database, this verb will copy the component instances and resource data also.

- TODB specifies the path to a destination file other than the one in the `edmprof` file. If omitted, it defaults to the DBPATH specified in the `edmprof` file.
- The value of FROMINST can be partially specified. For example, you can specify FROMINST=DIFF to specify all the instances that contain DIFF as any part of the string.
- A new object ID is generated for the copied objects. A new OBJDATE and OBJTIME will be generated unless KEEPDATE=YES.
- PREVIEW=YES will display a list of instance names that will be copied from the source class. PREVIEW=NO will display instances that have been copied.
- If REPLACE=NO and an existing instance is found, the function will stop, indicating the existing instance name to STDERR and the log, and listing in the log any instances that have been copied.
- The existing instance name will be written to STDERR, and instances that have been copied will be listed in the log.
- If PREVIEW=NO and the destination domain does not contain FROMCLAS, the source class (FROMCLAS) and its BASE_INSTANCE will be copied to the destination domain (TODOMAIN).
- NEWINST renames an instance (if TODB is specified).
If wildcards are used for FROMINST, NEWINST is not allowed.

Syntax:	<pre>(PREVIEW=YES/NO) (, TODB=) (, FILE=,) FROMDOMA=, FROMCLASS=, TODOMAIN=, FROMINST= (, NEWINST=) (, REPLACE=YES/ NO) (, KEEPDATE=YES/NO)</pre>
Example:	Copy instances and resource data from the SOFTWARE Domain to the SYSTEM domain, with the from-instance wildcard specified as CICS*ICO*: FROMDOMA=SOFTWARE, FROMCLAS=ZSERVICE, TODOMAIN=SYSTEM, FROMINST=CICS*ICO*

Tips:	Run once with PREVIEW=YES to get a list of instances to be copied. Also, run LIST_RESOURCES against the source domain to display a list of existing resources and instance names that can be copied, and then against the destination domain to see what was copied. Destination domain instance names that match the new instance name will be deleted.

COPY_NEW_SUFFIX

This verb copies the specified instances and resource data from one domain to another in the PRIMARY and RESOURCE files; and adds a new suffix to the destination instance. This verb allows wildcards for FROMINST.

- The value of FROMINST can be partially specified. For example, you can specify FROMINST=DIFF to select all the instances that contain DIFF as the first part of the string.
- If the length of NEWSUFF is longer than that of OLDSUFF, the new instance name must not exceed 32 bytes. If it does, a message will be placed in the log and the instance will not be copied.
- PREVIEW=YES will display the old instance name and the new instance name and the total number of instances to be copied.
- If the class does not exist in the TODOMAIN, the class and BASE_INSTANCE will be copied from the source domain (FROMDOMA).
- Instance names must match the prefix (FROMINST) and the suffix (OLDSUFF) to be copied and renamed.

Syntax:	(PREVIEW=YES /NO,) FROMDOMA=, FROMCLASS=, FROMINST=, TODOMAIN=, OLDSUFF=, NEWSUFF=
Example:	From the SOFTWARE Domain, copy and re-suffix instances and resource data prefixed with TSO and suffixed with ICO, to the SYSTEM domain with the new suffix, TSO_*)_NEW_ICO; and displaying only those that would be copied: FROMDOMA=SOFTWARE, FROMCLAS=ZSERVICE, TODOMAIN=SYSTEM, FROMINST=TSO, OLDSUFF=ICO, NEWSUFF=NEW_ICO
Tips:	Run once with PREVIEW=YES to verify that the new instance names will not exceed 32 bytes, and that the instances required will be copied. Also, run LIST_RESOURCES against the source domain to display a list of existing resources and instance names that can be copied and then against the destination domain to see what was copied. Destination domain instances that match the new instance name will be deleted.

CREATE_INSTANCES

This verb creates Instances in the specified CSDB Domain from data in an edited text file (INFILE). The INFILE must conform to the following specifications.

- The first line must contain a new Instance name starting in column 1 for a maximum of 32 bytes.
- The next line contains the variable field name, beginning in column 2, for a maximum of eight bytes; then a blank in column 10; followed by a two-byte index value (01-99) in columns 11 and 12; followed by the data to be populated beginning in column 13.
- Existing Instance names will be bypassed and the function will stop unless a /* (in columns 1 and 2) line follows the last data line of an Instance. In the example below, the function will continue to the next input Instance.

Column Number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Line 1	U	S	E	R	_	N	A	M	E															
Line 2		N	A	M	E		J	.	A	.	D	E	V	E	L	O	P	E	R					
Line 3		E	D	M	S	E	T	U	P		D	O	M	A	I	N	.	C	L	A	S	S		
Line 4		Z	P	R	I	O	R	I	T		9	9	9											
Line 5	U	S	E	R	_	N	A	M	E		L	.	Z	I	M	M	E	R						
Line 6		N	A	M	E																			
Line 7		E	D	M	S	E	T	U	P		U	S	E	R	.	U	S	E	R	3				
Line 8		Z	T	R	A	C	E		N	N	N													

Syntax:	INFILE=, DOMAIN=, CLASS=
Example:	Create the instances in the specified text file in CLASS=USER: INFILE=MYINPUT.TXT, DOMAIN=POLICY, CLASS=USER
Tip:	N/A

DELETE_CLASS

This EDMAMS verb deletes a class template and all of its instances.

As of version 4.4 of the Configuration Server Database, this verb will delete the component instances and resource data also.

- If PACKAGE=YES, all component and component data is deleted.

Syntax:	(FILE=) (, PREVIEW=YES/NO) (, PACKAGE=YES/NO,) DOMAIN=, CLASS=

Example:	Delete the class, USERTEST, from the SOFTWARE Domain: DOMAIN=SOFTWARE, CLASS=USERTEST
Tip:	N/A

DELETE_COMP_ORPHS

This verb deletes component orphans from the PACKAGE class. Orphans are defined as RESOURCE file data that have no mated instances in the associated PRIMARY file PACKAGE class.

- CLASS defaults to all classes of the specified domain.

Syntax:	(FILE=) (, PREVIEW=YES/NO,) DOMAIN= (, CLASS=)
Example:	Delete orphans from the TSTRESLT Class of the SOFTWARE Domain: DOMAIN=SOFTWARE, CLASS=TSTRESLT, PREVIEW=NO
Tip:	N/A

DELETE_DOMAIN

This verb deletes a domain or an alphabetical range of domains (class templates and instances) from the file that is specified.

As of version 4.4 of the Configuration Server Database, this verb will delete the component instances and resource data also.

- If TODOMAIN is omitted, only FROMDOMA will be deleted.
If TODOMAIN is specified, the range is inclusive.

Syntax:	(FILE=) (, PREVIEW=YES/NO,) FROMDOMA= (, TODOMAIN=)
Example:	From the PRIMARY file, delete only SOFTWARE: FILE=PRIMARY, FROMDOMA=SOFTWARE From the PROFILE File, delete USERA through USERF: FILE=PROFILE, FROMDOMA=USERA, TODOMAIN=USERF
Tip:	Run with PREVIEW=YES first.

Caution: Use caution when specifying a range of domains. Be certain of the range specified, because there is no confirmation request.

DELETE_FIELD

This verb deletes an attribute from a template along with its README field, and reorganizes the class accordingly.

- FILE will default to **PRIMARY** if it is omitted or if no value is specified.
- DOMAIN specifies the name of the CSDB Domain.

- CLASS specifies the name of the CSDB Class.
- FIELD must refer to an attribute, not a README (description) field.
- INDEX refers to multiple occurrences of the same attribute name. For example, if there were three occurrences of EDMSETUP, the third would be INDEX=3. Index deletes the *n*th occurrence of a multiple named variable.
Values for INDEX are the numbers 1 through 99.
The default is 1.

Syntax:	(FILE=,) DOMAIN=, CLASS=, FIELD= (, INDEX=)
Example:	Delete the attribute USERATTR from the USER Class: DOMAIN=POLICY, CLASS=USER, FIELD=USERATTR
Tip:	Use with discretion because deletion of an attribute causes a restructuring and rewrite of all the instances in the class.

DELETE_INSTANCE

This verb deletes an Instance or an alphabetical range of Instances within a specified Class of the CSDB.

The keywords, TOINST and SUFFIX, were deleted from this verb's functionality with version 4.3 of the Configuration Server Database.

As of version 4.4 of the Configuration Server Database, this verb will delete the component instances and resource data also.

- FROMINST and INSTANCE perform the same function; either one must be used. Both will delete groups of instances and any existing, associated resource data. To delete only one instance, the entire name must be specified.
- FROMINST and INSTANCE can specify wildcards (*). For example, DIFF* and DIFF*SOL*.
- PREVIEW=YES displays a list of instances that would be deleted with PREVIEW=NO.
- INDATA specifies the fully qualified path to, and name of, a file that contains the delete parameters. This file can be either an exported instance deck (.XPI), or a manually created file that contains the FILE, DOMAIN, CLASS, and INSTANCE values.
If using INDATA, the values that are specified for FILE, DOMAIN, CLASS, FROMINST, and INSTANCE on the command line are ignored because these values are extracted from the INDATA file.
When not using INDATA, either FROMINST or INSTANCE *must* be specified. Both can delete groups of instances.
FROMINST (instead of INSTANCE) can be specified inside the INDATA file to define a group of instances to be deleted.

Syntax:	(PREVIEW=YES/NO) (, INDATA=) (, FILE=,) DOMAIN=, CLASS= (, FROMINST=) (, INSTANCE=)
Example:	Delete all USER Class instances, beginning with the instance prefix name of OLD_USER: DOMAIN=POLICY, CLASS=USER, FROMINST=OLD_USER

Tip:	Run once with PREVIEW=YES to see which instances will be deleted. Also, run LIST_INSTANCES against the class to display a list of instance names that might be deleted.

Caution: Use caution when specifying a range of domains. Be certain of the range specified, because there is no confirmation request.

DELETE_ORPHANS

This verb will delete all orphans in all domains. Orphans are defined as RESOURCE file data that have no mated instance in the associated PRIMARY file.

- TRACE=YES provides additional diagnostic (tracing) confirmation in the log.

Syntax:	(PREVIEW=YES/NO) (, TRACE=YES/NO)
Example:	Delete all orphans: PREVIEW=NO
Tip:	N/A

DELETE_RESOURCE

This verb deletes the specified resource from the RESOURCE File of the CSDB.

The keyword, DELETE_RESOURCE, was deleted from the Configuration Server as of version 4.2. Its functionality is handled by the verb, ["DELETE_INSTANCE" on the previous page.](#))

EDIT_CLASS_PREFIX

This verb enables you to edit the first 60 bytes of the template's prefix.

- FIELD is the name of the class prefix field, such as the priority of the class, CLASSPRI.
- VALUE will not be read if FIELD is not specified.
Refer to the **Values** column in ["Changeable Fields" below](#) for the valid values for each FIELD type.
- If KEEPDATE=NO, a new OBJDATE and OBJTIME are generated.

Changeable Fields

Field	Values
CLASTYPE	P (POLICY_CLASS_TYPE) C (CONFIGURATION_CLASS_TYPE) T (COMPONENT_CLASS_TYPE) B (CLASS_TYPE_BLANK)
CLASSPRI	5 (PATH), 10 (METACLAS), 50 (FILE), 50 (ZSERVICE), 60 (DESKTOP), 70 (MACALIAS), 70 (REGISTRY), 50 (DEFAULT_PRIORITY), B (Blank)

DBTYPE	U (UNICODE) A (ASCII) Note: A third value (E) is not applicable to the current release of the Configuration Server.
OBJDM	D (DISTRIBUTED_MANAGER)
OBJTYPE	S (SINGLE_DIMENSIONAL_OBJECT) M (MULTIPLE_DIMENSIONAL_OBJECT)
SEQ_SENS	S (SEQUENCE_SENSITIVE) I (SEQUENCE_INSENSITIVE) Note: Specifies whether to process variables in the order in which they occur in the class template (S), or in order of attribute type (I)—that is, VARs then CONNs, and so on.
OBJNAME	Any text up to 20 bytes in length. If there are embedded spaces in the text, enclose the text with quotation marks (" "), as in the example.

Syntax:	DOMAIN=, CLASS=, FIELD=, VALUE= (, PREVIEW=YES/NO) (, KEEPDATE=YES/ NO)
Example:	DOMAIN=POLICY, CLASS=USER, FIELD=OBJNAME, VALUE="User names", PREVIEW=NO
Tip:	N/A

EXPORT_CLASS

This verb enables the exporting of class template data from a file or data set for importing to another file or data set.

- If PREVIEW=YES, OUTPUT is ignored.
- OUTPUT is the name of the destination output file (with extension) where the exported data is to reside.
- INPUT references a predefined input file, which enables multiple FILE.DOMAIN.CLASS combinations to be specified.
The input file must be specified in the following format.
`FILE=file_name, DOMAIN=domain_name, CLASS=class_name, INSTANCE=instance_name`
- HEADER=YES produces an output header file.
- COMMENT=YES adds a comment to the optional output file header.
- If the deck is being exported from a database with a *locale* or *codepage* that differs from the database into which it is being imported, the appropriate keyword (FROM_LOCALE or FROM_CODEPAGE) must be specified.
If the source database's *locale* or *codepage* is the same as the database into which it is being imported, neither keyword needs to be specified.

- If the deck is being exported from a database with a *locale* or *codepage* that differs from the database into which it is being imported, the appropriate keyword (TO_LOCALE or TO_CODEPAGE) must be specified.
If the source database's *locale* or *codepage* is the same as the database into which it is being imported, neither keyword needs to be specified.
 - The values for the CODEPAGE keywords must be integers, such as 1252, 65001, and 936.
 - The LOCALE keywords' values will accept the following values only: **LEGACY**, **UTF8**, and **UTF-8**.
 - **LEGACY** is an alias for the local machine's code page, such as CP_ACP, 1252.
 - **UTF8** and **UTF-8** are aliases for a UTF-8 code page, such as CP_UTF8, 65001.
- If TO_LOCALE=
 - **LEGACY**: XPR headers will be automatically converted to EBCDIC.
 - **UTF8**: an internationalized ZEDMAMS UTF-8 export deck is produced.

Note: The keywords FROM_LOCALE, FROM_CODEPAGE, TO_LOCALE, and TO_CODEPAGE will not show up on the command line when the syntax/usage is queried; however, they are functional and, when specified, will work as designed.

Syntax:	FILE=(, DOMAIN=) (, CLASS=) (, PREVIEW=YES/NO,) OUTPUT=(, COMMENT=) (, INPUT=) (, HEADER=YES/NO)
Example:	Export all the classes (templates) in the PRIMARY file to a file named EXPC.DAT : FILE=PRIMARY, PREVIEW=NO, OUTPUT=C:\EXPC.DAT
Tip:	N/A

EXPORT_INSTANCE

This verb enables the exporting of instances to an output file or data set for importing to another file or data set or for reporting purposes.

- If PREVIEW=YES, OUTPUT is ignored.
- OUTPUT is the name of the destination output file (with extension) where the exported data is to reside.
- KEEP specifies a text file that contains a list of the instance attributes to be retained in the resulting output file. These names are case-sensitive.
- DROP specifies the instance attributes that are not to be retained in the output file.
- If ORDER=YES, the resulting file will be ordered by attribute name.
- COMMENT=YES adds a comment to the optional output file header.
- Specify REPORT=YES to export instances to third-party vendor software.
- CSVL=YES produces a Comma Separated Variable listing for all attribute values. Currently, the output file you specify with CSVL=YES is created in a subdirectory of the current working directory.

- If BASE=YES, the values of the `_BASE_INSTANCE_` will be inherited.
- INPUT references a predefined input file, which enables multiple FILE.DOMAIN.CLASS.INSTANCE combinations to be specified.
The input file must be specified in the following format.
`FILE=file_name, DOMAIN=domain_name, CLASS=class_name, INSTANCE=instance_name`
- HEADER=YES produces an output header file.
- If SKIP_ERRORS=YES and database or consistency errors are encountered in the PROFILE File, a “bad object” event (return code=4) will be recorded in the log, but processing will continue.
If SKIP_ERRORS=NO (the default) and errors are encountered, the exporting will stop.
- PHEX=YES outputs the data portion of variables in printable hex format.
- If the deck is being exported from a database with a *locale* or *codepage* that differs from the database into which it is being imported, the appropriate keyword (FROM_LOCALE or FROM_CODEPAGE) must be specified.
If the source database’s *locale* or *codepage* is the same as the database into which it is being imported, neither keyword needs to be specified.
- If the deck is being exported from a database with a *locale* or *codepage* that differs from the database into which it is being imported, the appropriate keyword (TO_LOCALE or TO_CODEPAGE) must be specified.
If the source database’s *locale* or *codepage* is the same as the database into which it is being imported, neither keyword needs to be specified.
 - The values for the CODEPAGE keywords must be integers, such as 1252, 65001, and 936.
 - The LOCALE keywords’ values will accept the following values only: **LEGACY**, **UTF8**, and **UTF-8**.
 - **LEGACY** is an alias for the local machine’s code page, such as CP_ACP, 1252.
 - **UTF8** and **UTF-8** are aliases for a UTF-8 code page, such as CP_UTF8, 65001.
- If TO_LOCALE=
 - **LEGACY**: XPR headers will be automatically converted to EBCDIC.
 - **UTF8**: an internationalized ZEDMAMS UTF-8 export deck is produced.

Note: The keywords FROM_LOCALE, FROM_CODEPAGE, TO_LOCALE, and TO_CODEPAGE will not show up on the command line when the syntax/usage is queried; however, they are functional and, when specified, will work as designed.

Syntax:	FILE=(, DOMAIN=(, CLASS=(, INSTANCE=(, FROMINST=(, TOINST=(, FROMDATE=(, PREVIEW= YES/NO) (, TODATE=(, FROMTIME=(, TOTIME=(,) OUTPUT=(, KEEP=YES/ NO) (, DROP=(, ORDER=(, COMMENT=(, CSVL=YES/ NO) (, PHEX=YES/ NO) (, BASE=YES/NO) (, INPUT=(, HEADER=YES/NO) (, REPORT=YES/NO) (, SKIP_ERRORS=YES/NO)
Example:	Export all the instances in the PRIMARY file to a file named EXPI.DAT:

	FILE=PRIMARY, PREVIEW=NO, OUTPUT=C:\EXPI.DAT
Tip:	N/A

EXPORT_RESOURCE

This verb enables the exporting of resource data to an output file or data set for importing to another file or data set.

- If PREVIEW=YES, OUTPUT is ignored.
- OUTPUT is the name of the destination output file (with extension) where the exported data is to reside.
- Specify COMMENT=YES to add a comment to the optional output file header.
- INPUT references a predefined input file, which enables multiple FILE.DOMAIN.CLASS.INSTANCE combinations to be specified.
The input file must be specified in the following format.
`FILE=file_name, DOMAIN=domain_name, CLASS=class_name, INSTANCE=instance_name`
- HEADER=YES produces an output header file.
- If the deck is being exported from a database with a *locale* or *codepage* that differs from the database into which it is being imported, the appropriate keyword (FROM_LOCALE or FROM_CODEPAGE) must be specified.
If the source database's *locale* or *codepage* is the same as the database into which it is being imported, neither keyword needs to be specified.
- If the deck is being exported from a database with a *locale* or *codepage* that differs from the database into which it is being imported, the appropriate keyword (TO_LOCALE or TO_CODEPAGE) must be specified.
If the source database's *locale* or *codepage* is the same as the database into which it is being imported, neither keyword needs to be specified.
 - The values for the CODEPAGE keywords must be integers, such as 1252, 65001, and 936.
 - The LOCALE keywords' values will accept the following values only: **LEGACY**, **UTF8**, and **UTF-8**.
 - **LEGACY** is an alias for the local machine's code page, such as CP_ACP, 1252.
 - **UTF8** and **UTF-8** are aliases for a UTF-8 code page, such as CP_UTF8, 65001.
- If TO_LOCALE=
 - LEGACY: XPR headers will be automatically converted to EBCDIC.
 - UTF8: an internationalized ZEDMAMS UTF-8 export deck is produced.

Note: The keywords FROM_LOCALE, FROM_CODEPAGE, TO_LOCALE, and TO_CODEPAGE will not show up on the command line when the syntax/usage is queried; however, they are functional and, when specified, will work as designed.

Syntax:	FILE=(, DOMAIN=) (, CLASS=) (, INSTANCE=) (, FROMINST=) (, TOINST=) (, PREVIEW=

	YES/NO) (, FROMDATE=) (, TODATE=) (, FROMTIME=) (, TOTIME=,) OUTPUT= (, COMMENT=YES/ NO) (, INPUT=) (, HEADER=YES/ NO)
Example:	Export all resources in the RESOURCE file to a file named EXPR.DAT: FILE=PRIMARY, PREVIEW=NO, OUTPUT=C:\EXPR.DAT
Tip:	N/A

IMPORT_CLASS

This verb enables you to import template data from an exported data set or output file to a PRIMARY File specified in the `edmpprof` file.

- FILE is the name of the file or data set that contains the import class data..
- TIME=NEW – generates a new OBJDATE, OBJTIME, and OBJID.
TIME=OLD – retains the original OBJDATE, OBJTIME, and OBJID.
TIME=MOD – generates a new OBJDATE and OBJTIME, but retains the original OBJID.
- TODOMA is the domain with which matching source domains are replaced.
- If the FROMDOMA domain exists in the destination database, specify TIME=NEW to avoid duplicate object IDs.
If FROMDOMA is specified, TODOMA must also be specified.
- If REPLACE=NO, the class template will not be replaced.
- If the deck is being imported into a *locale* or *codepage* that differs from the database from which it was exported, the appropriate keyword (TO_LOCALE or TO_CODEPAGE) must be specified.
If the target *locale* or *codepage* is the same as the database from which the deck was exported, neither keyword needs to be specified.
- If the deck is being imported into a *locale* or *codepage* that differs from the database from which it was exported, the appropriate keyword (FROM_LOCALE or FROM_CODEPAGE) must be specified.
If the target *locale* or *codepage* is the same as the database from which the deck was exported, neither keyword needs to be specified.
 - The values for the CODEPAGE keywords must be integers, such as 1252, 65001, and 936.
 - The LOCALE keywords' values will accept the following values only: **LEGACY**, **UTF8**, and **UTF-8**.
 - **LEGACY** is an alias for the local machine's code page, such as CP_ACP, 1252.
 - **UTF8** and **UTF-8** are aliases for a UTF-8 code page, such as CP_UTF8, 65001.
- Translated data must fit the field size limits.
- Legacy XPR headers in EBCDIC are automatically converted to ASCII.

Note: The keywords FROM_LOCALE, FROM_CODEPAGE, TO_LOCALE, and TO_CODEPAGE will not show up on the command line when the syntax/usage is queried; however, they are functional and, when specified, will work as designed.

Syntax:	FILE=(, PREVIEW=YES/NO) (, TIME=OLD/NEW/MOD) (, REPLACE=YES/NO) (, FROMDOMA=) (, TODOMA=)
Example:	Import all the classes in the file specified by FILE to a PRIMARY file specified in the edmprof file: FILE= <i>input_file</i> , PREVIEW=NO
Tip:	N/A

IMPORT_INSTANCE

The IMPORT_INSTANCE verb enables an administrator to import instance and resource data from an exported data set or output file (an *import deck*) to a location in the CSDB. The import deck will be imported to the CSDB PRIMARY File that is specified for the DBPATH setting in the MGR_DIRECTORIES section of the edmprof file, such as:

```
[MGR_DIRECTORIES]DBPATH =  
<InstallDir>\ConfigurationServer\DB
```

The entire process consists of the following phases.

Verify the Import Deck

This verification checks the internal integrity (such as, size, referential integrity, and validity of data) of the incoming deck.

Preview the Import Deck

This phase is a comprehensive *analysis* that scans the database and the entire import deck, and reports the results, detailing differences that are relevant to this import session. This analysis checks for duplicate object IDs (OIDs), determines if fixes are possible, and determines if a new deck is required. The entire import deck is analyzed before this phase completes.

Note: For the import deck, an *all-or-nothing* rule applies. Therefore, if an error condition is realized for one instance of the import deck, the entire deck is invalid.

When OIDs in the deck are the same as OIDs in the database, this verb's behavior is dictated by the keywords REPLACE and CONTINUE. These keywords are discussed in detail.

Import the Instances and Resources

The instances and resources from the deck will be imported only if the integrity check (performed during the **Preview the Import Deck** phase) is free of errors.

The IMPORT_INSTANCE information is presented as follows:

- "[Verb History](#)" on the next page describes when the keywords were introduced or discontinued.

- ["Syntax" on the next page](#) details all the keywords that are associated with this verb. This includes their expected behavior, and their interactions with, and dependencies on, one another.
- ["Retired Syntax" on page 651](#) covers the keywords that have been retired from use in this version, and includes information about which new keywords have replaced them.
- ["Usage Considerations" on page 651](#) addresses some of the more noticeable and critical effects that might result from using this verb, as well as some of the **new features**.
- ["Examples" on page 652](#) presents a few examples of how to express the keywords.

Verb History

This verb was introduced with version 4.3 of the CM Configuration Server to manage its database.

- The keywords, VERIFY and LOGFILE, were added to this verb's functionality in version 4.4 of the CM Configuration Server.
- In version 4.5.2 of the CM Configuration Server:
 - The keywords, FROMDOMA, TODOMA, TIME, CHGCONS, FORCE, Y2K, and IMPORT_RESOURCE were removed from this verb's functionality—but continue to be supported. See the section, ["Retired Syntax" on page 651](#).
 - The keywords, XPR, DUPLICATES, FORCE, CONTINUE, NEW, AUTOFIX, MAP_DOMAIN, and COMMIT_CHANGES were added to provide enhanced behavior management in the event of duplicate object IDs, multiple domains, and import decks from older systems and databases.

Syntax

This section details the syntax (**keywords** and **values**) that is associated with this verb, including the most efficient and effective ways to use it.

```
FILE=(, PREVIEW=YES/NO) (, DUPLICATES=STOP/MANAGE) (, XPR=) (, NEW=)
(, REPLACE=YES/
NO) (, VERIFY=YES/NO) (, AUTOFIX=YES/NO) (, MAP_DOMAIN=) (, CONTINUE=YES/
NO) (, COMMIT_CHANGES=YES/NO) (, LOGFILE=)
```

Note: The keywords FROM_LOCALE, FROM_CODEPAGE, TO_LOCALE, and TO_CODEPAGE will not show up on the command line when the syntax/usage is queried; however, they are functional and, when specified, will work as designed.

- If the deck is being imported into a *locale* or *codepage* that differs from the database from which it was exported, the appropriate keyword (TO_LOCALE or TO_CODEPAGE) must be specified.
If the target *locale* or *codepage* is the same as the database from which the deck was exported, neither keyword needs to be specified.
- If the deck is being imported into a *locale* or *codepage* that differs from the database from which it was exported, the appropriate keyword (FROM_LOCALE or FROM_CODEPAGE)

must be specified.

If the target *locale* or *codepage* is the same as the database from which the deck was exported, neither keyword needs to be specified.

- The values for the CODEPAGE keywords must be integers, such as 1252, 65001, and 936.
- The LOCALE keywords' values will accept the following values only: **LEGACY**, **UTF8**, and **UTF-8**.
 - **LEGACY** is an alias for the local machine's code page, such as CP_ACP, 1252.
 - **UTF8** and **UTF-8** are aliases for a UTF-8 code page, such as CP_UTF8, 65001.
- Translated data must fit the field size limits.
- Legacy XPR headers in EBCDIC are automatically converted to ASCII.

Note: FILE is the only keyword that must be specified on the command line in order for this verb to execute. The other keywords are optional (as denoted by their inclusion in parentheses); if they are not specified, their defaults will be assumed and they will affect the processing of this verb.

- PREVIEW creates a preview listing of the input file contents, the expected results, and its ability to be imported. The results are written to the log file. The default is **YES**.

Note: To better understand the functionality of this keyword, think of it as asking, "Preview only?"

If PREVIEW=NO, the processing will run and the import deck data *can* be written to the database—depending on the other keywords and the results of the comparison.

If PREVIEW=YES (the default), the only result is a log file being generated—no action is taken on the database.

If PREVIEW=NO, the processing will run and, provided there are no errors, the import deck can be written to the database.

- FILE is the fully qualified path and filename of the file that contains the collection of instances for the import deck. This file is commonly suffixed with the extension **XPI**, as shown in the examples starting

Note: If a fully qualified path is not specified (as shown below), the location of the import file is assumed to be that from which ZEDMAMS is running.

ZEDMAMS VERB=IMPORT_INSTANCE,FILE=DB_001.XPI,PREVIEW=YES

- **DUPLICATES** enables an administrator to indicate the action to be taken when duplicate OIDs of instances are encountered (in the import deck and the database). The default is **STOP**. **DUPLICATES=STOP** (the default) will result in this operation stopping. The return code **8** will be reported.
DUPLICATES=MANAGE will result in a new deck being created to avoid the re-use of a previously allocated OID. If the instance is data bearing, it can be corrected only if the resource is available, in which case, XPR must be specified.

Note: A new deck could be created for any of the following reasons: duplicate OIDs; domain change; OBJRCRC is NULL; duplication, re-mapping, or OID difference between source decks and target database.

- XPR is the fully qualified path and filename of the resource deck.
- REPLACE dictates the behavior of the process (as defined in the following conditions) when identical instances are discovered in the database and the deck. The default is **NO**.
If REPLACE=YES, the data in the import deck can be written to the database.
If REPLACE=NO (the default), the database and import deck will be queried for differences, and the following logic will apply.
 - If no differences are found, processing will continue with the next instance in the import deck.
 - If differences are found, the value of CONTINUE is checked.
 - If CONTINUE=NO, each instance with a difference will be ignored and processing will continue with the next instance in the import deck, but the process will not proceed to the next phase. A return code of **8** will be returned.
 - If CONTINUE=YES, each instance with a difference will be ignored and processing will continue with the next instance in the import deck.
- CONTINUE dictates the behavior of the process when matching records are discovered. The default is **NO**.
If CONTINUE=YES:
 - For any class attribute found in the target class template, the import will continue as long as it doesn't result in the truncating of any significant (non-blank) data. In this case, the process will continue and an error will be reported.
 - For any class attribute found in the target class template, but the import cannot import the data without truncating significant (non-blank) data, the process will continue, an error will be reported, and the import will fail.
 - For any class attribute not found in the target class template, the field will be dropped—even if it contains significant (non-blank) instance data. (A warning or error message will be issued, indicating this occurrence.)

Note: Fields and data that are dropped will be documented in the log file.

- If CONTINUE=NO (the default) *and*...
- the class attribute is not defined in the target class template, the import will fail.
- VERIFY compares the date (ZOBJDATE) and time (ZOBJTIME) of incoming files with those of the database files, if specified as YES. The default is **NO**.
If VERIFY=YES and...
 - the dates and times do not match (**rc=8**), a warning message is issued, and processing will continue with the next instance in the import deck.
 - the dates and times match and XPR was specified (with a valid value), the VERIFY_IMPORT verb will run to check the integrity of the decks.
 - the dates and times match, but XPR has not been specified (or its value is invalid), verification is not possible.

The results are reported to ZEDMAMS.LOG (the default), unless a different log has been specified for LOGFILE.

Note: If the dates and times match, the ZEDMAMS.LOG will report a successful verify; if not, the ZEDMAMS.LOG will report a failed verify. These verifications are done on a per-instance basis and reported at the end of the process.
If VERIFY=YES, PREVIEW=YES and REPLACE=YES are assumed—but nothing is imported to the database.

If VERIFY=NO, no verification is done.

- NEW is the fully qualified path and filename prefix of the new decks that will be created. Optionally, just the filename prefix can be specified, in which case the new decks will be created in the current directory (by default, the `bin` directory).

Note: This keyword is applicable only if DUPLICATES=MANAGE.

- If either the filename prefix or the fully qualified path have embedded blanks, the entire string must be enclosed in quotation marks (“”).
- The decks will have the suffixes **.MPI** and **.MPR**.
- The defaults are the fully qualified paths of **XPI** (as specified by the keyword, FILE) and **XPR** (as specified by the keyword, XPR), respectively.
- AUTOFIX dictates whether to delete orphaned resources. If importing a data bearing instance and the resource file exists in the database, the existing resource will be deleted and the incoming resource will be written to the database. The default is **NO**.

Caution: This keyword should be used with extreme caution and only by an experienced administrator in a controlled manner. Incorrect use could result in the accidental removal of database elements that are critical to performance and operation.

If AUTOFIX=YES, orphaned resources will be deleted.

- MAP_DOMAIN enables an administrator to import all instances from one domain into a different domain, thereby facilitating application management.
Use MAP_DOMAIN=*source_domain/target_domain* to import all instances that originated in one domain, *source_domain*, into a domain with a different name, *target_domain*. Doing so triggers the creation of a new deck. All object IDs from a domain that matches *source_domain* are placed in the new deck. In the new deck, their domain value is replaced by that of *target_domain*.

For example, import all instances of the SOFTWARE Domain to the domain, SOFTBACK, by specifying,

```
MAP_DOMAIN=SOFTWARE/SOFTBACK
```

Note: The new domain must exist in the database. If it doesn't, the process will stop.

- COMMIT_CHANGES dictates whether to commit to the database, the data in the import deck. The default is **YES**.
COMMIT_CHANGES=**YES** (the default) will write the changes to the database.

If COMMIT_CHANGES=NO, the changes will not be written to the database, but new **.MPI** and **.MPR** decks will be produced (if required).

If there are no changes, this keyword is ignored.

- LOGFILE (see the section, "[LOGFILE](#)" on page 621).

Retired Syntax

As the EDMAMS verbs have evolved, changes have been made to enhance their processing. Because of this, and to maintain logic for the user, there have been changes to the syntax of some of the verbs.

This section details the keywords that have been retired from use for the IMPORT_INSTANCE verb.

Note: While retired, these keywords are still supported. These are superseded by new keywords where indicated.

- TIME=OLD (the default) retains the original OBJDATE, OBJTIME, and OBJID. (This is superseded by DUPLICATES=STOP.)
TIME=NEW generates a new OBJDATE, OBJTIME, and OBJID. (This is superseded by DUPLICATES=MANAGE.)
TIME=MOD generates a new OBJDATE and OBJTIME, but retains the original OBJID.
- FROMDOMA=<source_domain> and TODOMA=*target_domain* have been replaced by MAP_DOMAIN.
- FORCE=YES/NO. (This is superseded by CONTINUE.)
- CHGCONS specifies whether any embedded references to the name specified by FROMDOMA should be changed to the name specified by TODOMA.
- IMPORT_RESOURCE dictates whether this operation should import resources.

Usage Considerations

This section addresses some of the more noticeable and critical effects that might result from using this verb, as well as some of the new features.

- After the data from an import deck has been written to the CSDB, the changes are considered permanent. Therefore, it is imperative that a CSDB administrator using this verb be certain of the changes that are being considered.

Note: It is recommended to
Shut down the Configuration Server to ensure that the CSDB contents are not changing during the processing.
Back up the CSDB before running this verb.
Specify PREVIEW=YES and check the resulting log before committing any changes to the database.

- Executing this verb might result in the creation of new decks (**MPI/MPR**). The circumstances under which this might happen are:
 - There exist duplicate object IDs (ZOBJID) in the database instances and the import deck instances, and DUPLICATES=MANAGE.
 - There is a domain name change for the imported data (using the MAP_DOMAIN keyword).
 - The ZOBJRCRC (the object resource CRC) is NULL or empty and the value can be calculated and assigned in the process.

Note: All of these occur in conjunction with the existence of the XPR deck, it being specified on the command line, and the values of the CONTINUE and REPLACE allowing the processing to continue.
The XPR deck is necessary so that if changes are required, it is available to be updated at that time.

- This version of IMPORT_INSTANCE has more consistency checks that must be passed before the import deck is considered valid for import.
- Combining instances and resources on the same command line allows the import deck to be more thoroughly examined.
- Use the CONTINUE option to prevent data loss during import.
- In this version, an administrator:
 - Can manage the processing behavior if an import deck OID collides with a database OID.
 - Can specify the filename prefix if a new deck is generated.
 - Has a single keyword to facilitate changing domains.
- A new time-based format of object ID generation has been introduced, thereby eliminating the chance of randomly generating a duplicate OID. For more information on this feature, see the ["MGR_STARTUP" on page 547](#).

Examples

The following examples offer a look at the ways this verb can be used.

Note: Even though some keywords are dependent on another, the order in which they are specified on the command line is not significant.
If a keyword is not specified on the command line, but has a default, the default will be assumed.
Some keywords, while optional, become mandatory based on the specifications of others and the results of processing. For an example, see DUPLICATES=MANAGE.

Example1

Import the instance data that is contained in DB_001.XPI and DB_001.XPR to the PRIMARY file that is specified in the edmprof file. Do not write the changes to the database. Do not manage duplicate object IDs. Write the results to C:\Temp\EDMAMS\DB001\Test01.log.

```
ZEDMAMS VERB=IMPORT_INSTANCE, FILE=DB_001.XPI, XPR=DB_001.XPR, PREVIEW=YES, LOGFILE=C:\Temp\EDMAMS\DB001\Test01.log
```


In this run, the implied values (defaults) that affected the processing are Duplicates=STOP and Continue=NO.

Example 2

Import the instance data that is contained in **DB_001.XPI** and **DB_001.XPR** to the PRIMARY file that is specified in the `edmprof` file. Do not write the changes to the database. Manage any duplicate object IDs that are encountered. Query the database and check for matching records and, if any are found, continue processing. Delete any orphaned resources that are encountered. Write the results to `C:\Temp\EDMAMS\DB001\Test02.log`.

```
ZEDMAMS VERB=IMPORT_INSTANCE,FILE=DB_001.XPI,XPR=DB_001.XPR,PREVIEW=YES,DUPLICATES=MANAGE,CONTINUE=YES,AUTOFIX=YES,LOGFILE=C:\Temp\EDMAMS\DB001\Test02.log
```

In this run, the implied value (default) that affected the processing is REPLACE=NO.

Example 3

Assume that the Example 2 command line has completed as specified. Execute the same run, but this time, write the changes to the database and the results to `C:\Temp\EDMAMS\DB001\Test02.log`.

```
ZEDMAMS VERB=IMPORT_INSTANCE,FILE=DB_001.XPI,XPR=DB_001.XPR,PREVIEW=NO,DUPLICATES=MANAGE,CONTINUE=YES,AUTOFIX=YES,LOGFILE=C:\Temp\EDMAMS\DB001\Test02.log
```

In this run, the implied values (defaults) that affected the processing are: COMMIT_CHANGES=YES.

Note: The only difference between examples 2 and 3 is the value of PREVIEW=.

Example 4

Import the instance data (contained in **DB_001.XPI** and **DB_001.XPR**) from the SOFTWARE Domain to the SOFTBACK domain in the PRIMARY file in the database. Do not write the changes to the database. Do not manage duplicate object IDs. Query the database and check for matching records and, if any are found, continue processing. Delete any orphaned resources that are encountered. Write the results to `C:\Temp\EDMAMS\DB001\Test03.log`.

```
ZEDMAMS VERB=IMPORT_INSTANCE,FILE=DB_001.XPI,XPR=DB_001.XPR,MAP_DOMAIN=SOFTWARE/SOFTBACK,PREVIEW=YES,REPLACE=NO,CONTINUE=YES,AUTOFIX=YES,LOGFILE=C:\Temp\EDMAMS\DB001\Test03.log
```

In this run, the implied value (default) that affected the processing is `DUPLICATES=STOP`.

Example 5

Import the instance data that is contained in **DB_001.XPI** and **DB_001.XPR** to the PRIMARY file that is specified in the `edmprof` file. Write the changes to the database. Do not query the database and deck for matching records. Write the results to

`C:\Temp\EDMAMS\DB001\Test04.log`.

```
ZEDMAMS VERB=IMPORT_INSTANCE,FILE=DB_001.XPI,XPR=DB_001.XPR,PREVIEW=NO,REPLACE=YES,LOGFILE=C:\Temp\EDMAMS\DB001\Test04.log
```

In this run, the implied values (defaults) that affected the processing are `COMMIT_CHANGES=YES`, `DUPLICATES=STOP`, and `CONTINUE=NO`.

Import and Export files

"[Import and Export File Names](#)" below presents a list of the six default import/export files that are generated by the CSDB. Their level in the CSDB is part of the logic in their naming.

Import and Export File Names

Database Level	Import File Name	Export File Name
Class	.MPC – modified class file	.XPC – exported class file
Instance	.MPI – modified instance file	.XPI – exported instance file
Resource	.MPR – modified resource file	.XPR – exported resource file

Note: The export files (`XPI` and `XPR`) are the original decks that might be generated during the export process, and are the files that are imported either back into the existing database or into another database. So, each `XPI` and `XPR` file can be an export *and* import file. The `MPI` and `MPR` files are decks that are generated during a database import that resulted in correcting duplicate OID issues, changing domains, or correcting empty or NULL OBJRCRCs.

IMPORT_RESOURCE

This verb imports resource data from an exported data set or file to a RESOURCE File as specified in the `edmprof` file.

The keyword, `VERIFY`, was added to this verb's functionality in version 4.4 of the Configuration Server.

- If `REPLACE=NO`, the class template will not be replaced.
- If `VERIFY=YES`, an implied `PREVIEW=YES` is set.
If a resource exists, its size is compared to the size of the incoming resource to verify that they match.

- If the deck is being imported into a *locale* or *codepage* that differs from the database from which it was exported, the appropriate keyword (TO_LOCALE or TO_CODEPAGE) must be specified.
If the target *locale* or *codepage* is the same as the database from which the deck was exported, neither keyword needs to be specified.
- If the deck is being imported into a *locale* or *codepage* that differs from the database from which it was exported, the appropriate keyword (FROM_LOCALE or FROM_CODEPAGE) must be specified.
If the target *locale* or *codepage* is the same as the database from which the deck was exported, neither keyword needs to be specified.
 - The values for the CODEPAGE keywords must be integers, such as 1252, 65001, and 936.
 - The LOCALE keywords' values will accept the following values only: **LEGACY**, **UTF8**, and **UTF-8**.
 - **LEGACY** is an alias for the local machine's code page, such as CP_ACP, 1252.
 - **UTF8** and **UTF-8** are aliases for a UTF-8 code page, such as CP_UTF8, 65001.
- Translated data must fit the field size limits.
- Legacy XPR headers in EBCDIC are automatically converted to ASCII.

Note: The keywords FROM_LOCALE, FROM_CODEPAGE, TO_LOCALE, and TO_CODEPAGE will not show up on the command line when the syntax/usage is queried; however, they are functional and, when specified, will work as designed.

Syntax:	FILE= (, FROMDOMA=) (, TODOMA=) (, PREVIEW=YES/NO) (, REPLACE=YES/NO) (, VERIFY=YES/NO)
Example:	Import all resources in the file specified by FILE to a RESOURCE file specified in edmprof: FILE=RESOURCE, PREVIEW=NO
Tip:	N/A

LIST_CLASSES

This verb displays a list of class names, object IDs, and other 60-byte prefix information, such as ZOBJDATE, ZOBJTIME, persistence flag, sequence sensitive flag, Distributed Configuration Server flag and db type and count totals.

Syntax:	FILE=, DOMAIN=
Example:	List all classes in the PRIMARY file: DOMAIN=*
Tip:	N/A

LIST_CONNECTS

This verb displays a list of connect-to values (type C) for the specified instances.

- INSTANCE can be partially specified.
To display only one instance, the entire name must be specified.

Syntax:	DOMAIN=, CLASS= (, INSTANCE=)
Example:	List all of the connect-to values in the ZSERVICE Class of the SOFTWARE Domain: DOMAIN=SOFTWARE, CLASS=ZSERVICE
Tip:	To list data for all type C values in all instances of the specified class, omit INSTANCE.

LIST_CONS_VARS

This verb automatically displays a list of connect-to data (type C) and, optionally, variable data (type V) for the specified instances.

- Wildcards (*) can be specified in INSTANCE.
For example, specify DIFF* to select all the instances that contain DIFF as the first part of the string.

To display only one instance, the entire name must be specified.
- If VTYPE=YES, variable data will be included in the display.

Syntax:	(FILE=,) DOMAIN=, CLASS= (, INSTANCE=) (, VTYPE=YES/NO)
Example:	From the USER Class in the SOFTWARE Domain, list the connect-to and variable values only for those instances prefixed with DIFF: DOMAIN=SOFTWARE, CLASS=ZSERVICE, INSTANCE=DIFF, VTYPE=YES
Tip:	To list all C- and V-type data in all the instances of the specified class, omit INSTANCE.

LIST_DOMAINS

This verb displays an alphabetical list of domains for a specified file (the default is the PRIMARY file).

- FROMDOMA is the domain from which to start the list of domains.
If omitted, all the domains through the TODOMAIN will be listed.
- TODOMAIN is the domain at which to end the list of domains.
If omitted, all the domains following FROMDOMA will be listed.

Syntax:	FILE= (, FROMDOMA=) (, TODOMAIN=)
Example:	In the PRIMARY file, list all of the domains that follow the domain ACCT: FILE=PRIMARY, FROMDOMA=ACCT In the PROFILE File, list the domains in the range ACCT through SALES (inclusive): FILE=PROFILE, FROMDOMA=ACCT, TODOMAIN=SALES
Tips:	To list all the domains of the selected file, simply omit FROMDOMA and TODOMAIN. To list one domain, specify it for FROMDOMA and TODOMAIN.

LIST_FLAGS

This verb displays the attribute name, length, type, and Configuration Server and Sentient agent flags for a specific class template.

- Specify README=YES to include the README attributes.

Syntax:	DOMAIN=, CLASS= (, README=YES/NO)
Example:	List the attribute information for the attributes of the ZSERVICE Class of the SOFTWARE Domain and omit the README attributes: DOMAIN=SOFTWARE, CLASS=ZSERVICE
Tip:	N/A

LIST_INST_DATA

This verb displays, in a concise format, the attribute data of the specified instances.

- Wildcards (*) can be specified for INSTANCE.
For example, specify DIFF* to select all the instances that contain DIFF as the first part of the string.

To display only one instance, the entire name must be specified.

- Use FIELDS to specify up to six attribute names.
- Specify the fields with a space separating each name, and the entire string enclosed in quotation marks, as in:
"field1 field2 field3 field4 field5 field6"

To list all attribute data of all instances of the specified class, omit FIELDS.

Note: If FIELD is omitted, all attribute data of all instances of the specified class will be displayed. This might produce a very large log, which might hinder locating data.

If a fieldname does not exist in the template, it will be ignored.

Syntax:	DOMAIN=, CLASS= (, INSTANCE=) (, FIELDS=)
Example:	From the ZSERVICE Class in the SOFTWARE Domain, list the attribute data of all instances with the prefix, RAD: DOMAIN=SOFTWARE, CLASS=ZSERVICE, INSTANCE=RAD*
Tip:	To list all instances of the specified class, omit INSTANCE.

LIST_INSTANCE

This verb displays a list of instance names and object IDs for the class specified. It also displays the ZOBJTIME and resource size.

- The value of FROMINST can be partially specified.
For example, specify FROMINST=DIFF to select all the instances that contain DIFF as any part of the string.

Use a wildcard (*) to specify this value as a prefix, as in RAD*.

- The value of SUFFIX can be partially specified.
For example, specify SUFFIX=INT to select all the instances that have INT as a suffix.

Syntax:	DOMAIN=, CLASS= (, FROMINST=) (, SUFFIX=)
Example:	From the USER Class of the POLICY Domain, list the instance names and object IDs for all instances with the prefix RAD, and all instances with the suffix, PORT: DOMAIN=POLICY, CLASS=USER, FROMINST=RAD*, SUFFIX=PORT
Tip:	To list all instances, omit FROMINST.

LIST_PACKAGE

This verb lists the instances and all mated components of the PACKAGE class.

- The value of INSTANCE can be partially specified.
For example, specify INSTANCE=DIFF to select all the instances that contain DIFF as any part of the string.

Use a wildcard (*) to specify this value as a prefix, as in, RAD*; and as a suffix, as in, *INT.

Syntax:	(FILE=,) DOMAIN=, INSTANCE=
Example:	From the SOFTWARE Domain, list all instances with the prefix RAD: DOMAIN=SOFTWARE, INSTANCE=RAD*
Tip:	N/A

Note: CLASS is not an option because this verb applies to the PACKAGE class only.

LIST_PREFIX

This verb displays data from the Distributed Configuration Server prefix.

If CLASS is omitted, all class prefixes will be displayed.

Syntax:	(FILE=,) DOMAIN= (, CLASS=)
Example:	List the Distributed Configuration Server prefixes for all classes in SOFTWARE Domain of the PRIMARY file: DOMAIN=SOFTWARE
Tip:	N/A

LIST_RESOURCES

This verb displays a list of resource names (with promote dates, times, data names, data sizes, and object IDs) from the PRIMARY File.

As of version 4.3 of the Configuration Server Database, this verb was renamed. Its original name was LIST_RESOURCE.

- If there is no mated instance in the PRIMARY file, an appropriate message will be generated.
- If ORPHANS=YES, only the orphans will be listed.
- If CHKSIZE=YES, the size listed (from ZOBJRSIZ) is compared to the actual size (from the NvdDBFind).
Only those resources that have a size anomaly will be listed.
- SIZE is the size of the resource.
SIZE must be specified as a range (not a single byte size), and the range must be defined with a dash (-), as in, 100-500.

Syntax:	DOMAIN=, CLASS= (, ORPHANS=YES/NO) (, CHKSIZE=YES/NO) (, SIZE=nnnn-nnnn)
Example:	From the FILE Class of the SOFTWARE Domain, list all resources that are between 64 and 1024 bytes in length, and compare this with the actual size: DOMAIN=SOFTWARE, CLASS=FILE, CHKSIZE=YES, SIZE=64-1024
Tip:	N/A

LIST_ZRSC_FIELDS

This verb displays the data in all fields that begin with **ZRSC**, such as ZRSCSIZE and ZRSCVRFY. This verb allows wildcards for INSTANCE.

INSTANCE can be specified with a partial name. For example, specify INSTANCE=DIFF to select all the instances that contain DIFF as any part of the string.

Use a wildcard (*) to specify this value as a prefix, as in, RAD*; and as a suffix, as in, *INT. To display only one instance, the entire name must be specified.

Syntax:	DOMAIN=, CLASS= (, INSTANCE=)
Example:	From the ZSERVICE Class of the SOFTWARE Domain, list the instances that begin with CICS: DOMAIN=SOFTWARE, CLASS=ZSERVICE, INSTANCE=CICS*
Tip:	To list all instances of the specified class, do not specify INSTANCE.

MATCH_RESOURCES

This verb will compare resource data from the RESOURCE file against instance names from the PRIMARY file to determine and display whether those resources are mated (orphaned). This

comparison is made by reading the resource data, extracting the instance name from the resource prefix, and attempting to find the mated resource.

The keyword, PREVIEW, was added to this verb as of version 4.3 of the Configuration Server Database.

- If PREVIEW=NO, and there is resource data that has been determined to be orphaned, a search of the PRIMARY file instances will occur, to locate a matching instance object ID. If a match is made, the instance name is placed in the resource data prefix and the resource is updated, with the time stamp for the resource data being updated with the ZRSCDATE and ZRSCTIME.
- Totals at completion include resources found, as well as total resources (mated and orphaned).
- When using PREVIEW=NO, a great deal of I/O might occur.

Syntax:	DOMAIN=, CLASS= (, PREVIEW=YES/NO)
Example:	Match and display resource data names for all the SOFTWARE.FILE Instances: DOMAIN=SOFTWARE, CLASS=FILE, PREVIEW=YES
Tip:	N/A

If many orphans are detected with PREVIEW=YES, a more efficient way to update the resource data file is to use the verb "[ZRSOURCE_UNMATES](#)" on [page 666](#).

PACKAGE_UNMATES

This verb lists all PACKAGE class instances that do not have mated components in the domain that is specified.

- DOMAIN must be one that has a PACKAGE class.
- CLASS defaults to **PACKAGE**.

Syntax:	DOMAIN= (, CLASS=)
Example:	From the SOFTWARE Domain, list all the PACKAGE class instances that have no mated components: DOMAIN=SOFTWARE
Tip:	N/A

REFRESH_DMA

This verb will recount all the instances, classes, and domains in the PRIMARY file and, optionally, refresh the **count** (TotalInstanceCount, TotalClassCount, and TotalDomainCount) and **date** (LastUpdateDate, LastInstanceUpdateDate, and LastClassUpdateDate) fields in the appropriate Distributed Configuration Server prefix areas. Additionally, the updated output can be displayed in the log.

This verb was introduced with version 4.2 of the CM Configuration Server. It replaced the verb, REFRESH_COUNTS.

The keywords, DOMAIN and CLASS, were added with version 4.3; and the keyword, COUNTS_ONLY, was added with version 4.5.2.

- If PREVIEW=YES, a count of instances by class, domain, and file will be listed to the log, but no data will be written.
The log will display the actual count (as calculated by running this verb) and the current count (current values in the total count fields [mentioned in the introductory paragraph] in the database) in the Distributed Configuration Server prefix. If the actual count differs from the current count, the latter will be flagged with an asterisk (*).
- If PREVIEW=NO, the TotalInstanceCount, TotalClassCount, and TotalDomainCount (for each applicable Distributed Configuration Server prefix) will be computed and updated, in addition to updating the current counts.
- If COUNTS_ONLY=NO (the default) and PREVIEW=NO, the current count and date fields in the Distributed Configuration Server area for each class will be updated.
If COUNTS_ONLY=NO and PREVIEW=YES, the current count and date fields in the Distributed Configuration Server area for each class will be displayed.
- If COUNTS_ONLY=YES and PREVIEW=NO, the current count fields only will be updated.
This is effective for very large database files because, by not refreshing the date fields and not having to read every instance in the database, the function executes in less time.
If COUNTS_ONLY=YES and PREVIEW=YES, the current count and actual count fields will be displayed, but not updated; the date fields are not touched.
- Each class for each domain will be previewed separately, and at the end of each domain, a domain summary will be presented.
- After the last domain, a file summary will be presented by listing the ZBASE.ZBASE template information.

Syntax:	PREVIEW=YES/NO (, DOMAIN=) (, CLASS=) (, COUNTS_ONLY=YES/NO)
Example:	Preview the counts and update dates in the Distributed Configuration Server prefix for the entire PRIMARY file: PREVIEW=YES
Tip:	N/A

RENAME_INSTANCE

This verb will rename instances and the internal name of any mated resource data.

- KEEP indicates whether the old instance will be deleted.
- OLDPREFIX specifies existing instances that are to be renamed.
If a single instance is to be renamed, specify the entire name.

If multiple instances are to be renamed, a wildcard (*) is required.

For example, to change the names of the instances, **east_sales** and **north_sales** to **US_Sales**, specify **OLDPREFIX=*_sales,NEWPREFIX=US_Sales**.

- NEWPREFIX is that which will replace OLDPREFIX.

Syntax:	DOMAIN=, CLASS=, OLDPREFIX=, NEWPREFIX= (, KEEP=YES/ NO) (, PREVIEW= YES /NO)
Example:	In the ZSERVICE Class of the SOFTWARE Domain, rename all instances prefixed with EAST to NORTH_EAST, and delete the old prefix: DOMAIN=SOFTWARE, CLASS=ZSERVICE, OLDPREFIX=EAST, NEWPREFIX=NORTH_EAST, PREVIEW=NO
Tip:	N/A

SEARCH_INSTANCES

This verb will search the specified instances for the data contained in STRING.

- The data specified is not case-sensitive; if it contains embedded spaces, it must be enclosed in quotation marks (" ").
- The output log will contain the name of the instance, attribute, and the specified STRING value.
- If the value of STRING is not found in the specified instances, this will be reported in the log.
- FROMINST can be specified with a partial name.
For example, specify FROMINST=DIFF to select all the instances that contain DIFF as any part of the string.

Use a wildcard (*) to specify this value as a prefix, as in, RAD*; and as a suffix, as in, *INT.

To display only one instance, the entire name must be specified.

Syntax:	DOMAIN=, CLASS= (, FROMINST=,) STRING=
Example:	Search for the string "J. Q. Public" in all instances that end in EAST in the USER Class of the SOFTWARE Domain: DOMAIN=SOFTWARE, CLASS=USER, STRING="J. Q. Public", FROMINST=*EAST
Tip:	To list all instances of the specified class, omit FROMINST.

SORT_OBJECT_ID

This verb will sort object IDs within a Domain.

- FILE will default to **PRIMARY** if it is omitted or if no value is specified.
- DOMAIN can be a single Domain (*domain_name*) or all Domains (ALL) of the specified File.
To sort the object IDs of multiple (but not all) Domains in a File, execute this verb once for each Domain.
To sort the object IDs of multiple Domains in multiple Files, execute this verb once for each Domain.
- ORDER=NOSORT means that the object IDs will be listed in CLASS.INSTANCE groups.
ORDER=ASCEND/DESCEND specifies the order of sorting, based on object IDs.
Duplicate object IDs will be flagged in ascending or descending order.

- ALLIDS is valid only if ORDER=ASCEND or DESCEND.
ALLIDS=YES will duplicate (list all object IDs) the entire File.
ALLIDS=NO will list only duplicate object IDs.

Syntax:	(FILE=,) DOMAIN= (ALL/DOMAIN) (, ORDER=ASCEND/DESCEND/NOSORT) (, ALLIDS= YES/NO)
Example:	Sort, in descending order, all the object IDs of all domains, and duplicate the entire file: DOMAIN=ALL, ORDER=DESCEND, ALLIDS=YES
Tip:	All domains can be selected by specifying DOMAIN=ALL.

SYNC_CLASS

This verb will synchronize an existing (target) class with a newly formatted (source) class, and re-organize all existing class instances according to the mapping in the source class template.

- All target class attributes that have a match in the new template will adopt the characteristics of that matching attribute.
Any target class attributes that do not have a match in the new template will be deleted.
- TODOMAIN specifies the class that contains the target class template that is to be synchronized.
- SYNCDOMA specifies the class that contains the source class template. The default is **ZEDMSYNC**.
- CLASS is the target class (within the domain that is specified by TODOMAIN *and* SYNCDOMA) that will be synchronized.
The value of CLASS must exist in the TODOMAIN and SYNCDOMA domains; if it doesn't, the function will fail.
- CACHE=YES will update loaded cache when running as a Configuration Server method.
- If BASE=YES, this verb will copy the _BASE_INSTANCE_ from the source (SYNCDOMA) to the target (TODOMAIN).

Syntax:	TODOMAIN= (, PREVIEW=YES/NO) (, SYNCDOMA=ZEDMSYNC,) CLASS= (, CACHE=YES/ NO) (, BASE=YES/NO)
Example:	In the POLICY Domain, synchronize the existing USER Class with a newly formatted USER Class, imported from ZEDMSYNC. Include the _BASE_INSTANCE_ and update the loaded cache: TODOMAIN=POLICY, CLASS=USER, PREVIEW=NO, CACHE=YES, BASE=YES
Tip:	N/A

UPDATE_INSTANCES

This verb will update instances in the specified domain from data in an edited text file, INFILE. INFILE must conform to the following specifications.

- The first line must contain a new instance name (max. 32 bytes) starting in column 1. The next line contains the variable field name (max. eight bytes) starting in column 2; then a blank in column 10; a 1-byte index value (1-9) in column 11; a blank in column 12; and the data to be populated beginning in column 13.
- REPLACE=YES specifies that an attribute that contains existing data (non-blank) be overlaid.
- Use a slash-asterisk combination (/ *) in columns 1 and 2 to denote the end of instance data (see lines 5 and 10 below).

Column Number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Line 1	U	S	E	R	_	N	A	M	E															
Line 2		N	A	M	E								J	.	A	.	D	E	V	E	L	O	P	
Line 3		E	D	M	S	E	T	U	P		5		5	T	H	A	T	T	R	I	B	U	T	E
Line 4		Z	P	R	I	O	R	I	T				9	9	9									
Line 5	/	*																						
Line 6	U	S	E	R	_	N	A	M	2															
Line 7		N	A	M	E								L	.	Z	I	M	M	E	R				
Line 8		E	D	M	S	E	T	U	P		2		2	N	D	A	T	T	R	I	B	U	T	E
Line 9		E	D	M	S	E	T	U	P		3		3	R	D	A	T	T	R	I	B	U	T	E
Line 10	/	*																						
Line 11	U	S	E	R	_	N	A	M	E	_	3													
Line 12		N	A	M	E								J	I	M		D	O	E					
Line 13		E	D	M	S	E	T	U	P		4		4	T	H	A	T	T	R	I	B	U	T	E

Syntax:	INFILE=, DOMAIN=, CLASS= (, REPLACE=YES/NO)
Example:	Update the instances in the MYINPUT.TXT file in the POLICY Domain's USER Class: INFILE=MYINPUT.TXT, DOMAIN=POLICY, CLASS=USER
Tip:	Run LIST_INST_DATA or LIST_FLAGS to determine the index of a like named variable to be updated.

UPDATE_MGRIDS

This verb updates the specified Manager ID, Manager name, managing Manager ID, and managing Manager name in the Distributed Configuration Server prefix.

Note: The Configuration Server was previously called the *Manager*. Therefore, the keywords associated with this verb retain the 'Manager' designation, as in, Manager name (MNAME).

- All keywords are optional. However, at least one keyword other than DOMAIN and CLASS must be specified to avoid a usage error being displayed to STDERR.

Syntax:	(FILE=) (, DOMAIN=) (, CLASS=) (, MID=) (, MMID=) (, MNAME=) (, MMNAME=)
Example:	Update the managing Configuration Server IDs and managing Configuration Server names in the USER Class of the POLICY Domain in the PRIMARY file: FILE=PRIMARY, DOMAIN=POLICY, CLASS=USER, MMID=010, MMNAME=New_Mngng_RCS
Tip:	To update an entire file, omit DOMAIN and CLASS.

VERIFY_CLASS

This verb will display class templates, as specified, to determine if any gaps, overlays, or other anomalies exist. The output log will consist of a template entry number, the attribute name, its length, and its displacement in the heap. If an error is found, it will be indicated in the appropriate place.

- DOMAIN must be specified.
All domains can be selected by specifying the value as ALL.
- CLASS must be specified.
All classes can be selected by specifying the value as ALL.

Syntax:	(FILE=,) DOMAIN=, CLASS=
Example:	In the POLICY Domain of the PRIMARY file, verify all the class templates: FILE=PRIMARY, DOMAIN=POLICY, CLASS=ALL Verify the ZSERVICE Class templates in the PRIMARY file: DOMAIN=ALL, CLASS=ZSERVICE
Tip:	N/A

VERIFY_DATABASE

This verb will validate the integrity of a CSDB. It can be used at any time to check database integrity, and can run standalone or as a Configuration Server method. This database validation must be done in read-only mode.

This verb was introduced with version 4.5.1 of the CM Configuration Server.

- DOMAIN can be a single domain or all domains of the CSDB. The default is ALL.
- DEPTH can be DOMAIN, CLASS, INSTANCE, or RESOURCE. The default is RESOURCE.
- ZFILE is the name of the output file, which contains suggested actions to be carried out when there are errors in the database. This can be used as input file for running ZFILE option of ZEDMAMS.
- LARGEDB=YES, will use 'file based storage' for verify database processing rather than using limited 'process virtual memory'. This option is added to support verify database of very large database with millions of instances. The default is NO.
When this option is used, a temporary file with the format 'zedmams_ldb_<datetime>.dat' is created for processing and automatically deleted after the processing is over.
- STMPATH is the path used for creating the temporary file, with the option LARGEDB=YES. The temporary file default location assumed on Windows is the bin folder in which the zedmams.exe utility is running.

To verify the integrity of multiple (but not all) domains in the CSDB, execute this verb once for each domain.

Note:

- This is an exhaustive check of the database and, as such, might take a long time to run, possibly several hours. It makes two passes over the database – first, checking the PRIMARY file and any associated resources; and second, checking the RESOURCE file for orphans. As a result of the two passes, it has sufficient information to do a check for duplicate object IDs.
- If you verify a very large database by running ZEDMAMS command with verb VERIFY_DATABASE and DEPTH=RESOURCE / INSTANCE, you can set the OBJECTID_VERDB_INITIAL_ENTRIES parameter in MGR_CACHE section of the edmpconf.dat file, to avoid memory allocation errors. Set the value for the OBJECTID_VERDB_INITIAL_ENTRIES parameter to approximately 1.2 times the total number of instances in the database.
- If you have a very large database (with millions of instances) and memory allocation errors still occurs after the above suggested step. Use the option LARGEDB=YES, which will use 'file based storage' for processing rather than using limited 'process virtual memory'.

Syntax:	DOMAIN= (ALL / <i>any_domain</i>) , (DEPTH=) , (ZFILE=) , (LARGEDB=YES / NO) , (STMPATH= <i>file_path</i>)
Example:	Verify the integrity of the CSDB SOFTWARE Domain: DOMAIN=SOFTWARE Verify the integrity of all domains in the database: DOMAIN=ALL Verify the integrity of all domains in the database using LARGEDB option: DOMAIN=ALL, DEPTH=RESOURCE, LARGEDB=YES, STMPATH=D:\temp
Tip:	Specify DOMAIN=ALL to select all domains in the CSDB.

ZRSOURCE_UNMATES

This verb will match Instances in the PRIMARY File with the resource data in the RESOURCE file, based on the specified Domain and Class, and the object ID from the PRIMARY File instances.

- If resource data is not found, the Instance is considered *unmated*.
- If resource data is found, the Instance name in the resource data prefix is compared with the Instance name from the PRIMARY File.
If it does not match, it is reported in the log as an inconsistency.
- If PREVIEW=NO, the resource prefix is updated to reflect the true Instance name from the PRIMARY file.
- At completion, totals are listed in the log to indicate the number of unmated Instances, inconsistencies, and so forth.
- DOMAIN is that which contains resource data, usually SOFTWARE.
- If DEBUG=YES, all Instances are listed as they are verified.

Syntax:	DOMAIN=, CLASS= (, PREVIEW=YES/NO) (, DEBUG=YES/NO)
Example:	In the FILE Class of the SOFTWARE Domain in the PRIMARY File, list all unmated Instances and inconsistencies: PREVIEW=YES, DOMAIN=SOFTWARE, CLASS=FILE, DEBUG=YES
Tip:	N/A

Configuration Server Database Utility (RadDBUtil)

Caution: It is recommended to create a backup of the Configuration Server Database before executing any of the commands that are shown in this chapter.

RadDBUtil (`raddbutil.exe`) is the Configuration Server Database tool that manages:

- Configuration Server Database updates (imports, exports, and deletions),
- Configuration Server communications,
- activity logging, and
- version queries.

This chapter focuses on the functionality, syntax, and common-use attributes of this tool, and provides examples of each of these capabilities.

Note: The Configuration Server does not need to be running in order for RadDBUtil to run. Additional information is detailed in the section, "[Running RadDBUtil from a Command Line](#)" on the next page.

Following are the components that benefit from using RadDBUtil:

- Configuration Server
- Configuration Server Database
- Distributed Configuration Server

Following are the processes that benefit from using RadDBUtil:

- Importing/exporting to/from the CSDB
- Output materials produced by, or during, importing and exporting
- Deleting instances and resources from the CSDB
- Querying and manipulating the Configuration Server lock status
- Version information queries
- Activity logging

Running RadDBUtil from a Command Line

This section provides information regarding running RadDBUtil from a command line.

Note: It is important to note that RadDBUtil must be able to lock the CSDB, which it cannot do when it is run from a command line if:

The Configuration Server is running

and

MANAGER_TYPE=STANDALONE.

To run RadDBUtil from a command line, do either (or both) of the following *before* running RadDBUtil.

- Shut down the Configuration Server.
- Open the `edmpprof` file and, in the MGR_STARTUP section, set MANAGER_TYPE to something other than STANDALONE—either DISTRIBUTED or SERVER.

Implementation Details

RadDBUtil has a dependency on the `edmpprof` file, and should be placed in the same directory as it—typically the Configuration Server `bin` directory on Windows, RadDBUtil must be able to find a valid CSDB and the Configuration Server `log` directory.

Caution: It is recommended to create a backup of the CSDB before executing any of the commands that are shown in this chapter.

Sentient Enterprise Patch Manager Considerations

This section contains important information and warnings about using RadDBUtil to import and export Sentient Enterprise Patch Manager (Patch Manager) bulletins.

Note: Important information regarding the deleting of Patch Manager bulletins is detailed in the section, ["Deleting Bulletins from a Database" on page 678](#).

IMPORT

Caution: When using the `-domain` switch, all instances and resources (of the XPI file that is specified for INPUT) will be imported to the target domain. Therefore, it is imperative that the necessary domains and classes exist in the target domain.

EXPORT

Note: In order for the RadDBUtil tool to successfully export bulletins, in the PATCHMGR.ZSERVICE Class template there must be an attribute named SYNC of the type CONNECTION with the appropriate default value. See ["EXPORT" on page 675](#) for an example of the SYNC-CONNECTION attribute being specified for a Patch Manager bulletin. If this attribute is not present in the PATCHMGR.ZSERVICE Class template, it must be added.

EDMPROF File Settings

The following edmprof file settings affect the operation of RadDBUtil. Be sure to verify these settings and adjust them accordingly.

MGR_LOG.DIRECTORY

specifies where the activity and audit logs will be generated. For more information, see ["Output Files" on page 674](#)

MGR_DIRECTORIES.DBPATH

determines the CSDB location.

MGR_STARTUP.MGR_ID

is the unique, three-character identifier of the Configuration Server that was specified during its installation.

MGR_STARTUP.MGR_NAME

is the identifying name of the Configuration Server that was specified during its installation.

RadDBUtil Verbs

In this section, each of the six RadDBUtil verbs is detailed with syntax options, syntax descriptions, and verb-specific considerations.

General Syntax

- The executable, verbs, keywords, and values are not case-sensitive.
- The verbs are ["VERSION" on the next page](#), ["LOG" on the next page](#), ["IMPORT" on page 671](#), ["EXPORT" on page 675](#), ["DELETE" on page 677](#), and ["RCS" on page 679](#).
- The verbs can be specified in either of the following formats:
`-keyword value`

If this syntax is used, pairs of combinations must be separated by a space, as in:

```
-keyword value -keyword value -keyword value
```

or

```
keyword=value
```

If this syntax is used, pairs of combinations must be separated by a space, a comma, or both, as in:

```
keyword=value keyword=value, keyword=value, keyword=value
```

Note: A double-dash (--) indicates the end of the options.

The acceptable Boolean values are:

TRUE: 1, YES, ON, and TRUE

FALSE: 0, NO, OFF, and FALSE

VERSION

The VERSION verb produces build information for the RadDBUtil tool and all embedded executables. For examples of the syntax, see ["IMPORT and EXPORT Examples" on page 680](#).

Syntax

The syntax of the verb, VERSION, is shown below.

```
Raddbutil version
```

LOG

This verb places RadDBUtil-specific messages in the audit log and activity log.

The audit log information is:

BuildInfo	CommandLine
CompletionCode	Timestamp (ISO)

There are two messages per command—one when RadDBUtil begins and one when it ends.

```
20050126 15:24:04 C:/RadDBUtil/radddbutil.exe 60 --> import20050126  
15:24:04 C:/RadDBUtil/radddbutil.exe 60 <-- rc=8
```

Syntax

The syntax of the LOG verb is shown below. For examples of the syntax, see ["IMPORT and EXPORT Examples" on page 680](#).

```
RadDBUtil LOG "record this information"
```

The RadDBUtil entry in the log can be accompanied by customized text.

The text must be enclosed within quotation marks, as shown above. For example, to delineate a nightly log entry for a specific date, specify:

```
RadDBUtil log "Nightly Log for June 22, 2005"
```

IMPORT

Caution: It is recommended to create a backup of the CSDB before executing any of the commands that are shown in this section.

The IMPORT verb simplifies the importing of materials into the CSDB. It can import materials from one domain into a domain of a different name within a CSDB, and from one CSDB to another. It offers the following options to optimize the import operation.

- Automatically recognize and re-use the instances that exist in the target domains, and
- Import only those elements that do not exist in the target domains.

Additionally, by having a feature that allows the IMPORT verb to identify packages and dialogs that exist in the database and to dynamically adapt to them, RadDBUtil reduces the size of the targeted domain—as well as Distributed Configuration Server execution times—without affecting the integrity of the imported materials. Other features of the IMPORT verb are:

- Parameter validation
- Database receptiveness to receiving the import materials
- Deck verification (XPI and XPR)
- Attributes that are dropped from the import operation will be noted with a warning level (--?) in the activity log.

Note: For the IMPORT verb, an “all-or-nothing” rule applies—that is, if RadDBUtil *cannot* do *all* of that which is requested, it will do *none* of that which is requested. If a RadDBUtil IMPORT operation fails, the Configuration Server log will have an entry reflecting this and the return code will be rc=8.

Syntax

The syntax of the IMPORT verb is shown in this section. For examples of the syntax, see ["IMPORT and EXPORT Examples" on page 680](#).

The keywords and values are not case-sensitive, as can be seen in the following examples.

Note: Optional keyword-value combinations are in parentheses. Default values are underlined.

```
Radddbutil import -input FileName (-output value) (-domain value) (-commit false) (-root *) (-accept a) (-reject u+d) (-ignore s)
```

```
radddbutil IMPORT INPUT=value(, OUTPUT=value(, DOMAIN=domain_name { :REUSE } ) (, COMMIT=TRUE/FALSE) (, ROOT=*) (, ACCEPT=A) (, REJECT=U+D) (, IGNORE=S)
```

IMPORT Keywords

The following table lists and defines the keywords for the verb, IMPORT.

RadDBUtil IMPORT Keywords

Keyword	Explanation
INPUT	This is the prefix (the fully specified drive and path) of the input files. • If a fully specified drive and path is not provided with the input file name, the current directory is used. • If <code>.xpi</code> is specified, it will be stripped off. • This keyword is mandatory; it does not have a default value. • The associated resource file must be in the same location and have the same name, but with the extension <code>.XPR</code>. This is the only location that will be searched if any of the import Instances require resource data. • If the fully specified drive and path contains blanks or other special characters, it is necessary to enclose in quotation marks the entire file identifier, including the drive letter and all directories.
OUTPUT	This is the prefix of the output files (the merged “accepts” and “ignores”). Specify a filename to be used for the result of importing this and any modifications to the input media that were required due to events such as duplicate OIDs, different domains, different parentage, etc.
	• This value is required if any of the output of the modified XPI and XPR decks is required. It defines the directory and name, in XPI and XPR, of the generated, modified output. • If <code>.xpi</code> is specified, it will be stripped off. While this keyword is optional, there is no default—if it is omitted or specified without a valid value, no output files other than logs will be available after the completion of RadDBUtil.
COMMIT	Specifies whether the CSDB will be updated. <pre>-commit true</pre> allows those elements that meet the import criteria to be passed into the CSDB for processing. <pre>-commit false (the default)</pre>

	forces RadDBUtil to only scan and reconcile the input files and the database.
ROOT	Specifies the root instance of the model (either FILE.DOMAIN.CLASS, .INSTANCE, DOMAIN.CLASS.INSTANCE, CLASS.INSTANCE, or CLASS). - Only matching instances and those that are referenced (directly or indirectly) by root will be processed. - The default is *. - COMPONENT classes and instances cannot be specified.
DOMAIN	Specifies an existing CSDB domain into which this import deck is to be placed. - If the specified value does not correctly identify an existing domain in the target database, RadDBUtil will end with an error. If a domain is specified, the input files will be imported into that CSDB domain. However, if no domain is specified (the default), the input files will be imported into the domains that are specified in the decks. - All of the instance and resource data that are contained in the INPUT-specified decks will be examined and considered for import into the single CSDB domain that is specified by this keyword. - Domain names in the CSDB are limited to 32 bytes. The <code>-domain</code> switch is designed to put the contents of the input decks into a single destination. Therefore, when importing a Patch Manager bulletin, do not use this switch because if the instances are coming from multiple domains they must be imported back into multiple domains.
REUSE	Specify as: <code>-domain domain_name : REUSE</code> This is an optional directive that allows RadDBUtil to compare the target database and import deck for matching package instances and, when present, re-use those that are in the database. This allows RadDBUtil to avoid creating duplicates of package instances and resources in the target domain if they already exist, in either the original source or target domain. <i>For each non-root and non-component element in the deck:</i> the domain that is specified by the DOMAIN keyword is searched first, followed by the domain that is identified in the package instance of the import deck. If a package instance has the <i>same name</i> as the package instance that is found in the CSDB, RadDBUtil will: - ABORT if it is determined that the contents are different from that which is in the import deck. - REUSE it if it is determined that the contents are identical to that which is in the import deck. This means that the existing database package instance will be referenced by the appropriate connection attributes in the import deck, and the matching Package Instances in the import deck will be removed. While the REUSE option might remove duplicates from the import process, the output files (<code>INPUT.XPI</code> and <code>INPUT.XPR</code>) will contain the instances and resources that were in the original import deck, and which could have been used to update the database but were deferred as a result of this option.
ACCEPT, REJECT,	The instances in the import deck are compared to the instances in the database. These optional keywords dictate the action—based on that comparison—that

and IGNORE	RadDBUtil is to take on the import deck and database instances. There are four instances <i>types</i>. They are: • Adds (A) are instances that are in the import deck and are to be added to the database. The default behavior is to ACCEPT these additions. • Deletes (D) are instances that are to be deleted from the database. The default behavior is to REJECT these deletions. • Sames (S) are instances in the import deck that are identical to those in the database. The default behavior is to IGNORE these instances. • Updates (U) are instances in the database that will be updated by a matching instance in the import deck. The default behavior is to REJECT these instances. These actions (ACCEPT, REJECT, and IGNORE) are applicable only to these four instance types (A, D, S, and U), and act on only those instance types that have been specified for each. The following points are additional items for consideration when using these keywords. • An instance type cannot be specified for multiple actions in a single RadDBUtil execution. That is, Adds cannot be configured to be accepted <i>and</i> ignored. • Multiple instance types can be specified for one action (as in, <code>-reject d+u</code>). If multiple instance types are specified for an action, a plus sign (+) must separate them. • Any operations that match the REJECT parameters will cause the tool to not commit, and will return a non-zero return code when processing has completed.

MSI Files

RadDBUtil has an option that allows an import error to be overruled if inconsistent MSI files are discovered. This might occur if the package with the materials matching the IDX file has been renamed (and, as a result, cannot be found by either import or tree export), or where the ACP file might not yet have been imported into the database, but is in a set of imports that is to be processed subsequent to the current import materials.

The command to overrule the import error is IGNORE=BADMSI.

Output Files

This section provides information about the log files that will be automatically generated by the IMPORT verb. It also discusses additional logs and files, and the conditions under which they might be generated.

- **Standard Files**
 - **RADDBUTIL.LOG** located in the directory specified by DIRECTORY in the MGR_LOG section of the edmpof file. This is a text file that records the actions (activity log) taken by RadDBUtil in processing each command invocation, and the results of these actions. Each execution of RadDBUtil overwrites the previous log.

Note: To save the logs of previous RADDBUTIL.EXE executions, rename the (RADDBUTIL.LOG) file.

This log queries the `edmpprof` file and identifies the location of the `edmpprof` file, the CSDB, and the Configuration Server log. It also contains return codes and summary information about execution results.

- **RADDBUTIL.AUDIT.LOG** located in the directory specified by `DIRECTORY` in the `MGR_LOG` section of the `edmpprof` file. This file contains a record of all RadDBUtil calls and the corresponding return codes; it is designed for archival reference only. Located in the directory specified by `DIRECTORY` in the `MGR_LOG` section of the `edmpprof` file. This file contains a record of all RadDBUtil calls and the corresponding return codes; it is designed for archival reference only.
- **STDERR** contains the same information as **RADDBUTIL.LOG** but, by default, is directed to the console. This log can be redirected as desired.
contains the same information as **RADDBUTIL.LOG** but, by default, is directed to the console. This log can be redirected as desired.

- **Conditional Files**

- XPR and XPI Files**

- If RadDBUtil import updates the database, these two files will be created in the directory that is optionally specified by the keyword `OUTPUT`. The contents of these files can be returned to the customer's **Digital Source Library** (DSL) as a record of the materials as imported into the target CSDB.

- If RadDBUtil import updates the database, these two files will be created in the directory that is optionally specified by the keyword `OUTPUT`. The contents of these files can be returned to the customer's **Digital Source Library** (DSL) as a record of the materials as imported into the target CSDB.

EXPORT

Caution: It is recommended to create a backup of the CSDB before executing any of the commands that are shown in this section.

The `EXPORT` verb allows for the simultaneous exporting of the XPC, XPI and, optionally, the XPR decks that are needed to ensure that the exported portions are accurately reproduced in another database. This includes class templates, instances and, optionally, the resources. It also allows the specifying of the entire database, or individual parts of the database (such as domain, class, instance, package, and service) to be exported. `EXPORT` offers the ability to:

- Perform deletions based on the results of object resolution.
- Specify multiple inputs, such as exporting four services on one export operation.

Additionally, exporting can include:

- The associated resources, and
- All required packages (similar to a client resolution).

Syntax

The syntax of the EXPORT verb is shown below. For examples of the syntax, see ["IMPORT and EXPORT Examples" on page 680](#).

Note: Optional keyword-value combinations are in parentheses. Default values are underlined>.

```
RADDBUTIL EXPORT (, SUBSTITUTE=TRUE|FALSE) (, DATA=TRUE|FALSE)
(, WALK=TRUE|FALSE) (, OUTPUT=stemname) INPUT
```

```
Raddbutil export (-substitute 0|1) (-data 0/1) (-walk 0/1) (-
outputstemname) input
```

EXPORT Keywords

The following table lists and defines the keywords for the EXPORT verb.

RadDBUtil EXPORT Keywords

Keyword	Explanation
DATA	Indicates whether to export the resource files in the CSDB. The default is 0 (FALSE, NO) .
WALK	Indicates whether to do a resolution—CSDB should be traversed. The default is 1 (TRUE, YES) . <i>INPUT is not a keyword, like the others; it is a documentation placeholder. A value must be specified without “input” being used as an indicator, as shown in this table.</i>
OUTPUT	This is the prefix of the output files. - If <code>.xpi</code> is specified, it will be stripped off. - This keyword does not have a default value.
SUBSTITUTE	Indicates whether to do a <i>variablized substitution</i> . The default is 1 (TRUE, YES) . It is used while resolving the variablized substitutions when exporting. <code>-substitute true</code> (the default): variablized substitution done to <code>*</code>, export all <code>-substitute false</code>: variablized substitution done to <code>_UNKNOWN_</code>, export none For example, when <code>-substitute true</code> : <code>PRIMARY.SOFTWARE.ZSERVICE.A001&(ZCONFIG.ZHDWCOMP)</code> would resolve to <code>PRIMARY.SOFTWARE.ZSERVICE.A001*</code>
INPUT	This, the only mandatory parameter, indicates the CSDB Instances that are to be exported. <i>INPUT is not a keyword, like the others; it is a documentation placeholder. A value must be specified without “input” being used as an indicator, as shown below:</i> <code>Raddbutil export (-data 0/1) (-walk 0/1) (-output <u>stemname</u>) file.domain.class.*</code>
	- The format can be either: - FILE.*.*.*, - FILE.DOMAIN.*.*,

	■ <code>FILE.DOMAIN.CLASS.*</code>, ■ <code>FILE.DOMAIN.CLASS.INSTANCE</code>, or ■ <code>FILE.DOMAIN.CLASS.INSTANCE (msg)</code>. • Wildcards (*) are valid values for the domain, class, and instance keywords. • More than one database instance can be specified; multiples must be separated a blank space. The INPUT value must be specified at the end of the command line; otherwise the operation will fail.

Output Files

The EXPORT verb will always generate XPC and XPI files. If -data is specified, an XPR file will also be generated.

DELETE

Note: It is recommended that you back up the CSDB before running any of the commands that are shown in this section.

This verb deletes instances and resources from the CSDB. It offers the ability to perform deletions based on:

- The results of object resolution.
- The contents of an XPI file.

Note: The XPI file that is passed to RadDBUtil when using this verb must be an XPI file that was exported from the CSDB from which the instances and/or resources are to be deleted.

Syntax

The syntax of the DELETE verb is shown below. For examples of the syntax, see ["IMPORT and EXPORT Examples" on page 680](#).

Note: Optional keyword-value combinations are in parentheses. Default values are in underlined>.

```
RADDBUTIL DELETE (, PREVIEW=TRUE|FALSE) (, FILE=value) (, WALK=TRUE|FALSE)
(, IGNORE=value), INPUT
```

```
Raddbutil delete (-preview 1|0) (-file value) (-walk 1|0) (-ignore value)
input
```

```
RADDBUTIL DELETE (, PREVIEW=TRUE|FALSE) (, FILE=value) (, WALK=TRUE|FALSE)
(, LOGFILE=value) (, IGNORE=value), INPUT
```

```
Raddbutil delete (-preview 1|0) (-file value) (-walk 1|0) (-logfile value)
(-ignore value) input
```

DELETE Keywords

The following table "RadDBUtil DELETE Keywords" lists and defines the keywords for the verb, DELETE.

RadDBUtil DELETE Keywords

Keyword	Explanation
PREVIEW	Indicates whether to preview the changes only, or make the deletions. The default is 0 (FALSE, NO, OFF) .
FILE	The name of XPI file that specifies what is to be deleted from the CSDB.
WALK	Indicates whether to do a resolution—CSDB should be traversed. The default is 1 (TRUE, YES, ON) .
LOGFILE	The name of the custom log file where logs will be written. If LOGFILE parameter is not added, the logs are written to the default log file.
IGNORE	This keyword specifies (in) which CSDB instances should not be deleted. - The format can be either F.D.C.I or f.d.c.i format. - Wildcards (*) are valid in the domain, class, and instance specifications. - More than one database instance can be specified; multiples must be separated a plus sign (+).
<i>INPUT</i>	This parameter indicates which CSDB instances are to be deleted. <i>INPUT is not a keyword, like the others; it is a documentation placeholder. A value must be specified without "input" being used as an indicator, as shown below.</i> <pre>Raddbutil delete file.domain.class.*</pre> - The format can be either: - FILE.*.*.*, - FILE.DOMAIN.*.*, - FILE.DOMAIN.CLASS.*, - FILE.DOMAIN.CLASS.INSTANCE, or - FILE.DOMAIN.CLASS.INSTANCE (msg) . - Wildcards (*) are valid in the domain, class, and instance specifications. More than one database instance can be specified; multiples must be separated a blank space.

Deleting Bulletins from a Database

To permanently delete Patch Manager bulletins from the Configuration Server Database, specify the following classes with the IGNORE keyword.

- PRIMARY.PATCHMGR.CMETHOD
- PRIMARY.PATCHMGR.OPTIONS
- PRIMARY.PATCHMGR.METADATA

- PRIMARY.SYSTEM.ZMETHOD
- PRIMARY.SYSTEM.PROCESS
- PRIMARY.PATCHMGR.PRODUCT

PRIMARY.PATCHMGR.SP

- PRIMARY.PATCHMGR.RELEASE
- PRIMARY.PATCHMGR.PATCHARG
- PRIMARY.PATCHMGR.PG2PR
- PRIMARY.PATCHMGR.PROGROUP

The following example shows these classes being included with the IGNORE option. Make sure that there is no space between * and + when you run this command.

```
Raddbutil.exe delete -walk 1 -ignore  
PRIMARY.SYSTEM.PROCESS.*+PRIMARY.SYSTEM.ZMETHOD.*  
+PRIMARY.PATCHMGR.CMETHOD.*+PRIMARY.PATCHMGR.METADATA.*  
+PRIMARY.PATCHMGR.OPTIONS.*+PRIMARY.PATCHMGR.PATCHARG.*  
+PRIMARY.PATCHMGR.PRODUCT.*+PRIMARY.PATCHMGR.RELEASE.*  
+PRIMARY.PATCHMGR.SP.*+PRIMARY.PATCHMGR.PG2PR.*  
+PRIMARY.PATCHMGR.PROGROUP.*  
-preview 0 PRIMARY.PATCHMGR.ZSERVICE.MS07-042 (SYNC)
```

RCS

This verb communicates with the CSDB and allows for:

- Querying of the CSDB **lock status**.
- Unlocking of the CSDB.

Syntax

The syntax of the RCS verb is shown below. For examples of the syntax, see the section, ["IMPORT and EXPORT Examples" on the next page](#).

```
raddbutil rcs status  
raddbutil rcs unlock
```

RCS Keywords

The following table "RadDBUtil RCS Keywords" lists and defines the keywords for the verb, RCS.

RadDBUtil RCS Keywords

Keyword	Explanation
STATUS	Displays the lock status of the Configuration Server status.
UNLOCK	Unconditionally unlocks the Configuration Server.

Return Codes

The following table shows the return codes that are associated with the RadDBUtil executable.

RadDBUtil Return Codes

Return Code	Meaning
0	SUCCESS
4	WARNING
8 or 16	FAILURE No database update occurred, or A database update was started but not completed. Note: If the latter, the database might be in an error state.

IMPORT and EXPORT Examples

This section presents a few import and export examples of the simpler and more direct **RADDBUTIL.EXE** syntax.

Note: As previously stated, the **RADDBUTIL.EXE** utility supports the following two syntax formats:

`-keyword value`

and

`keyword=value`

The examples in this section are presented in the `-keyword value` format.

Performing a Simple Import

Run a routine, daily import of the file, `sample.xpi` to add instances to the CSDB.

```
raddbutil import -input sample -commit yes
```

Stop if there are updates and/or deletes.

```
raddbutil import -input sample -accept A -reject U+D-commit yes
```

Ignore any updates and deletes.

```
raddbutil import -input sample -accept A -ignore U+D-commit yes
```

Results:

- The instances and resource data from the files `sample.xpi` and `sample.xpr` are imported directly into the domain and class that are specified in the input deck.
- No domain mapping is performed

Performing a Simple import and Replacing the Old Instances

```
raddbutil import -input sample -accept A+U+D -commit yes
```

Importing to a Domain Other Than SOFTWARE

Import all instances from the input deck into the SOFT0002 domain. Re-use any database elements that are identical to elements of the deck, overrule the inconsistent MSI file import error, and reject any updates.

Example 3a

Note: The syntax of the following example is fully supported by RadDBUtil.

```
Raddbutil Import -Input Sample -Domain SOFT0002:Reuse-Commit Yes -  
Ignore Badmsi
```

Example 3b

Note: The following example is the same as Example 2a, but includes the fully specified drive and path (with special characters) of the input material, and the name of the output decks.

```
RADDBUTIL IMPORT -INPUT "G:\CONTAINS BLANKS\SAMPLE"-OUTPUT Sample_  
Soft_2 -DOMAIN SOFT0002:REUSE -COMMIT YES -IGNORE BADMSI
```

An Easy Method to Create Export Files

The following command will create `AMORTIZE.XPC` and `AMORTIZE.XPI` containing just the specified class and instance.

```
raddbutil export -output amortize PRIMARY.SOFTWARE.ZSERVICE.AMORTIZE
```

Using the -walk Command

The following command will create `AMORTIZE.XPC` and `AMORTIZE.XPI` containing the classes and instances. This command will resolve the package, and include any other required packages because `-walk` is specified.

```
raddbutil export -output amortize -walk 1  
PRIMARY.SOFTWARE.ZSERVICE.AMORTIZE
```

Note: No resources will be exported.

Using the -walk and -data Commands

Example 6a

The following command will create `AMORTIZE.XPC`, `AMORTIZE.XPI`, and `AMORTIZE.XPR` which contain the classes, instances, and resources (because `-data` is specified), and the package will be resolved because `-walk` is specified.

```
raddbutil export -output AMORTIZE -walk 1 -data 1  
PRIMARY.SOFTWARE.ZSERVICE.AMORTIZE
```

Example 6b

The following command will create ALL.XPC, ALL.XPI, and ALL.XPR containing the classes, instances, and resources of all services in the SOFTWARE Domain.

```
raddbutil export -output ALL -walk 1 -data 1  
PRIMARY.SOFTWARE.ZSERVICE.*
```

Exporting and Importing the PRIMARY File (ZEDMAMS Used to Import the Class)

The following series of commands will export and then import the entire PRIMARY file of the CSDB.

```
raddbutil export -output ALL -walk 1 -data 1 PRIMARY.*.*.*  
  
zedmams verb=import_class,file=ALL.xpc,preview=no,logfile=ALL_  
PRIMARY.log  
  
raddbutil import -input ALL -commit yes
```

Exporting, Deleting, and Importing a Bulletin

The following commands will export, delete, and import (respectively) the Patch Manager bulletin, MS07-042.

Export

```
./raddbutil export -walk 1 -data 1 -output ms07-042  
"PRIMARY.PATCHMGR.ZSERVICE.MS07-042 (SYNC) "
```

Delete

```
raddbutil.exe delete -walk 1 -ignore  
PRIMARY.SYSTEM.PROCESS.*+PRIMARY.PATCHMGR.ZMETHOD.*  
+PRIMARY.SYSTEM.CMETHOD.*+PRIMARY.PATCHMGR.METADATA.*  
+PRIMARY.PATCHMGR.OPTIONS.*+PRIMARY.PATCHMGR.PATCHARG.*  
+PRIMARY.PATCHMGR.PRODUCT.*+PRIMARY.PATCHMGR.RELEASE.*  
+PRIMARY.PATCHMGR.SP.*+PRIMARY.PATCHMGR.PG2PR.*  
+PRIMARY.PATCHMGR.PROGROUP.* PRIMARY.PATCHMGR.ZSERVICE.MS07-042 (SYNC)
```

Import

```
raddbutil import -input ms07-042 -commit yes
```

Note: In the import example, the **-domain** switch was not used because the instances are coming from multiple domains.
See the warning under DOMAIN in ["RadDBUtil IMPORT Keywords"](#) on page 672.

The DELETE Verb

This series of example focuses on the DELETE verb commands.

- Delete the service PRIMARY.SOFTWARE.ZSERVICE.MSOFFICE.
`raddbutil delete PRIMARY.SOFTWARE.ZSERVICE.MSOFFICE`
- Delete the service PRIMARY.SOFTWARE.ZSERVICE.MSOFFICE; do not delete the methods.
`raddbutil delete -walk 1 -ignore PRIMARY.SYSTEM.ZMETHOD.*
PRIMARY.SOFTWARE.ZSERVICE.MSOFFICE`
- Delete the instance PRIMARY.SOFTWARE.ZSERVICE.MSOFFICE.
`raddbutil delete -walk 0 PRIMARY.SOFTWARE.ZSERVICE.MSOFFICE`
- Delete the instance PRIMARY.TD1.ZSERVICE.MSOFFICE and write the log to the `user_log.log` file with increased logging level
`raddbutil delete -preview 0 -walk 1 -loglvl 9 -logfile user_log.log
PRIMARY.TD1.ZSERVICE.MSOFFICE`
- Delete the contents of `foo.xpi`.
`raddbutil delete -file foo.xpi`
- Delete the content of the `foo.xpi` and write the logs to the `user_log.log` file
`raddbutil delete -file foo.xpi -preview 0 -walk 1 -loglvl 9 -logfile
user_log.log`

The Configuration Server Database

The following commands pertain to the Configuration Server Database.

- Show the current status of the CSDB.
`raddbutil rcs status`
- Unlock the CSDB.
`raddbutil rcs unlock`

Configuration Server Performance

This section describes about the configuration server performance.

Performance Issues

The purpose of this chapter is to discuss system performance issues as they relate to the Configuration Server. The next chapter, explores some problem determination issues.

Performance issues are associated with enhancing the efficiency of a working system, while troubleshooting deals with features, functions, and components that are not operating as expected. Taking into account performance and usage considerations before configuring the Configuration Server might prevent many of the conditions that require troubleshooting.

The Configuration Server is a multi-processing server framework for:

- Policy Management
- Component Management
- Network Management
- Version Management
- Asset Management
- State Management

There are many aspects of Configuration Server performance. In addition, there are specific phases of Configuration Server operations. Each phase has different performance characteristics and requirements. Because of these many variables, there is no easy “cook book” approach to Configuration Server performance.

General Performance and Usage Considerations

Three important performance and usage considerations must be taken into account before beginning any discussion of Configuration Server issues.

- *What is the overall system infrastructure?*
This includes numbers, types, and speeds of processors; total size and type of memory; and network capability and configuration.
- *What are the performance benchmarks?*
These include average performance levels, as well as the high and low levels.
- *What are the workload parameters?*
These include average demand, peak load requirements, and idle times.

Before undertaking any further performance initiatives, become familiar with the above considerations as they apply to your Configuration Server.

CPU Requirements

There are minimum CPU requirements specified at installation for each Configuration Server platform. It must be noted, however, that these are *minimum values*. The real requirements for CPU utilization can only be determined by workload—essentially, the number of resolutions that a Configuration Server can process.

CPU and Object Resolution

As each Sentient agent connects to the Configuration Server, an identifier object (ZMASTER) is sent from the Sentient agent to the Configuration Server triggering the dynamic construction of an object model for that Sentient agent. This process is known as *object resolution*. The object resolution process exhausts most of the processor time required by the Configuration Server.

When trying to determine how many object resolutions can take place simultaneously, use the simple formula outlined below.

Total Number of Possible User Resolutions = Total Number of Available CPU Seconds/Total Number of CPU Seconds Required per User

Total Number of Available CPU Seconds

This value is obtained by multiplying the number of CPUs by the number of seconds in the connection window of opportunity. The window of opportunity is the timeframe in which the Sentient agents need to connect.

For example: a two-processor machine with a six-hour window of opportunity (e.g., 12:00 AM - 6:00 AM) would result in a total number of available CPU seconds of 43,200. (2 processors X 6 hours (21,600 seconds) = 43,200 seconds).

Total Number of CPU Seconds Required Per User

By taking the average number of FILE objects per user (ZRSOURCE being the most common object in a user's model) and multiplying it by three (an estimate of how many objects are resolved to end up with a fully resolved ZRSOURCE), the average number of objects to be resolved per user is fetched. This is divided by the number of objects the Configuration Server can resolve in a CPU second, and the number of CPU seconds required to resolve the average Sentient agent's object model is determined.

For example, let us assume that the average number of ZRSOURCE objects per user in your environment is 1000. Multiply that by 3, and get 3000 objects per user. Now divide 3000 by 860 (the average number of objects resolved per second by the Configuration Server), and you get approximately 3.5 seconds of CPU time required to resolve the average user's model.

$$1000 \times 3 = 3000$$
$$3000/860 = \sim 3.5 \text{ (3.488...)}$$

Total Number of Possible User Resolutions

This value is obtained by dividing the (*Total Number of Available CPU Seconds*) by the (*Total Number of CPU Seconds Required per User*). Using the results of our previous examples, divide the 43,200 (available CPU Seconds based on a two-processor machine with a six-hour window) by 3.5 (number of CPU seconds required to resolve the average user's model) resulting in a (*Total Number of Possible User Resolutions*) of approximately 12,000.

$$43,200/3.5 = \sim 12,000 \text{ (12,342.8571...)}$$

Note: This calculation does not mean that 12,000 users could be successfully configured in the six-hour period. Other variables must be considered, such as disk I/O for data being transferred/received to/from Sentient agents, the platform's network card capability (for concurrent communications between the Configuration Server and Sentient agents), and the system's available memory (process/memory swapping requires system overhead). Also, note that other tasks running on the machine will be sharing system resources with the Configuration Server.

Memory

There are two features that are related to the memory usage, **content caching** and **index caching**. They are established in the MGR_CACHE section of the Configuration Server `edmpprof` file.

Content Caching

refers to loading a portion of the CSDB (class templates, base instances, and other instances) into memory to speed up the resolution process. This enhances performance by eliminating disk I/O. In configuring index caching, the size used for each content cache entry needs to reflect the size of the instance in the database before any resolution has been performed. This provides a resolution boost across the Configuration Server.

² refers to loading a portion of the CSDB (class templates, base instances, and other instances) into memory to speed up the resolution process. This enhances performance by eliminating disk I/O. In configuring index caching, the size used for each content cache entry needs to reflect the size of the instance in the database before any resolution has been performed. This provides a resolution boost across the Configuration Server.

² **Index Caching** is used to keep in memory all names of the instances of the class that have been cached. Caching the names of all instances of cached classes eliminates the need to read the directory to determine which instances begin with the specified prefix. Index caching makes a significant performance improvement when the general resolution feature is utilized. General resolutions are those that use a partially specified CLASS.INSTANCE naming format that terminates in an asterisk (*), indicating that all instances with the same prefix are to be resolved.

is used to keep in memory all names of the instances of the class that have been cached. Caching the names of all instances of cached classes eliminates the need to read the directory to determine which instances begin with the specified prefix. Index caching makes a significant performance improvement when the generic resolution feature is utilized. Generic resolutions are those that use a partially specified CLASS.INSTANCE naming format that terminates in an asterisk (*), indicating that all instances with the same prefix are to be resolved.

MGR_CACHE

The MGR_CACHE section of the Configuration Server `edmpprof` file defines the values that determine how much **virtual storage** is reserved for **content cache**. The two values are `CACHE_SEGMENTS`—which determines the number of separate memory areas—and `CACHE_SIZE`—which is allocated at startup and used exclusively for content cache. The product of `CACHE_SEGMENTS` x `CACHE_SIZE` is the amount of memory that will not be available for resolution purposes during connection.

In a Windows environment, the maximum virtual storage that will be available for any single process is 2 GB. If the Windows Enterprise Server is used, this is increased to 3 GB, and for Windows 2003 Server x64, up to 4 GB of virtual memory per 32-bit process, while it might be constrained by the size of real memory and the size of the page space available.

The Configuration Server process will attempt to use all of the virtual storage that is available to it and might cause all of the defined page space to be in use, so make sure that the total page file

in this section should be set to the size of the largest Instance of the Classes being cached. The default is **2048** bytes.

A parameter called ICACHE_SIZE is available for **index caching**. It is enabled by simply specifying a value for the keyword. The easiest means by which to correctly size this is to take the total of all instances to be cached, multiply by 100, and place the result as the ICACHE_SIZE value size. ICACHE_SIZE is of benefit when general resolution is active, that is, any connection to SOFTWARE.PACKAGE.* that requires the Configuration Server to process all of the potential Instances prefixed by the string. While ICACHE is most important for SOFTWARE.FILE, the caching mechanism (described below in MGR_CLASS) uses the same criteria for selecting which DOMAIN.CLASS Instances to cache.

MGR_CLASS

The MGR_CLASS section controls two separate processes: initial Classes to be cached and the amount of storage to be used for in-storage objects during resolution of each Sentient agent (as differentiated from CSDB Classes and Instances where the Class name might be the same as the in-storage object name, as is the case of ZSERVICE). In-storage objects of interest are generally persistent and multi-heap. Controlling the storage and processing of these objects provides for performance improvements.

For index caching and content caching, the contents of MGR_CLASS are processed in the order in which they are presented, so the first DOMAIN.CLASS is processed completely (index cache is loaded and content cache is loaded) before the second, and so on.

For each DOMAIN.CLASS (for example, SOFTWARE.FILE) that is identified in this section, four parameters are specified. The first and second control the caching behavior, and the third and fourth are used exclusively for persistent object virtual storage allocation during resolution.

For a detailed explanation of the MGR_CLASS settings, including performance and usage considerations, see ["MGR_CLASS" on page 511](#). The first of the four parameters (Value1) allows one to specify whether the Class template and _Base_Instance_ are to be cached. A value of Y is recommended because it is generally necessary to load the Class template and _Base_Instance_, and this eliminates a tremendous amount of disk I/O for Classes that are commonly involved in resolution.

Networking

This section describes about networking.

Bandwidth Throttling

Bandwidth throttling refers to reserving a percentage of the available TCP/IP bandwidth for use by other processes on the device. It was designed to help maximize network resources while running the Configuration Server. Bandwidth throttling is configured in the Configuration Server using the SEND_THROTTLE setting of the MGR_TIMEOUT section of the `edmprof` file. It specifies the number of milliseconds that the Configuration Server will wait before sending packets. The default is **0**, meaning no delay. The range of values is 0 to 4 GB.

There are three variables in the Sentient agent's ZMASTER object that also have an effect on bandwidth throttling, ZBWMGR, ZBWMAX, and ZBWPCT.

- ZBWMGR=YES means the Configuration Server will be controlling the bandwidth.
- ZBWMAX is the maximum speed (bytes/second) of the “sends.”
- ZBWPCT is the percentage of the maximum to use (0–100).

Note: The ZBWMGR, ZBWMAX, and ZBWPCT values will override the SEND_THROTTLE setting.

Distributed Configuration Server

The Distributed Configuration Server (DCS) is used to synchronize the Configuration Server Database (CSDB) *contents only* and forms a subset of the normal synchronization process of the Core to Satellites. The DCS components are pre-configured with the Core and Satellite installation. You do not have to change these configurations or run the DCS Destination Component (`dmabatch`) to synchronize the CSDB contents. DCS is a tool that enables an administrator to manage multiple CSDBs in a networked environment.

An administrator can use DCS to replicate domains from one CSDB to another. This means that enterprise-wide changes can be made once, on one Configuration Server machine, then simply distributed to (replicated on) multiple Satellite Servers within the enterprise. This ability offers several benefits:

- **Efficiency**
Less time and manpower are required for making the database changes.
- **Reliability**
Increased control over the integrity of the changes that are made.

DCS is a two-piece extension of the Configuration Server. The components—**Source** and **Destination**—function separate from, but in conjunction with, one another. Both, however, have some dependence on a Configuration Server.

Note: For more information on the functionality of the DCS components, see the section, ["DCS Components" below](#).

In a **multi-tier** configuration, both components can be installed on the same machine to accommodate *peer synchronizations* via a *middle-tier Configuration Server*. (Peer synchronizations and middle-tier Configuration Servers are defined in the table Distributed Configuration Server Terminology.)

DCS is designed to synchronize DCS-enabled Configuration Server Databases throughout an enterprise so, although it is not essential that the CSDBs directly communicate with one another, DCS must be able to communicate with both Configuration Servers that consists of the synchronization pair (see ["A Synchronization Pair" on page 690](#)).

In a synchronization, DCS compares the control information of one CSDB with that of another, for the domains that have been selected.

Note: If SSL functionality is used during a DCS synchronization, files larger than 2 GB will fail.

DCS Components

In as much as there are two Configuration Servers involved in DCS synchronizations, the two DCS components perform different functions and must be installed separately, based on the intended role of the host Configuration Server.

- Each Configuration Server that will act as a Source must have the DCS *Source* component installed.
- Similarly, each Configuration Server that will act as a Destination must have the DCS *Destination* component installed.
- If a Configuration Server has both components of the DCS installed, it can act as Source and Destination, albeit in separate DCS operations.

With the DCS components installed on the Configuration Server machines, DCS:

- Provides the synchronization facilities to contact the Source and Destination,
- Reconciles the differences between the selected domains, and
- Provides the intermediate facilities to make identical the Source and Destination domains.

Note: The Destination is always a replica of the source.

The Source and Destination components perform different functions during the Distributed Configuration Server synchronization. Therefore, it is important to correctly install these components to ensure: 1) the availability and accessibility of the appropriate Source-Destination synchronization pairs, and 2) the expected synchronization results.

Source Component

The Source component must be installed on any Configuration Server that is going to function as the master in a synchronization. This component contains the Tcl web server, the product suite's HTTP server. For a brief description of Tcl web server and how it relates to DCS, see "[Tcl Web Server](#)" below.

The Source component loads the database instances into a single repository. This repository can be directly accessed, thereby eliminating the excessive overhead of opening, storing, transferring, and writing individual files for each CSDB instance.

Tcl Web Server

Tcl Web Server is the Client Automation product suite's HTTP file server. It facilitates multiple concurrent file transfer sessions (HTTP "get" requests) and the creation of the instance-container file (see Container File in table "Distributed Configuration Server Terminology").

It integrates several independent modules—**Sentient Enterprise Portal (Portal)** and **Sentient Enterprise Proxy Server (Proxy Server)**—giving them access to all the functions and resources under its control.

Destination Component

The Destination component must be installed on any Configuration Server that is going to function as the target in DCS synchronization. This component provides direct access to the target file system.

A Synchronization Pair

Two Configuration Servers, one defined as the Source and the other as the Destination, consists of a Distributed Configuration Server *synchronization pair*.

Note: Cross-format synchronizations are not supported. To be synchronized, the Source and Destination databases must be of the same format; that is, a **UTF-8** database to a **UTF-8** database, and a **Legacy** database to a **Legacy** database.

DCS accepts one synchronization pair only, per execution. Operationally, because a synchronization can go in only one direction, this means that if two Configuration Servers (for example, MGR_001 and MGR_002) need domains from one another, two DCS executions must be done—with MGR_001 being the Source in one synchronization, and MGR_002 being the Source in the other.

Configuration Server Eligibility

To be eligible to participate in a DCS operation, a Configuration Server must meet the following requirements.

- In its `edmpprof` file, it must be DCS-enabled. This is done by specifying:
`[MGR_STARTUP]MANAGER_TYPE=DISTRIBUTED`

or

`[MGR_STARTUP]MANAGER_TYPE=SERVER`

Note: All Configuration Servers are installed as DISTRIBUTED, so the SERVER value will have to be manually specified in the `edmpprof` file. For performance-improvement information, see ["Improving Performance using MANAGER_TYPE" below](#).

- It must have either the DCS Source or Destination component installed.

Improving Performance using MANAGER_TYPE

Typically, Destination and middle-tier Configuration Servers (see "Middle-tier Configuration Server" in table "Distributed Configuration Server Terminology") get database updates from their up-line Source Configuration Server only. Therefore, some default database administrative processes are not necessary. Disabling these unnecessary processes will improve performance.

- If a DCS-enabled Configuration Server is going to be a Destination or middle-tier Configuration Server—getting its updates from a Source Configuration Server only and not managed via any other process—you can improve its performance by setting:

```
[MGR_STARTUP]  
MANAGER_TYPE=SERVER
```

Domain Ownership

Configuration Server Database domains on each Configuration Server have three distinguishing characteristics: **domain name**, **owning MGR_ID**, and **current MGR_ID**. Their ownership is determined by the value of MGR_ID, and is established:

- When a CSDB is installed.
- When a domain is added to a CSDB.

Note: For planning purposes, it is recommended to maintain unique names for CSDB domains.

A self-owned domain is a CSDB domain that is owned by the current Configuration Server. The owning MGR_ID and current MGR_ID are the same.

A foreign-owned domain is a CSDB domain that is owned by a Configuration Server other than the current one, and which is present as the result of a DCS synchronization. The owning MGR_ID and current MGR_ID are not the same.

Domain Naming Considerations

To minimize the likelihood of synchronization problems, consider the following points when creating domain names and configuring synchronizations.

- A Configuration Server cannot contain two domains with the same name.
- A Configuration Server cannot obtain one of its self-owned domains from a Configuration Server that foreign-owns the domain. For example, MGR_001 cannot receive from another Configuration Server any domain for which it (MGR_001) is listed as the owning MGR_ID.

Note: The version that is resident at the owner is always considered the current and correct copy. Its contents will always supersede and replace any changes introduced by other Configuration Servers.

One Owner vs. Multiple Owners

When planning domain ownership, it is helpful to consider whether to assign the proprietorship of all the domains to one Configuration Server, thereby centralizing control; or to disperse control by establishing domain ownership at several Configuration Servers at various, strategic points across the enterprise.

The tables in this section detail the advantages and disadvantages of each method. For additional planning considerations, see the following table.

One Owning Configuration Server

The following table lists the benefits and drawbacks of one Configuration Server owning all the domains.

One Domain-Owning Configuration Server

Advantages	Disadvantages
Control of all applications, access rules, and users	Central control might make the database very large, depending on the organization and structure
One Configuration Server Database to backup	Central control might make the database very large, depending on the organization and structure
Data flow throughout the environment is one-way	Data flow throughout DCS is one-way
Aligns with highly centralized organizations	

Multiple Owning Configuration Servers

The following table lists the benefits and drawbacks of domain ownership being assigned to multiple Configuration Servers.

Multiple Domain-Owning Configuration Servers

Advantages	Disadvantages
Aligns readily with highly de-centralized organizations	Does not align well with highly centralized organizations
Databases are smaller and indicative of regional Source Configuration Servers	Multiple Configuration Servers must be administered and backed-up
Applications and users can be managed locally	Enables two-way data flow, adding complexity to the DCS design
Corporate or common information can be managed centrally, while local information is managed locally	
Enables two-way data flow between central and local Configuration Servers	

Note: Any Configuration Server with self-owned domains should be backed up. Foreign-owned domains can always be obtained through synchronization with the owning Configuration Server.

Domain Eligibility

The list of domains that are eligible for synchronization is dynamically compiled by Distributed Configuration Server. This list is based on the chosen synchronization pair and:

- The database control information about the most recent synchronization for the synchronization pair, or
- The most recent update with administrative components (such as **Sentient Enterprise Administrator Configuration Server Database Editor (CSDB Editor)**, and **Sentient Enterprise Administrator Publisher (Publisher)**).

Only domains that have the same owner (on the Source and Destination) can be synchronized between that pair of Configuration Servers.

Selecting Domains

It is not necessary to synchronize all eligible domains between two DCS-enabled Configuration Servers. At the start of each session, an administrator can specify which of the eligible domains are to be synchronized.

Domain Eligibility Rules

The primary DCS domain synchronization eligibility rules are listed below. These apply to each domain independently. For log error messages, see the *Sentient Enterprise Troubleshooting Guide*

- Synchronization cannot occur into a self-owned domain.
- There is no replication into an owning Configuration Server.

Note: If a self-owned domain is deleted, it must be restored from a backup; it cannot be replicated from a DCS-enabled Configuration Server on which it is foreign-owned.

- Domains that are not owned by the same Configuration Server are considered *unrelated*. A domain must be owned by the same MGR_ID at the Source and Destination to be eligible for synchronization.
- Once a foreign-owned domain is locally updated with another Client Automation component, it cannot be used as the Source in a *peer synchronization*.

Note: A local update occurs when a CSDB is updated by a Client Automation component (such as CSDB Editor and Admin Publisher) other than Distributed Configuration Server.

- When it is possible to make such a distinction, the DCS will prevent the regression of a more current Destination by a less current peer Source. If the Destination domain has been locally updated, and the relative currency cannot be determined, the synchronization is enabled.

Configuring DCS

Distributed Configuration Server functionality must be configured for two Configuration Servers.

DCS requires a communications connection between the Source and Destination Configuration Servers.

DCS: Batch Mode

The command-line mode (also known as the **batch** mode) of DCS is invoked by the executable, `DMABATCH.EXE`. Once the synchronization is started, it will execute with no administrator action required. This is discussed in more detail in ["DCS Options and DMABATCH" on page 696](#).

Synchronization Logs

When a synchronization is executed, logs and objects are created. Each subsequent run causes its predecessor's logs to be overwritten, so that these logs and objects represent the most recent DCS synchronization.

Simultaneous Synchronizations

A Configuration Server can be simultaneously involved in multiple synchronizations in which it is the Source only. This is possible because a Source database is only being read from, whereas a Destination database is being written to.

- A Configuration Server cannot simultaneously be a Source and Destination for different synchronizations.
- A Configuration Server cannot be the Destination in multiple, simultaneous synchronizations.

The EDMPROF File and DCS Security

Editing the EDMPROF File

The `edmpprof` file is the text file in which the operational parameters of the Configuration Server are configured and stored. Two of its sections—**MGR_STARTUP** and **MGR_DMA**—are integral to enabling DCS and ensuring its proper operation.

Information on these `edmpprof` sections, including their settings, acceptable values, and effect on DCS processing is presented in this section.

Note: For a comprehensive look at the `edmpprof` file, see ["Viewing Configuration Server Settings" on page 500](#).

The `edmpprof` file can be edited in a standard text-editing application.

Caution: Be sure to review the important information in this section before editing this file. Failure to do so could adversely affect your Sentient environment.

It is recommended to back up the `edmpprof` file before editing it.

- You must use a UTF8-aware text editor when editing the `edmpprof` file.
- Be sure to select UTF-8 as the *encoding type* when saving the file.
 - Windows users are advised to use **Notepad**.

Failure to use a recommended text-editing application and saving the file as recommended in this section could result in the file's changes not being correctly applied to the Configuration Server. This could adversely affect your Sentient environment.

MGR_STARTUP Section

The MGR_STARTUP section dictates startup behavior for the Configuration Server. The following table describes the MGR_STARTUP settings are essential to the operation of the DCS.

MGR_STARTUP Settings and Values

Setting	Explanation
MANAGER_TYPE	To ensure that a Configuration Server is DCS-enabled, verify that this value is set to DISTRIBUTED (the default) or SERVER . For additional information, see "Configuration Server Eligibility" on page 690 .
MGR_TYPE	A 32-alphanumeric character (max.) Configuration Server identifier.
MGR_ID	The unique, 3-digit, hexadecimal ID for a Configuration Server. • DCS uses this value to create object IDs in the CSDB. • Each character in this identifier can have the values 0-9 and A-F. Exception: The 256 consecutive positions from F00 through FFF are reserved for use with Client Automation.
TCP_PORT	The port on which the Configuration Server will listen. This must match the port that is specified for DCS communications (<code>-master-port</code> in <code>dmabatch.rc</code>).

MGR_ID

The MGR_ID setting establishes a unique identity for each Configuration Server. All Domains in a Configuration Server Database are *owned* by a Configuration Server—identified by the value of MGR_ID.

Note: It is possible that a Domain is not owned by the Configuration Server that is hosting its database.

DCS uses the value of MGR_ID to determine which Configuration Server owns each Domain.

Domain ownership is important because in order for a Domain to be eligible for synchronization its owning MGR_ID must be the same on the Source and Destination Configuration Servers. If the owning MGR_IDs do not match, synchronization cannot occur.

Note: Although the MGR_ID must match for both Domains, it is possible that neither the Source nor the Destination is the owner.

MGR_DMA Section

You can define a DCS-related setting, DMA_TIMEOUT, which is detailed in the following table.

MGR_DMA Settings and Values

Setting	Explanation of Value
DMA_TIMEOUT	Specify the number of seconds that DCS is to wait for non-DCS tasks to complete before applying a lock to the CSDB. If DCS times out before the task ends, it will stop. The default is 0. • When soft-locking the CSDB, DCS must wait for all administrator tasks to end. • When hard-locking the CSDB, DCS waits for all non-DCS tasks to end.

DCS Options and DMABATCH

DCS Configuration Files

The Distributed Configuration Server (DCS) uses two configuration files. The `dcs.cfg` file contains settings for the DCS Source. The "[dmabatch.rc](#)" [below](#) file contains settings for the DCS Destination.

dcs.cfg

This section details the `SNDBUF` parameter, which can be added to the DCS Source configuration file, `dcs.cfg`. To add this option, open `dcs.cfg` in a text editor.

- `SNDBUF` is used to change the DCS network send buffer size (the default is 32K). If you notice slow DCS download or data transfer speeds, you can set this parameter to 64K. For example:

```
dcs::init {  
  DBPATH "C:/Program Files/PSL/RCA/ConfigurationServer/DB"  
  SNDBUF 64K  
}
```
- If the TCP window scaling feature is enabled with the TCP window size set to greater than 64K, `SNDBUF` can be increased accordingly (for example: 64K, 128K, or 254K). You can select the size that provides the best performance.

dmabatch.rc

The table dmabatch.rc Options describes the settings specified in the DCS Destination configuration file, `dmabatch.rc`. To edit these settings, open `dmabatch.rc` in a text editor.

- All of the options that are important to the basic operation of DCS are populated by values that were specified during the DCS Destination installation programs (see the [Set by Install](#) column in the table dmabatch.rc Options). You only need to modify an option when a non-default value is preferred.

In the dmabatch.rc Options, the following DCS terminology is used.

- Master = the Source Configuration Server
- Slave = the Destination Configuration Server

dmabatch.rc Options

Option	Default Value or Required	Set by Install	Description
-master-host	Required	Y	IP name/address of the master Configuration Server
-master-id	Required	Y	3 hexadecimal-digit ID of the master Configuration Server
-master-port	Required	Y	TCP port of the master Configuration Server (usually 3464)
-master-ssl-port	444	Y	SSL port of the master Configuration Server (usually 444)
-master-timeout	3600	Y	Master Configuration Server request timeout (in seconds)
-slave-host	Required	Y	IP name/address of the slave Configuration Server (local computer name or <i>localhost</i>)
-slave-id	Required	Y	3 hexadecimal-digit ID of the slave Configuration Server
-slave-port	Required	Y	TCP port of the slave Configuration Server (usually 3464)
-slave-ssl-port	444	Y	SSL port of the slave Configuration Server (usually 444)
-slave-timeout	3600	Y	Slave Configuration Server request timeout (in seconds)
-http-port	3466	Y	-http-port 3466 Y TCP port of the IS (usually 3466)

Option	Default Value or Required	Set by Install	Description
-https-port	443	Y	SSL port of the IS (often 443; must not be the same as that of <code>-master-ssl-port</code>)
-domains	" " (null)	Y	Quoted list of blank-separated CSDB Domains; null list means "ALL DOMAINS"
-ssl	0	Y	Enable SSL 1=enable SSL; 0=disable
-userid	DMABATCH	Y	User ID to be used for Configuration Server and/or Tcl web server authentication
-password_cipher	AES	Y	TP password-encryption method. Valid values are AES and DES.
-password	<none>	Y	Password to be used for Configuration Server and/or Tcl web server authentication. Note: This value can be AES-encrypted or cleartext; it must be consistent with the <code>-password_cipher</code> setting.
-dataless	0	Y	Skip resources 1=skip resources; 0=include resources
-loglvl	3	Y	Logging level 2=terse, 3=normal, 4=debug, 5=debug+
-logfile	dmabatch.log	N	Log file name (no pathing)
-logmode	w	N	Overwrite the log file w=overwrite log file; a=append the log file
-loglines	100000	Y	Number of lines to write to the log file before the log rolls over
-logerr	1	Y	Echo log to <code>stderr</code> 1=echo; 0=don't echo
-logpath	<none>	N	Override LOGPATH in <code>nvd.ini</code> (location of <code>dmabatch.log</code>)
-libpath	<none>	N	Override LIBPATH in <code>nvd.ini</code> (location of client objects)
-commit	1	N	Commit database updates 1=commit; 0=don't commit, but remain locked

Option	Default Value or Required	Set by Install	Description
			See also "Deferred Commit" on page 703 .
-report	0	N	Send status-reporting objects to master Configuration Server 1=send; 0=don't send
-dmastats-userid	DMA_ <masterid>_ <slaveid>_ <domains>	N	Value of "ZUSERID" on page 702 in reporting object, when -report 1. Note: For use with the Configuration Server method ZPUTPROF.
-reset	0	N	Reset state and unlock Configuration Server on error 1=reset state and unlock CS's on error; 0=leave locked on error
-lock-to	FAIL	N	Action on pending hard-lock timeout: FAIL, RETRY, or FORCE Note: FORCE will kill all non-DCS tasks on slave Configuration Server.
-download	1	N	A (1 0) toggle that enables you to review the database differences before proceeding with the synchronization. For a more detailed description, see the section, "Download" below .
-max-deletes	-1	N	A "failsafe" option that can be set to prevent the inadvertent deletions of large parts of the database. For more information, see the section, "Maximum Deletions" on the next page .
-unchanged-domains	1	Y	By default, the time stamps are not checked before synchronization. To improve the performance by checking the time stamps, set the value to 0 in the destination.
-resource-prefix	" " (NULL)	N	Set this option to <code>/RESOURCE</code> to get data directly from the Apache server on the Core and if the <code>-dataless</code> option is set to 0. This may also improve the performance of downloading data (resources).

Download

- If 1, DCS will download the resources and synchronize the target database.
- If 0, processing will halt after the Differencing step. The resources are not downloaded and no metadata is committed. If any differences were detected, both databases will remain locked. Before deciding whether to proceed with the synchronization, check the differences in `zdreport.txt`—to see if they are what was expected.
 - Any subsequent execution of `dmabatch` without this option, enables the synchronization to resume using the differences that were already created.
 - To stop this synchronization and differencing, run `dmabatch action=reset`.

Maximum Deletions

An accidental deletion on the master database could be propagated to all the slave databases. This option limits the maximum number of items that can be deleted in one synchronization at the slave database. If the limit is exceeded in the Differencing step, the process is terminated without changing anything. It's up to the user to decide the *reasonable* maximum amount of deletions.

- A value of **0** is valid and will prevent all deletions.
- The default of **-1** means no deletion limit.

Configuration Object Equivalents

Some of the options in `dmabatch.rc` have operational equivalents in the configuration objects `ZMANAGER` and `ZMGRSYNC`. The following table lists these equivalents.

Configuration Object Equivalents of `dmabatch.rc` Options

Option	Set by Install	Configuration Object Equivalent
<code>-master-host</code>	Y	<code>ZMANAGER.ZTCPADDR</code>
<code>-master-id</code>	Y	<code>ZMGRSYNC.ZSRMGID</code>
<code>-master-port</code>	Y	<code>ZMANAGER.ZTCPPOINT</code>
<code>-master-timeout</code>	Y	<code>ZMANAGER.ZTIMEO</code>
<code>-slave-host</code>	Y	<code>ZMANAGER.ZTCPADDR</code>
<code>-slave-id</code>	Y	<code>ZMGRSYNC.ZDSTMID</code>
<code>-slave-port</code>	Y	<code>ZMANAGER.ZTCPPOINT</code>
<code>-slave-timeout</code>	Y	<code>ZMANAGER.ZTIMEO</code>
<code>-domains</code>	Y	<code>ZMGRSYNC.ZDOMAINS</code>
<code>-userid</code>	Y	<code>ZMGRSYNC.BATUSER</code>

Option	Set by Install	Configuration Object Equivalent
-password	Y	ZMGRSYNC.BATPWD
-dataless	Y	ZMGRSYNC.NOES
-report	N	ZMGRSYNC.REPORT
-dmastats-userid	N	ZMGRSYNC.REPTNAME
-reset	N	ZMGRSYNC.BATRESET
-lock-to	N	ZMGRSYNC.BATLOKTO

DMASTATS

The following table defines the fields of the DMASTATS object.

DMASTATS Fields Defined

Field	Definition
BATCHDAT	Date of this report
BATCHTIM	Time of this report
BATSTDAT	Date of correlated starting (id=1) report
BATSTTIM	Time of correlated starting (id=1) report
BATCHRC	Character return code (if REPORTID > 1) Note: See table " REPORTID Values Defined " for REPORTID values. See table " BATCHRC and BATCHMSG values " for detailed BATCHRC information.
BATCHMSG	Completion message (if REPORTID > 1) Note: See " REPORTID Values Defined " for REPORTID values.
BATARGS	DMABATCH command line
BATLKSTA	Result of DMABATCH ACTION=LOCKSTATUS: U (Unlocked); S (Soft-locked); X (Exclusive Soft-locked); H (Hard-locked)
DMASTATE	0 = Initial; 1 = Compared; 2 = Downloaded; 3 = Committed
ZSRCMGID	-master-id (Source MGR_ID)
ZDSTMGID	-slave-id (Destination MGR_ID)
REPORTID	Identifies which DCS processing point sent the report.

Field	Definition
	Note: See table "REPORTID values defined" for detailed REPORTID information.
SCOPE	N/A
ZDOMAINS	List of domains
ZUSERID	User name for use with PUTPROF method (see "ZUSERID" on the next page)

The following table identifies which DMASTATS.REPORTID processing point sent the report.

REPORTID Values Defined

REPORTID	Definition
1	Starting
2	Differencing completed, differences found
3	Differencing completed, no differences found
4	Staging completed
5	Commit completed
6	Ending

Every DMABATCH execution sends REPORTIDs 1 and 6. In addition, synchronizations might send REPORTIDs 3 or 2, 4, and 5 for intermediate status.

The following table lists the Configuration Server BATCHRC and corresponding BATCHMSG responses to DCS requests.

BATCHRC and BATCHMSG Values

BATCHRC	BATCHMSG
000	No differences found. / Successfully Completed
001	dmabatch internal structural error
016	Execution failed (see BATCHMSG for details).
101	Invalid cmdline keyword or keyword combination; master and slave IDs same.
103	Configuration Server ID/host/port not specified (see BATCHMSG for details).
108	No eligible domains.

ZUSERID

If `-dmatstats-userid` was specified in the configuration file (`dmabatch.rc`), ZUSERID uses that value. This name can be:

- A 32-character (maximum) alphanumeric name.
If it is longer than 32 characters, it will be truncated.
- US national characters, such as `@`, `$`, `#`, and `_` are allowed.

If `-dmatstats-userid` was not specified in the configuration file (`dmabatch.rc`), ZUSERID is created based on one of the following.

- If a synchronization operation:

```
DMA_src-id_dest-id_DOMS_domains
```

where *domains* is an underscore-separated list of Domains in this synchronization, or `$ALL$` if `ZDOMAINS=*`. For example,

```
DMA_100_203_DOMS_SOFTWARE_POLICY
```

- If a special stand-alone operation (such as `ACTION=LOCK`):

```
DMA_<target_id>_<action>
```

```
DMABATCH ACTION=LOCKSTATUS MGRID=123
```

creates

```
DMA_123_LOCKSTATUS
```

Note: *target_id* can be independent of *src_id* and *dst_id*.

DMABATCH Command-line Options

Reset

Normally, if synchronization fails during the Staging phase (for example, because of a lost connection), it is left in a state that ensures that it can subsequently be restarted from the point of failure. This entails leaving both Configuration Servers locked. If leaving both Configuration Servers locked, pending a restart, is not acceptable, this option enables the session to be reset to the initial state.

- To manually reset a failed session, specify:
DMABATCH ACTION=RESET

This action causes an immediate unlocking of both Configuration Servers; staged resources will immediately be freed. The trade-off is that "ability to restart" is sacrificed, which can be a problem if staging failed near the end of a long process.

Deferred Commit

DCS offers the ability to defer committing the database updates on the Destination Configuration Server to a time when it is less busy. To do this, use the DMABATCH command, COMMIT, as shown below.

- To defer the “commit” to a time when the Destination Configuration Server is less busy, specify:
`DMABATCH COMMIT=NO`

All DCS processing will be halted after the Staging phase.

Note: This command is equivalent to `-commit 0` in `dmabatch.rc`.

If Staging is successful, `BATCHRC=000` and the following message will be returned,

```
ZMGRSYNC.BATCHMSG="Commit bypassed by COMMIT=NO"
```

At this point the:

- Source Configuration Server will be unlocked,
- Destination Configuration Server will be soft-locked.

To commit the database updates to the Destination Configuration Server, the Commit phase must subsequently be done without `COMMIT=NO`.

DCS Objects

Obtain the following objects from the default Distributed Configuration Server directory. If DCS is running on a desktop with either an Sentient agent or Sentient Administrator, look in `IDMLIB`.

- ZMANAGER contains the properties of all the Configuration Servers that have been defined to DCS. TP parameters (including TP trace level) are defined here, per Configuration Server.
- ZMGRSYNC contains information about the synchronization pair, including any applicable password information. This object is refreshed when: 1) another synchronization pair is defined and 2) when there is a domain change. The object retains information of saved DCS sessions for subsequent recall.

The non-TP trace level is determined by `ZMGRSYNC.ZTRACEL`.

Two ZMGRSYNC variables are related to DMABATCH—`BATCHMSG` and `BATCHRC`. These are described in the ["DMASTATS" on page 701](#).

Note: Settings in the `dmabatch.rc` file will always supersede settings in the ZMANAGER and ZMGRSYNC objects.

DCS Files

This section defines the DCS `.MK`, `.DAT`, and `.IDX` files. Each of these files is preceded by a domain name, as in:

```
domain.dat
```

.MK

These are *metakit* database files that contain, in a compact and searchable format, a domain's metadata; its instances and class definitions.

- .MK files are built on all platforms, on the Source and Destination Configuration Servers.

.DAT

These files cache a domain's *small resource* files. This is a performance feature that minimizes CSDB file operations.

- .DAT files are built on Solaris and Windows platforms only; on Source Configuration Servers only.

Note: The amounts of free disk space and the space used by metakits and resource caches, before and after each domain analysis operation, are shown in `dmabatch.log`.

.IDX

These are *index* files for the corresponding .DAT files.

- .IDX files are built on Solaris and Windows platforms only; on Source Configuration Servers only.

DMABATCH Scripting Commands

The functions that are described in this section are DMABATCH command-line arguments, intended for use in a script that executes DMABATCH and handles error conditions. These functions are called with the command:

```
DMABATCH ACTION=
```

Note: If no value is specified for **ACTION** (as seen above), or if **ACTION=SYNC**, a normal synchronization is done. Any other **ACTION** value does the indicated action only, with no synchronization.

DMABATCH Line Commands

This section details the use and functionality of the DMABATCH commands.

DMABATCH ACTION=QUIESCE [CSID=id][QTYPE=TASK|TRANS]

This command puts the Destination in a "quiescent" state, thereby increasing the chance of later getting a hard-lock.

- If **CSID=** is omitted, the default is the Destination's ID.
- **QTYPE=TASK**: prevents any new, non-DCS Sentient agent tasks from starting.

- **QTYPE=TRANS:** same as **TASK**, but also, for currently running non-DCS Sentient agent tasks, the Sentient agent connection terminates when the Sentient agent sends the next transaction.
- This action would likely be scripted to run before a synchronization, thereby decreasing the likelihood of later having to stop any tasks.

DMABATCH ACTION=RESUME [CSID=id]

This command ends a "quiescent" state on the Destination.

- If **CSID=** is omitted, the default is the Destination's ID.

DMABATCH ACTION=KILLTASKS [CSID=id]

This command should be used if **QUIESCE** was not sufficient to clear out other tasks in time for DCS to enter the Commit phase. It terminates all non-DCS Sentient agent tasks on the Destination, enabling a DCS run to obtain a hard-lock and commit the changes.

- If **CSID=** is omitted, the default is the Destination's ID.
- Use **KILLTASKS** in a script after a synchronization terminates with **BATCHRC=003** (hard-lock timeout).

DMABATCH ACTION=RESET

This command will reset an incomplete synchronization session to its initial state, cause an immediate unlock of both Configuration Servers, and release staged resources (if any). Any subsequent synchronization of the defined synchronization pair will start from scratch.

- Use **RESET** in a script after a synchronization fails (**BATCHRC** not = 000) if it is determined that the synchronization cannot be resumed in a timely manner and the Configuration Servers cannot be left locked.

**DMABATCH ACTION=SOFTLOCK [CSID=id], DMABATCH ACTION=UNLOCK [CSID=id],
DMABATCH ACTION=HARDLOCK [CSID=id]**

These commands are intended for use with multiple synchronizations from a shared Source database.

- Soft-locking the Source guarantees that resource data that is created for Source Domains will be retained in cache instead of being recalculated for each synchronization, thereby increasing performance.
- If **CSID=** is omitted, the default is the Destination's ID.

If the **HARDLOCK** command is run, the specified Configuration Server will be hard-locked, thereby preventing Sentient agents from connecting. Also, any Sentient agent tasks that are running on the Configuration Server are killed.

The specified Configuration Server must currently be unlocked. The default is the Destination Configuration Server.

Note: This operation is intended for use in environments in which multiple Configuration Servers share a database—a practice that is *not advised*.

CSID Value Considerations

- If `CSID=` is omitted, the default is the Destination Configuration Server's ID.
- If `CSID` $\neq$ the Source or Destination Configuration Server (master or slave), the `-port` and `-address` of the Configuration Server must be specified.

Results of DMABATCH

The results of a DMABATCH synchronization are found in the *batch-message* variable, `BATCHMSG`, and the associated *batch return-code* variable, `BATCHRC` (see table "[BATCHRC and BATCHMSG values](#)"), in the `ZMGRSYNC` object.

- After the Differencing step, a `BATCHRC` of `000` indicates that no domain differences were found.
The associated `BATCHMSG` message is "No differences found."
- During the Staging and Commit phases of DCS, the `000` return code indicates that the phase was successful.
The corresponding `BATCHMSG` is "Successfully Completed."

Configuration Server Messages

Introduction

The Configuration Server generates a message for each task that it performs. These messages are written to the Configuration Server activity log, or another location of your choice, and provide information about the status of the Configuration Server's activities.

These activities include startup, the logging on of clients, and the launching of **Sentient Enterprise Configuration Server Database Editor (CSDB Editor)** sessions. As with the client logs, the tracing levels of the Configuration Server's diagnostic logging can be customized. The volume and detail level of the Configuration Server messages depend on two factors: the **trace settings** and **logging options** specified for that Configuration Server.

This section helps you understand the following points.

- *Why do I have a Configuration Server activity log?*
The messages that are written to the activity log can assist in troubleshooting when problems occur. Additionally, the Persistent Systems Technical Support group usually requests these logs when called upon for assistance in problem resolution.
- *What do the messages mean?*
The messages that the Configuration Server generates provide an insight into what it is doing, and when it is being done. The Configuration Server reports whether everything is normal and, if not, what and where the possible cause of the problem is.
- *What can I do with the information?*
Once you can read and understand the Configuration Server messages, you can take the necessary measures to correct the problem and re-establish the integrity of your computing environment.
- *How can I control/configure the Configuration Server log information?*
In the `MGR_MESSAGE_CONTROL` section of the `edmprof` file, you have the ability to specify where you would like log messages sent (logged); in the `MGR_LOG` section, you can

determine various aspects of log behavior; and in the MGR_TRACE section, you can control diagnostic aspects of logging.

Configuring Configuration Server Log Location and Settings

An administrator determines where the Configuration Server messages log will be written. For information on specifying a destination for the log messages, see ["MGR_MESSAGE_CONTROL Section" on page 711](#).

During the installation process, allocate space for the Configuration Server log. After the installation, specify the logging options for the log file in the MGR_LOG section of the `edmprof` file. See ["MGR_LOG Section" on the next page](#).

A third section that will be beneficial is MGR_TRACE, which controls the diagnostic logging for the Configuration Server. The section ["MGR_TRACE Section" on page 712](#) offers an overview of the trace options for logging.

Note: The MGR_XXXX sections mentioned above, as well as other sections of the `edmprof` file that impact the Configuration Server log are covered in greater detail in the ["Tuning the Configuration Server" on page 498](#). For example, in the MGR_STARTUP section, the SHOW_VERINFO setting allows you to display version information for each Configuration Server module at the beginning of the log.

The `edmprof` file can be edited in a standard text-editing application.

Caution: Be sure to review the important information in this section before editing this file. Failure to do so could adversely affect your Sentient environment.

It is recommended to back up the `edmprof` file prior to editing it.

- You must use a UTF8-aware text editor when editing the `edmprof` file.
- Be sure to select UTF-8 as the *encoding type* when saving the file.
 - Windows users are advised to use **Notepad**.

Failure to use a recommended text-editing application and saving the file as recommended in this section could result in the file's changes not being correctly applied to the Configuration Server. This could adversely affect your Sentient environment.

MGR_LOG Section

The settings in the MGR_LOG section determine where the log is located, its appearance, and its flexibility.

MGR_LOG Settings

Setting	Description
DIRECTORY	The fully qualified directory to which the Configuration Server log is written.

Setting	Description
DISABLE_NT_EVENT_LOGGING	This parameter affects only the event logging of Configuration Server log messages. Some Windows Event log records are written without any corresponding Configuration Server log messages (Windows only).
DISABLE_SNMP_TRAP_LOGGING	This parameter affects only the trapping of Configuration Server log messages. Some SNMP traps are issued without any corresponding Configuration Server log messages.
FLUSH_SIZE	The total buffer size (in bytes) of all messages stored by the Configuration Server, before it writes them to the log.
MESSAGE_DATE	Allows insertion of the date into every line in the log. For more information on including the date in the log messages, see "Reading the Configuration Server Log" on page 713 .
MESSAGE_DELIMITER	The left and right delimiters that are used for enclosing the task name and task ID in the Configuration Server log messages. The format is MESSAGE_DELIMITER=xy, where x is the left delimiter and y is the right. The options are: [], (), < >, and { }. The default is [].
MESSAGE_PREFIX	The three-digit, alphabetic, Configuration Server identifier that precedes Configuration Server log message numbers. Specify any three-digit, alphabetic value. The default is NVD .
MESSAGE_WIDTH	The length (in number of characters) of each message line.
PIPESIZE	The number of messages (in bytes) that the Configuration Server will collect before processing and storing them in the log buffers.
SWITCH_TOD	A time-of-day specification that will, each day, automatically trigger log switching.
THRESHOLD (or THRESHHOLD)	The number of lines that the log file will contain before it is dumped to another file. When the log is dumped, it will be given a temporary file name in the format of NVDMRxxx, and the Configuration Server creates a new log file with a different log ID. For more information, see "THRESHOLD" on page 524 .

Examples

Windows

```

[MGR_LOG]
DIRECTORY          = C:\Program Files\PSL\RCA\ConfigurationServer\log

FLUSH_SIZE         = 1000

MESSAGE_DATE       = JULIAN

MESSAGE_PREFIX     = NVD

MESSAGE_WIDTH      = 256

PIPE_SIZE          = 1000000

THRESHHOLD         = -5000000

```

MGR_LOG Settings Values

Settings	Value as Installed	Default	Minimum
DIRECTORY	N/A	Program Files\PSL\RCA\Configuration Server\log	N/A
DISABLE_NT_EVENT_LOGGING	N/A	NO	N/A
DISABLE_SNMP TRAP_LOGGING	N/A	NO	N/A
FLUSH_SIZE	1000 bytes	1000 bytes	1
MESSAGE_DATE	N/A	N/A	N/A
MESSAGE_DELIMITER	N/A	N/A	N/A
MESSAGE_PREFIX	NVD	NVD	N/A
MESSAGE_WIDTH	256	90	80
PIPESIZE	1000000 bytes	6553516 KB	1
SWITCH_TOD	N/A	N/A	N/A
THRESHOLD	-5000000 bytes	1000000 bytes	1

Performance and Usage Considerations

- Increasing the buffer size (FLUSH_SIZE) will enhance performance, but will delay the flushing of messages to the log file.
- If log messages are truncated, increase the MESSAGE_WIDTH value.

- If you use the Configuration Server log to closely monitor system status, set THRESHOLD to a positive value to create and save successive portions of the Configuration Server log. If disk storage is critical, set the value to a negative number to reuse the allocated log disk space.
- If you are invoking a number of Configuration Server methods, you can control the size of the log for each method using the LOG_LIMIT setting of the MGR_METHODS section. Additionally, TASK_LOG_LIM (in the MGR_TASK_LIMIT section) controls the number of messages printed by the execution of each task.
- When modifying parameters in this section as they relate to memory or disk utilization, take care to be certain that you do not exceed the maximum amount of memory or storage space available.

After monitoring your system's performance, you can tune the MGR_LOG settings for maximum efficiency. Change the settings to maximize memory usage and for other performance considerations. For more information on the MGR_LOG settings, see ["MGR_LOG" on page 521](#).

The following section describes your message destination choices.

MGR_MESSAGE_CONTROL Section

The MGR_MESSAGE_CONTROL section of the `edmprof` file is where you specify the selection and destination of log messages. This section provides an overview of the:

- Various destinations that are available for log messages,
- Configuration of, and settings for, the MGR_MESSAGE_CONTROL section.

This section of the `edmprof` file specifies where log messages are to be sent and whether they are to be suppressed. MGR_MESSAGE_CONTROL allows you to specify an individual message, a series of messages, or a range of messages, and a set of message destinations.

The format of lines in the MGR_MESSAGE_CONTROL section is:

(Message Number(s)) = (Message Destination(s)).

The left side of the equals sign (=) specifies the message numbers that are affected by the command. There is also a special directive; ALL, which can be used on the left side of the equals sign to affect all messages (0001-9999). If no destination is specified after the equals sign, the message has no associated destination and is suppressed.

The table ["MGR_MESSAGE_CONTROL Settings"](#) shows the destinations to which the log messages can be written.

MGR_MESSAGE_CONTROL Settings

Setting	Destination
EVENTLOG	Specify to write log messages to the Windows Event log (Windows only).
LOG	Specify to write log messages to the Configuration Server log.
REXX	Specify to have the messages sent to a pre-determined REXX (named MGRLOG). Important: MGRLOG is not an existing REXX, and must be created in order to

Setting	Destination
	use this facility. Note: The MGRLOG REXX will receive the message number and the message text as its first and second input parameters. You can then process the information in whatever manner that you choose.
SNMPTRAP	Specify this to write log messages as traps to the current SNMP Manager.
USERLOG	Specify this to have the messages sent to the user log. Note: You must first activate the user log feature in the MGR_USERLOG section of the <code>edmp_{prof}</code> file by specifying <code>ACTIVATE=YES</code> .

For detailed performance and usage information, see the MGR_MESSAGE_CONTROL section in the "MGR_MESSAGE_CONTROL" on page 528.

MGR_TRACE Section

The MGR_TRACE section of the `edmpprof` file controls the diagnostic logging for the Configuration Server. All diagnostic output produced by MGR_TRACE keywords is written to the active Configuration Server log.

The trace settings specified in the MGR_TRACE section are invoked at Configuration Server initialization and are displayed at the beginning of the Configuration Server log. They remain in effect until changed.

A Sample Configuration Server Activity Log

A properly sized and configured log will give you appropriate detail on the entire scope of Configuration Server operations. While individual events vary from one installation to another, the general flow of the Configuration Server log consists of five phases:

- Configuration Server startup
- Sentient agent connect
- Object resolution
- End of Sentient agent connect
- Configuration Server shutdown

The next sections describe the types of messages that result from each phase.

Startup

A series of messages at the beginning of the log reflect Configuration Server startup, starting with the opening of the log. Next, the Configuration Server processes the parameters in the `edmpprof` file and starts various Configuration Server tasks. At this point, the Configuration Server is fully

configured and operational, waiting for agents to connect in order to continue processing. Messages associated with this phase are: Configuration Server tasks starting, Configuration Server tasks attaching, and Configuration Server Settings section names and values.

Agent Connect

The Sentient agent connect occurs when an agent requests a session with the Configuration Server in order to be resolved to its desired state. Each agent connect is a Configuration Server task. The Configuration Server might defer the session to a time when it is not occupied, or will connect to the agent at the time of the request. Messages associated with agent connect can be:

- “Client Automation Agent task has started.”

and...

- “Connected to Client Automation agent.”

Object Resolution

Object resolution follows a successful Sentient agent connect. The Configuration Server processes agent objects; performs methods resolved from those objects and creates resulting objects; transfers data to and from the agent; and creates objects that contain audit and profile information for that agent. When the agent has been returned to its desired state, the Configuration Server releases it and waits for another connect request. Messages associated with the object-resolution phase include:

- “Object resolution *object_name* begins/ends.”
- “Current path (pointing to the methods directory or resource file).”
- Warnings that objects could not be found or duplicate objects exist, method identification.
- Return codes for methods.

End of Agent Connect

The end of the agent connect occurs when the agent requests that its session with the Configuration Server be terminated as a normal logoff. The message associated with end of agent connect is

- “Client Automation Agent task has ended.”

Shutdown

Configuration Server shutdown occurs when a request to shut down the Configuration Server has been received. Messages associated with this phase include:

- “Shutting down Client Automation Configuration Server.”
- Messages halting communications.
- Error messages that result in the Configuration Server ending operations.

Viewing the Configuration Server Log

The Configuration Server log file reports program activities. Use a text editor to view the Configuration Server log. The Configuration Server log is located in the directory that was specified during installation.

Note: The Windows operating systems have GUI interfaces (program groups and icons) for viewing the Configuration Server log.

Reading the Configuration Server Log

The logging facility generates an activity log with multiple messages. In order to read and understand the information that is presented in the log, it is helpful to be familiar with its format.

Each Configuration Server log message consists of six fields:

• "Message Number" below	• "Time Stamp" on page 717
• "Task Name" on page 718	• "Task Type" on page 719
• "Message Type" on page 719	• "Message" on page 719

A typical log entry is displayed in the following sample from a Configuration Server log on a Windows server machine.

Example

```
NVD8349I 13:48:22 [NFYTMGR1/144] System Task  
--- TCP Notify <NFYTMGR1> started
```

The sections that follow provide more information about each of these fields.

Message Number

The message number is an alphanumeric, eight-character string that begins with a three-letter prefix, followed by the four-digit, unique message number, and ends with a letter that indicates the severity of the message.

Three-letter Prefix

The default three-letter prefix; NVD, is customizable through the MESSAGE_PREFIX setting in the MGR_LOG section of the `edmpprof` file. For example, if your business name is Acme Data Solutions, you can customize your log prefix to **ADS**. For more information, see the ["MESSAGE_PREFIX" on page 522](#).

Four-digit Message Number

The message number is one of several thousand numbers that has been logically grouped and individually assigned to a Configuration Server operations activity.

Severity Indicator

The alphabetic indicators are:

- **E**, indicating Error messages,

Note: Error messages are easily distinguishable because they are presented in all UPPERCASE letters.

- **W**, indicating Warning messages, and
- **I**, indicating Information messages.

Note: In addition to the standard Error, Warning, and Information message types, there are a few instances of *Diagnostic* and *Always* messages. These are denoted by the letters **D** and **A**, respectively, and like the other indicators (I, E, and W), will be found in the last position of the message number. The severity of these messages is comparable to I and W messages, and they are intended for internal diagnostics.

Time Stamp

The time stamp (HH:MM:SS) indicates the time at which the event occurred.

Adding a Date Field

You have the option of adding the date to this field. This enables an administrator to scan the Configuration Server log and quickly distinguish each day's event logging.

The date can be expressed in one of three formats: **Julian**, **Gregorian**, or **International Standards Organization (ISO)**. The following examples are of February 04, 2005 in each format:

- ISO (YYYYMMDD): **20050204**
- JULIAN (YYYYDDD): **2005035**
- GREGORIAN (DDMMYYYY): **04022005**

In the MGR_LOG section of the `edmpconf` file, add the setting `MESSAGE_DATE`, and define its value as (uppercase) `JULIAN`, `GREGORIAN`, or `ISO`.

The following example shows a Configuration Server log message that might result if `JULIAN` were specified for `MESSAGE_DATE`. Note that the time stamp (13:48:22) is now preceded by the Julian date (2005035).

```
NVD8349I 2005035 13:48:22 [NFYTMGR1/144] System Task  
--- TCP Notify <NFYTMGR1> started
```

Task Name

This is a two-part field that indicates: the task for which the activity is being logged or the communications address (IP address) that is being used, followed by the task ID.

```
[taskname (xxxxxxx) or commaddr (nnn.nnn.nnn.nnn) / taskid]
```

- The task name can be either:
 - A Configuration Server task, such as `ZUTILMGR`, or
 - The communications address (IP address) of the Configuration Server, such as `192.168.22.12`.

Task Name Categories

The value of the *taskname* field is dependent on the task that has initiated the message. The table "Task Name Categories" lists the various categories of tasks; a set of specific functions is associated with each category.

Task Name Categories

• Agent Manager	• Client
• Clock Manager	• Log Manager
• REXX Manager	• Task Manager
• TCP/IP Manager Emulator	• TCP/IP Notify Manager

When a message is written to the log, the *taskname* field will be populated with specific task information (as described in the preceding bullets), not the category. Task information includes, but is not limited to, internally assigned names, user assigned names, network host names, and IP addresses.

- The task ID represents the operating system process number.

Note: On some operating systems, taskid is referred to as **processid** (PID).

Task Type

The task type indicates the type of task (such as System and Admin tasks) that initiated the message.

Message Type

Like the alphabetic indicator in the last position of the message number, the message type indicates the severity of the message. The message type indicator is a three-character field that enables an administrator to scan a Configuration Server log and quickly identify warning and error messages. The indicators are:

- --! for Error messages,
- --? for Warning messages, and
- --- for Information messages.

Message

The message field contains the message text.

Note: Typically, Configuration Server logs contain multiple messages for a single activity. Most activities that require multiple messages will be framed with *activity begins* and *activity*

ends messages.

Configuration Server Messages

This section is divided into the following two parts:

- "[Numeric Ranges](#)" [below](#), details the numeric classifications of the various Configuration Server log messages, and
- "[Configuration Server Messages](#)" [on page 724](#), presents a sequential list of all the messages that are generated by the Configuration Server logging facility.

Numeric Ranges

The following table presents the numeric ranges of the Configuration Server messages. Each major area and corresponding numeric range is shaded and in bold text followed by a sub-topic and its range.

Configuration Server Messages Numeric Ranges

Message Classification	Numeric Range
Configuration Server Tasks	0001 – 0999
ZTOPTASK	0001 – 0099
ZTASKMGR	0100 – 0199
ZLOGMGR	0200 – 0299
ZCLKMGR	0300 – 0399
Communication	0400 – 0499
ZUTILMGR and ZSNMPUTL	0500 – 0599
ZSNMPMGR	0600 – 0699
ZTCBG and ZCVT	0700 – 0749
ZMCSTMGR	0750 – 0799
ZCONMGR and ZLINECMD	0800 – 0899
ZSMTRMGR and ZSMTSMGR	0900 – 0999
Memory	1000 – 1099
Allocation, Expansion, etc.	1000 – 1099
Heap	1100 – 1199
<i>Currently unassigned.</i>	1200 – 1299

Message Classification	Numeric Range
<i>Currently unassigned.</i>	1300 – 1399
<i>Currently unassigned.</i>	1400 – 1499
Cache	1500 – 1599
Cache	1600 – 1699
Semaphores and IPC	1700 – 1799
<i>Currently unassigned.</i>	1800 – 1899
ZPOOLS	1900 – 1999
Communications Functions	2000 – 2999
<i>Currently unassigned.</i>	2000 – 2099
Sentient Agent Connection	2100 – 2199
Pipe	2200 – 2207
Compress	2208 – 2249
FTP	2250 – 2299
Miscellaneous	2300 – 2399
<i>Currently unassigned.</i>	2400 – 2499
<i>Currently unassigned.</i>	2500 – 2599
File Input/Output	2600 – 2699
<i>Currently unassigned.</i>	2700 – 2799
<i>Currently unassigned.</i>	2800 – 2899
<i>Currently unassigned.</i>	2900 – 2999
Resolution	3000 – 3999
Object Resolution	3000 – 3099
User ID Errors	3100 – 3199
Object Resolution	3200 – 3299
Tree Resolution	3300 – 3399
<i>Currently unassigned.</i>	3400 – 3499
Metaclass Resolution	3500 – 3599
Class Resolution	3600 – 3699

Message Classification	Numeric Range
<i>Currently unassigned.</i>	3700 – 3799
<i>Currently unassigned.</i>	3800 – 3899
<i>Currently unassigned.</i>	3900 – 3999
Verification	4000 – 4999
Security	4000 – 4099
License and Profile	4100 – 4199
Database & Y2K	4200 – 4299
Database	4300 – 4399
<i>Currently unassigned.</i>	4400 – 4499
<i>Currently unassigned.</i>	4500 – 4599
<i>Currently unassigned.</i>	4600 – 4699
<i>Currently unassigned.</i>	4700 – 4799
<i>Currently unassigned.</i>	4800 – 4899
<i>Currently unassigned.</i>	4900 – 4999
Variables & Objects	5000 – 5999
Variable Substitution	5000 – 5099
Instances	5100 – 5199
Objects and Variables	5200 – 5299
Object Processing	5300 – 5399
Resource	5400 – 5499
<i>Currently unassigned.</i>	5500 – 5599
<i>Currently unassigned.</i>	5600 – 5699
<i>Currently unassigned.</i>	5700 – 5799
PROC_API	5800 – 5899
EDMMAPID	5900 – 5999
ZADMIN	6000 – 6999
ZATTACH	6000 – 6099
Methods	6100 – 6199

Message Classification	Numeric Range
Methods	6200 – 6299
Methods	6300 – 6399
Methods	6400 – 6499
Methods	6500 – 6599
Methods	6600 – 6699
Methods	6700 – 6799
<i>Currently unassigned.</i>	6800 – 6899
Package Consolidation	6900 – 6999
Database	7000 – 7999
Find	7001 – 7080
Invalid Name	7081 – 7099
Database Prefix Area	7100 – 7199
Control	7200 – 7299
DMA	7300 – 7399
Object ID	7400 – 7499
Object ID	7500 – 7599
<i>Currently unassigned.</i>	7600 – 7699
<i>Currently unassigned.</i>	7700 – 7799
<i>Currently unassigned.</i>	7800 – 7899
ZVSAM	7900 – 7999
Methods	8000 – 8999
Errors	8000 – 8099
Informational	8100 – 8199
Expressions	8200 – 8299
ZNFY	8300 – 8399
ZMETHOD	8400 – 8449
ZRTRYMGR	8450 – 8499
<i>Currently unassigned.</i>	8500 – 8599

Message Classification	Numeric Range
<i>Currently unassigned.</i>	8600 – 8699
<i>Currently unassigned.</i>	8700 – 8799
<i>Currently unassigned.</i>	8800 – 8899
<i>Currently unassigned.</i>	8900 – 8999
General	9000 – 9999
Multicast	9000 – 9099
General	9100 – 9149
ZSYSINFO	9150 – 9299
ZTASKLIM	9300 – 9399
I18N / UTF-8	9400 – 9499
I18N / UTF-8	9500 – 9599
<i>Currently unassigned.</i>	9600 – 9699
<i>Currently unassigned.</i>	9700 – 9799
<i>Currently unassigned.</i>	9800 – 9899
<i>Currently unassigned.</i>	9900 – 9999

Configuration Server Messages

The following table contains a sequential list of all the messages that are generated by the Configuration Server logging facility.

Note: In the following table, symbolic substitution is used as follows:

<%s> = a character string, such as a file name or an object;

<%4d> = a decimal number, such as an IP Port.

In the Configuration Server log, however, these symbols will be replaced by the data that are applicable to the message.

Configuration Server Messages

Message Number	Log Message
0000-0099	ZTOPTASK
NVD0009I	"%s"
NVD0010I	"%s consists of the following <%d> modules:"

Message Number	Log Message
NVD0011E	"CANNOT LOAD MODULE %s, ZLOADM FAILED, RC = <%08X>"
NVD0012E	"MODULE <%s> COULD NOT BE LOADED"
NVD0013E	"MODULE <%s> DOES NOT CONTAIN VERSION INFORMATION"
NVD0014E	"CANNOT OPEN RESOURCE FILE FOR %s <%d>"
NVD0015E	"CANNOT OPEN OUTPUT FILE <%s>"
NVD0016E	"CANNOT OPEN <%s>"
NVD0017E	"CANNOT UNLOAD MODULE <%s>, ZUNLOADM FAILED, RC = <%d>"
NVD0018E	"NO MODULES FOUND"
NVD0019E	"UNIDENTIFIED MODULES INCLUDED = <%d>"
NVD0051I	"Verifying License"
NVD0052I	"Verifying Database"
NVD0053I	"Initializing Cache"
NVD0054I	"MGR_ID must be specified in EDMPROF and be made up of three hex digits (A-F and 0-9)."
NVD0055I	"Current Client Automation Configuration Server directory is <%s>"
NVD0056E	"CLIENT AUTOMATION CONFIGURATION SERVER <%s> - LICENSE VERIFICATION FAILED"
NVD0057E	"CLIENT AUTOMATION CONFIGURATION SERVER WILL BE SHUTDOWN"
NVD0058E	"DATABASE OPEN FAILED, CLIENT AUTOMATION CONFIGURATION SERVER CANNOT BE STARTED"
NVD0059E	"MGR_ERROR_CONTROL DBError = SHUTDOWN, CLIENT AUTOMATION CONFIGURATION SERVER WILL BE SHUTDOWN"
NVD0060W	"MGR_ERROR_CONTROL DBError = IGNORE, Client Automation Configuration Server initialization continues..."
NVD0061E	"DATABASE VERIFICATION FAILED, RC <%ld>"
NVD0064I	"Database Verification complete, rc <%ld>"
NVD0065I	"Attaching Clock Manager Task"
NVD0066I	"Client Automation Configuration Server starting, MGR_ID = <%s>"
NVD0067I	"Attaching Task Manager Task"

Message Number	Log Message
NVD0068I	"Client Automation Configuration Server <%s> cache is not available"
NVD0069E	"CLIENT AUTOMATION CONFIGURATION SERVER <%s> ERROR REATTACHING CLOCK MANAGER"
NVD0070I	"Client Automation Configuration Server <%s> re-attached Clock Manager"
NVD0071E	"CLIENT AUTOMATION CONFIGURATION SERVER <%s> ERROR REATTACHING TASK MANAGER"
NVD0072I	"Client Automation Configuration Server <%s> re-attached Task Manager"
NVD0073I	"Child with PID <%d> died...Cleaning up"
NVD0074I	"Shutting down Client Automation Configuration Server <%s>"
NVD0075I	"Requesting ZTASKMGR (ID=<%d>) to shutdown"
NVD0076I	"Terminating ZTASKMGR (ID=<%d>)"
NVD0077I	"Terminating Task <%s> ID=<%d>"
NVD0078I	"\n"
NVD0079W	"Client Automation Configuration Server program <%-8.8s> preload failed"
NVD0080W	"Preload failed for <%-8.8s>"
NVD0081E	"%s"
NVD0082W	"%s"
NVD0083E	"LICENSE VERIFICATION PROBLEM, PLEASE CALL TECH-SUPPORT! **"
NVD0084I	"Client Automation Configuration Server is not running APF authorized. RC = <%d>"
NVD0085I	"Client Automation Configuration Server could not be set non-swappable. RC = <%d>"
NVD0086I	"Client Automation Configuration Server is running APF authorized and non-swappable"
NVD0088I	"%-8s %-8s: %6.6d"
NVD0089I	"License usage report for Client Automation Configuration Server %s"
NVD0091I	"%s"
NVD0092W	"No license report file <%s> found"
NVD0093I	"%s"
NVD0094W	"License report method launch failed"

Message Number	Log Message
0100-0199	ZTASKMGR
NVD0100I	"Attaching <%s>"
NVD0119E	"INSUFFICIENT NUMBER OF SLOTS IN ATTACH TABLE - NO RESTART FOR <%s> INDEX <%d>"
NVD0120E	"THE CLIENT AUTOMATION CONFIGURATION SERVER IS NOT LICENSED FOR SSL. <%s> WILL NOT START"
NVD0121E	"ERROR ATTACHING <%s> - ERROR LOADING DLL <%s>"
NVD0122E	"ERROR ATTACHING <%s> - DLL <%s> FUNCTION <%s> NOT FOUND"
NVD0123E	"ERROR ATTACHING <%s> - PROGRAM <%s> UNKNOWN TO ZTASKMGR"
NVD0124E	"UNABLE TO ATTACH <%s>"
NVD0130I	"Requesting Task <%s> ID=<%d> to shutdown"
NVD0131I	"Terminating <%s> TASKID <%d>"
NVD0132E	"UNABLE TO RE-START <%s>"
NVD0133E	"UNABLE TO OPEN PROFILE -- NO TASK WILL BE ATTACHED"
NVD0134E	"NO ATTACH LIST SPECIFIED OR LIST IS EMPTY. NO TASK ATTACHED"
NVD0135I	"<%s> PID <%i>, PPID <%i> updated to <%i>"
NVD0136I	"Will try to restart <%s>"
NVD0137E	"%s: UNABLE TO OPEN PROFILE -- NO SPECIFIED TASKS ATTACHED"
NVD0138E	"SSL MANAGER: MISSING SSL PARAMETERS - TERMINATING ZSSLMGR"
NVD0139E	"SSL MANAGER: CERTIFICATE FILE NOT SPECIFIED - TERMINATING ZSSLMGR"
NVD0140W	"SSL Manager: Private key file not specified, using certificate file"
NVD0141E	"SSL MANAGER: PRIVATE KEY PASSWORD CANNOT BE GREATER THAN <%d> BYTES"
NVD0148I	"Shutting down Task Manager <%s>"
NVD0149I	"Waiting up to %d seconds for termination of <%s> TASKID <%08X>"
0200-0299	ZLOGMGR
NVD0200I	"----- --- %s"

Message Number	Log Message
NVD0201I	"----- --- %s has occurred on date %s at %s"
NVD0202I	"----- --- running on a %1d.%2.2d.%3.3d%c Manager at service pack %s"
NVD0203I	"----- --- with a Manager name of %s"
NVD0204I	"----- --- and a Manager ID of %s"
NVD0205I	"----- --- %s"
NVD0206I	"----- --- on a %s system"
0400-0499	Communications
NVD0401I	"Transaction ended"
NVD0402I	"%s task has started"
NVD0403I	"TCP/IP Port# <%04d>"
NVD0404I	"TCP/IP accepting requests at address <%s> on port <%04d>"
NVD0405I	"Received request for new connection from <%s>"
NVD0406I	"Requesting all <%s> ZTERMTSK to shutdown"
NVD0407I	"Requesting Task PID (%d) to shutdown"
NVD0408I	"Terminating Method (%d) of Task PID (%d)"
NVD0409I	"Starting Task for <%s>, active tasks <%d>"
NVD0410E	"APIACCEPT ERROR <%s>"
NVD0411I	"LU62 Manager accepted a new Client Automation Agent (conv_id=<%08X%08X>)"
NVD0412I	"HTTP Port# <%04d>"
NVD0413I	"HTTP Manager accepting requests at address <%s> on port <%04d>"
NVD0414I	"%s Manager Task has started"
NVD0415W	"%s Rexx complete, rc <%d> <%s>"
NVD0416I	"TCP SSL Port# <%04d>"
NVD0417I	"TCP SSL Manager accepting requests at address <%s> on port <%04d>"
NVD0418E	"TERMINATING <%s> TASKID <%d>]"
NVD0419E	"ATBALLOC ERROR - %s"

Message Number	Log Message
NVD0420E	"ERROR, COULD NOT BIND SOCKET TO <%s> PORT %hd"
NVD0421E	"ERROR, CLOSING SOCKET, RC = <%d>"
NVD0422E	"ERROR, PORT NUMBER SPECIFIED <%s> IS INVALID."
NVD0423E	"CLOSE ACB ERROR FOR <%s> - <%s>"
NVD0424E	"ERROR OF DosSetExceptionHandler(): RETURN CODE = <%d>"
NVD0425E	"ERROR LOADING LU6.2 LIBRARY, RC = <%d>"
NVD0426I	"LU6.2 Manager TP Name is <%s>"
NVD0427I	"LU6.2 Manager LU Name is <%s>"
NVD0428E	"ATBALLOC ERROR - <%s>"
NVD0429E	"ZLU62MGR ENDING DUE TO ERRORS"
NVD0430E	"APIOPEN ERROR - <%s>"
NVD0431E	"APISTARTLOGON ERROR - <%s>"
NVD0432I	"ZLU62MGR Simulation discontinued due to short on storage condition"
NVD0433I	"ZLU62MGR Simulation discontinued due to task attach error"
NVD0434I	"ZLU62MGR ending"
NVD0437E	"APIGETLOCALHOST ERROR - <%s>"
NVD0439I	"Logons Disabled -- closing connection"
NVD0440I	"TASKGEN closing connection"
NVD0441I	"Accepting New Client Automation Agent <%s>"
NVD0442I	"Attached ZTERMTSK <%s>"
NVD0448I	"TASKGEN address in use - <%s>"
NVD0449I	"TASKGEN terminating previous connection <%d>"
NVD0450E	"TASKGEN ADDRESS STILL IN USE - <%s>"
NVD0460E	"ERROR LOADING LIBRARY zsys_loadDynamicLibrary() OF <%s>, RC = <%d>"
NVD0461E	"ERROR DURING zsys_getDynamicLibraryFunctionByName() OF ORDINAL <%ld> IN <%s>, RC = <%d>"
NVD0462I	"znetwork: Dumping Sent Contents"

Message Number	Log Message
NVD0463I	"znetwork: Dumping Received Contents"
NVD0464I	"Client Automation Agent task started"
NVD0465I	"Connected to Client Automation Agent"
NVD0466E	"ERROR IN CLINIT - <%s>"
NVD0467I	"Client Automation Agent task has ended"
NVD0468I	"%s Manager now running with LIMIT=CLOSE option selected"
NVD0469I	"Connection closed due to invalid IP address received: <%s>"
NVD0470I	"TCP Manager ending"
NVD0471E	"TCP MANAGER ENDING DUE TO ERRORS"
NVD0472I	"SSL Manager accepting requests at address <%s> on port <%04d>"
NVD0473I	"SSL Manager ending"
NVD0474E	"SSL MANAGER ENDING DUE TO ERRORS"
NVD0475I	"HTTP Manager ending"
NVD0476E	"HTTP MANAGER ENDING DUE TO ERRORS"
NVD0477I	"HTTPS Manager ending"
NVD0478E	"HTTPS MANAGER ENDING DUE TO ERRORS"
NVD0479I	"HTTPS Manager accepting requests at address <%s> on port <%04d>"
NVD0480E	"%s: HTTPS MANAGER TERMINATING DUE TO MISSING SSL PARAMETERS"
NVD0481E	"%s: HTTPS MANAGER TERMINATING DUE TO CERTIFICATE FIEL WAS NOT SPECIFIED"
NVD0482W	"%s: Private key file not specified, using certificate file"
NVD0483I	"%s Manager Task Terminating"
NVD0484I	"Date has been changed. New date is : %s"
NVD0485I	"%s *%. *s*"
0500-0599	ZUTILMGR and ZSNMPUTL
NVD0550E	"CLIENT AUTOMATION CONFIGURATION SERVER TASKID <%d> INVALID SNMP LOGGER SEMAPHORE ID"
NVD0551E	"CLIENT AUTOMATION CONFIGURATION SERVER TASKID <%d> COULD

Message Number	Log Message
	NOT LOCK SNMP LOGGER SEMAPHORE"
NVD0552E	"CLIENT AUTOMATION CONFIGURATION SERVER TASKID <%d> COULD NOT UNLOCK SNMP LOGGER SEMAPHORE"
NVD0553E	"ERROR CREATING SNMP TRAP, RC = <%d>"
NVD0554E	"CLIENT AUTOMATION CONFIGURATION SERVER TaskID <%d> COULD NOT ALLOCATE PDU TRAP BUFFER OF <%d> BYTES"
NVD0555E	"CLIENT AUTOMATION CONFIGURATION SERVER TaskID <%d> INVALID SNMP LOGGER SEMAPHORE ID"
NVD0556E	"CLIENT AUTOMATION CONFIGURATION SERVER TaskID <%d> COULD NOT LOCK SNMP LOGGER'S SEMAPHORE"
NVD0557E	"CLIENT AUTOMATION CONFIGURATION SERVER TaskID <%d> COULD NOT UNLOCK SNMP LOGGER'S SEMAPHORE"
NVD0558E	"ERROR CREATING SNMP TRAP, RC = %d"
0600-0699	ZSNMPMGR
NVD0600W	"EDMSNMPG socket bind done, socket = <%d>"
NVD0601W	"Sending datagram, <%d> bytes to send"
NVD0602E	"ERROR SENDING DATAGRAM, RC = <%d>"
NVD0603W	"Receive for response timed out after <%d> seconds"
NVD0604E	"ERROR RECEIVING DATAGRAM, RC = <%d>"
NVD0605W	"Received datagram, <%d> bytes received"
NVD0606E	"TIMED OUT AFTER WAITING <%d> SECONDS FOR A RESPONSE"
NVD0607W	"Parameter <%d> = <%s>"
NVD0608E	"MISSING VALUE FOR SWITCH: -%c"
NVD0609E	"-%c SWITCH CANNOT HAVE A VALUE"
NVD0610E	"ONLY ONE OID ALLOWED WITH WALK SWITCH"
NVD0611E	"INVALID SWITCH: <%s>"
NVD0612E	"TOO MANY NON-SWITCH PARAMETERS: <%s>"
NVD0613E	"INVALID INTERNAL VALUE: <%d>"
NVD0614E	"TRACE LEVEL VALUE SPECIFIED <%s> IS INVALID."
NVD0615E	"HOSTNAME/IP ADDRESS OF AGENT IS REQUIRED."

Message Number	Log Message
NVD0616E	"AT LEAST ONE POSITIONAL PARAMETER (OID) IS REQUIRED"
NVD0617E	"INVALID VALUE IN OID <%s>"
NVD0618I	"Default community name \"public\" will be used."
NVD0619I	"Default port number (161) will be used."
NVD0620I	"Default timeout value (3 seconds) will be used."
NVD0621E	"TIMEDOUT VALUE SPECIFIED <%s> IS INVALID."
NVD0622I	"Default retries allowed value of 3 will be used."
NVD0623E	"RETRIES ALLOWED VALUE SPECIFIED <%s> IS INVALID."
NVD0624I	"Default SNMP version of 1 will be used."
NVD0625E	"SNMP VERSION SPECIFIED <%s> IS INVALID."
NVD0626I	"Default request id of 0 will be used."
NVD0627E	"REQUEST ID SPECIFIED <%s> IS INVALID."
NVD0628E	"ERROR BUILDING SNMP GET REQUEST, RC = <%d>"
NVD0629E	"RESPONSE HAS WRONG PDU TYPE = <%02X>"
NVD0630E	"RESPONSE WITH ERROR INDEX = <%d>, STATUS = <%d> = <%s>"
NVD0631E	"OID RECEIVED HAS TOO MANY IDS"
NVD0632E	"ERROR, OCTET STRING TOO LONG"
NVD0633E	"ERROR, UNSUPPORTED DATA TYPE RECEIVED"
NVD0634I	"%s - Snapping <%p> for length <0x%08x=%ld>"
NVD0635I	"<%08X>"
NVD0636W	"EDMSNMPS socket bind done, socket = <%d>"
NVD0637E	"+%c SWITCH CANNOT HAVE A VALUE"
NVD0638E	"TOO MANY OID PARAMETERS: <%s>"
NVD0639E	"TOO MANY TYPE PARAMETERS: <%s>"
NVD0640E	"TOO MANY VALUE PARAMETERS: <%s>"
NVD0641E	"AT LEAST THREE POSITIONAL PARAMETERS (OID, type, value) ARE REQUIRED"
NVD0642E	"MISSING TYPE/VALUE PARAMETER TO MATCH OID POSITIONAL PARAMETER."

Message Number	Log Message
NVD0643E	"MISSING VALUE PARAMETER TO MATCH OID POSITIONAL PARAMETER."
NVD0644E	"TYPE <%s> IS NOT SUPPORTED BY THIS PROGRAM"
NVD0645E	"INVALID TYPE <%s> SPECIFIED FOR OID <%s>"
NVD0646E	"INVALID CHARACTER FOR TYPE <%s> IN VALUE <%s>"
NVD0647E	"INVALID IP ADDRESS IN VALUE<%s>"
NVD0648E	"INVALID IP ADDRESS IN VALUE<%s> - MISSING DOT(s)"
NVD0649E	"INVALID VALUE IN OID VALUE<%s>"
NVD0650E	"VALUE FOR TYPE NULL MUST BE SPECIFIED AS 0"
NVD0651E	"INTERNAL ERROR, INVALID EDIT TYPE ENCOUNTERED"
NVD0652E	"ERROR BUILDING SNMP SET REQUEST, RC = <%d>"
NVD0653I	"SNMP Agent Task has started"
NVD0654E	"ERROR IN DosSetExceptionHandler : RETURN CODE = <%d>"
NVD0655E	"\"RUN_AS_EXTENSION = YES\" IS VALID ONLY WHEN RUNNING UNDER NT"
NVD0656E	"ERROR CREATING NAMED PIPE TO SNMP EXTENSION DLL, RC = <%d>"
NVD0657E	"SNMP EXTENSION DLL, RC = <%d>"
NVD0658E	"ERROR READING FROM SNMP EXTENSION DLL NAMED PIPE, RC = <%d>"
NVD0659E	"MESSAGE FROM DLL: reqType = <%d>, index = <%d>"
NVD0660E	"RECEIVED INCORRECT VERSION NUMBER <%d> FROM SNMP EXTENSION DLL"
NVD0661E	"MESSAGE FROM SNMP EXTENSION DLL NAMED PIPE IS TOO SHORT, READ <%d> BYTES"
NVD0662E	"ERROR WRITING TO SNMP EXTENSION DLL NAMED PIPE, RC = <%d>"
NVD0663E	"ERROR FLUSHING SNMP EXTENSION DLL NAMED PIPE, RC = <%d>"
NVD0664E	"ERROR DISCONNECTING SNMP EXTENSION DLL NAMED PIPE, RC = <%d>"
NVD0665E	"ERROR DISCONNECTING FROM SNMP EXTENSION DLL NAMED PIPE AFTER ERROR, RC = <%d>"

Message Number	Log Message
NVD0666E	"ERROR ESTABLISHING START UP TIME"
NVD0667E	"SNMP COMMUNITY NAME NOT SPECIFIED IN EDMPROF"
NVD0668E	"ERROR FORMATTING SNMP_MANAGER_IP_ADDR"
NVD0669E	"ERROR FORMATTING SNMP_MANAGER_IP_ADDR2"
NVD0670E	"ERROR FORMATTING SNMP_MANAGER_IP_ADDR3"
NVD0671E	"ERROR, COULD NOT BIND SNMP SOCKET TO <%s%s> PORT <%hd>"
NVD0672E	"ERROR, COULD NOT ALLOCATE SNMPZREQ BUFFER"
NVD0673E	"ERROR SENDING SNMP COLDSTART TRAP, RC = <%d>"
NVD0674E	"ERROR SENDING SNMP AUTH_FAILURE TRAP, RC = <%d>"
NVD0675I	"Received SNMP request with invalid set community name = <%s>"
NVD0676I	"Received SNMP request with invalid community name = <%s>"
NVD0677E	"SNMP ERROR SUMMARY FOLLOWS AS \"ERROR NUMBER - ERROR COUNT\"."
NVD0678E	"1-%04lu 2-%04lu 3-%04lu 4-%04lu 5-%04lu 6-%04lu 7-%04lu 8-%04lu 9-%04lu 10-%04lu"
NVD0679E	"11-%04lu 12-%04lu 13-%04lu 14-%04lu 15-%04lu 16-%04lu 17-%04lu 18-%04lu 19-%04lu 20-%04lu"
NVD0680E	"21-%04lu 22-%04lu 23-%04lu 24-%04lu 25-%04lu 26-%04lu 27-%04lu 28-%04lu 29-%04lu 30-%04lu"
NVD0681E	"31-%04lu 32-%04lu 33-%04lu 34-%04lu 35-%04lu 36-%04lu"
NVD0682E	"ERROR CREATING NAMED PIPE, RC = <%d>"
NVD0683E	"ERROR CONNECTING TO NAMED PIPE, RC = <%d>"
NVD0684I	"Shutting down Client Automation Configuration Server at request of SNMP set command"
NVD0685I	"Switched Client Automation Configuration Server logs at request of SNMP set command"
NVD0686I	"SNMP logging enabled by SNMP request"
NVD0687I	"SNMP logging disabled by SNMP request"
NVD0688I	"Event logging for critical log messages enabled by SNMP request"
NVD0689I	"Event logging for critical log messages disabled by SNMP request"

Message Number	Log Message
NVD0690I	"Command received from SNMP Manager: <%s>"
NVD0691I	"Turning ADMIN trace off at request of SNMP set command"
NVD0692I	"Turning ADMIN trace on at request of SNMP set command"
NVD0693E	"ZERROR TRAPPING SEVERITY SET TO <%lu> BY SNMP REQUEST"
NVD0694E	"%s SET TO <%lu> BY SNMP REQUEST"
NVD0695I	"Enabling all logons to the Client Automation Configuration Server at request of SNMP set command"
NVD0696I	"Disabling all logons to the Client Automation Configuration Server at request of SNMP set command"
NVD0697I	"Turning <%s> all Client Automation Configuration Server tracing at request of SNMP set command"
NVD0698I	"Turning <%s> trace <%s> at request of SNMP set command"
NVD0699I	"SNMP Event: <%s>"
0700-0749	ZCVT and ZTCBG
NVD0701E	"ERROR, COULD NOT FIND ZCVT OF CLIENT AUTOMATION CONFIGURATION SERVER TASK <%ld>"
NVD0702E	"ERROR, COULD NOT FIND ZTCBG OF CLIENT AUTOMATION CONFIGURATION SERVER TASK <%ld>"
0750-0799	ZMCSTMGR
NVD0750I	"Multicast Manager Task has started"
NVD0751E	"COULD NOT CREATE MULTICAST WORK DIRECTORY"
NVD0752E	"UNABLE TO INITIALIZE MULTICAST SEGMENT"
NVD0753E	"ERROR ADDING NEW MULTICAST ENTRY <%s>"
NVD0754E	"ERROR ADDING QUEUE ENTRY <%8.8s>"
NVD0755E	"FAILED ATTACHING ZMCASTER - RC = <%d>"
NVD0756I	"Started multicast task for <%s>"
NVD0757I	"Requesting all multicast tasks to shutdown"
NVD0758E	"MULTICAST TASK HAS NOT BEEN ENABLED."
NVD0759E	"ERROR LOADING MULTICAST DATA FROM ZMASTER"
NVD0760E	"ERROR WRITING TO MULTI CAST QUEUE RC <%d>"

Message Number	Log Message
NVD0761E	"DOMAIN VARIABLE <%8.8s> NOT FOUND FOR MULTICAST INSTANCE"
NVD0762E	"CLASS VARIABLE <%8.8s> NOT FOUND FOR MULTICAST INSTANCE"
NVD0763E	"INSTANCE VARIABLE <%8.8s> NOT FOUND FOR MULTICAST INSTANCE"
NVD0764E	"INVALID DOMAIN NAME FOR MULTICAST"
NVD0765E	"INVALID CLASS NAME FOR MULTICAST"
NVD0766E	"INVALID INSTANCE NAME FOR MULTICAST"
NVD0767E	"MISSING <%s> VARIABLE FOR MULTICAST"
NVD0768E	"INVALID <%s> FOR MULTICAST"
0800-0899	ZCONMGR and ZLINECMD
NVD0801E	"CLIENT AUTOMATION CONFIGURATION SERVER - INVALID COMMAND: <%s>"
NVD0802E	"INVALID REFRESH COMMAND: <%s>"
NVD0803E	"REFRESH COMMAND CANNOT DETERMINE PARMLIB DATASET NAME, RC <%d>"
NVD0804E	"INVALID PARMLIB MEMBER NAME IN REFRESH COMMAND <%s>"
NVD0805E	"REFRESH COMMAND CANNOT OPEN <%s>"
NVD0806E	"REFRESH COMMAND CANNOT READ <%s>"
NVD0807E	"REFRESH FAILED TO PROCESS MGR_TRACE in <%s>"
NVD0808E	"REFRESH FAILED TO PROCESS MGR_MESSAGE_CONTROL in <%s>"
NVD0809E	"REFRESH FAILED TO PROCESS MGR_POOLS - GLOBAL HEAP NOT FOUND "
NVD0810E	"REFRESH FAILED TO PROCESS MGR_POOLS in <%s>"
NVD0811I	"Refresh from <%s> completed successfully"
NVD0812E	"INVALID QUEUE COMMAND: <%s>"
NVD0813E	"INVALID QUEUE NAME IN COMMAND: <%s>"
NVD0814E	"NO QUEUE NAME IN COMMAND: <%s>"
NVD0816E	"NO REXX COMMAND NAME IN: <%s>"
NVD0817E	"NO IP ADDRESS IN: <%s>"

Message Number	Log Message
NVD0818E	"NO PORT NUMBER IN: <%s>"
NVD0819E	"NO USERID IN: <%s>"
NVD0820E	"NOTIFY NOT SUPPORTED BY MODIFY COMMAND"
NVD0821I	"Command Accepted: <%s>"
NVD0823E	"INVALID MODIFY COMMAND: <%s>"
NVD0824E	"INVALID DISPLAY SUBCOMMAND IN: <%s>"
NVD0825I	"Display Active:"
NVD0826I	"Heaps To_Snd Sent Rslved CurrObj Method State"
NVD0827I	"%5hd %6hd %6hd %6hd %8s %8s %s"
NVD0828I	"No active Client Automation Agent connects found"
NVD0829I	"Display Status:"
NVD0830I	"%5d - %s"
NVD0831E	"INVALID POOL SUBCOMMAND IN: <%s>"
NVD0832E	"POOL STATS: COMMAND FAILED - ZCVT NOT FOUND"
NVD0833E	"POOL STATS: COMMAND FAILED - GLOBAL HEAP NOT INITIALIZED"
NVD0834I	"POOL STATS: - None, Heap is disabled"
NVD0835E	"POOL STATS: NO TASK FOUND WITH NAME <%s>"
NVD0836I	"POOL STATS for <%s>:"
NVD0837I	"Size Number Waste InUse HighWM Allocs Empty Waste"
NVD0838I	"%7d%c %6d %3d %6lu %6lu %11lu %11lu %11lu"
NVD0839E	"POOL MISSES: COMMAND FAILED - ZCVT NOT FOUND"
NVD0840E	"POOL MISSES: COMMAND FAILED - GLOBAL HEAP NOT INITIALIZED"
NVD0841I	"POOL MISSES: - None, Heap is disabled"
NVD0842E	"POOL MISSES: - ERROR, TASK NAME WAS NOT SPECIFIED"
NVD0843E	"POOL MISSES: NO TASK FOUND WITH NAME <%s>"
NVD0844I	"POOL MISSES for <%s>:"
NVD0845I	"%9.9d %8.8s/%d"

Message Number	Log Message
NVD0846I	"Misses table is empty"
NVD0847E	"POOL DISABLE: COMMAND FAILED - ZCVT NOT FOUND"
NVD0848E	"POOL DISABLE: COMMAND FAILED - GLOBAL HEAP NOT INITIALIZED"
NVD0849I	"POOL DISABLE: - None, Heap is disabled"
NVD0850E	"POOL DISABLE: - NO POOL SIZE IN COMMAND"
NVD0851E	"POOL DISABLE: - POOL SIZE IS NON-NUMERIC"
NVD0852E	"POOL DISABLE: - POOL SIZE <%D> NOT DISABLED \n"
NVD0853I	"POOL DISABLE: Command successful - Pool size <%d> disabled"
NVD0854E	"POOL ENABLE: COMMAND FAILED - ZCVT NOT FOUND"
NVD0855E	"POOL ENABLE: COMMAND FAILED - GLOBAL HEAP NOT INITIALIZED"
NVD0856E	"POOL ENABLE: - ERROR, HEAP IS DISABLED"
NVD0857E	"POOL ENABLE: - NO POOL SIZE IN COMMAND"
NVD0858E	"POOL ENABLE: - POOL SIZE IS NON-NUMERIC"
NVD0859E	"POOL ENABLE: - COMMAND FAILED - POOL SIZE <%d> NOT ENABLED"
NVD0860I	"POOL ENABLE: Command successful - Pool size <%d> enabled"
NVD0861E	"POOL CLEAR COUNTERS: COMMAND FAILED - ZCVT NOT FOUND"
NVD0862E	"POOL CLEAR COUNTERS: COMMAND FAILED - GLOBAL HEAP NOT INITIALIZED"
NVD0863E	"POOL CLEAR COUNTERS: - ERROR, HEAP IS DISABLED"
NVD0864E	"POOL CLEAR COUNTERS: - NO POOL SIZE IN COMMAND"
NVD0865E	"POOL CLEAR COUNTERS: - POOL SIZE IS NON-NUMERIC"
NVD0866E	"POOL CLEAR COUNTERS: - COMMAND FAILED - COUNTERS FOR POOL SIZE <%d> NOT CLEARED"
NVD0867I	"POOL CLEAR COUNTERS: Command successful - Counters for pool size <%d> cleared"
NVD0868E	"POOL ADJUST: COMMAND FAILED - ZCVT NOT FOUND"
NVD0869E	"POOL ADJUST: COMMAND FAILED - GLOBAL HEAP NOT INITIALIZED"
NVD0870E	"POOL ADJUST: - NO POOL SIZE IN COMMAND"
NVD0871E	"POOL ADJUST: - NO NUMBER OF ELEMENTS IN COMMAND"

Message Number	Log Message
NVD0872E	"POOL ADJUST: - POOL SIZE IS NON-NUMERIC"
NVD0873E	"POOL ADJUST: - POOL SIZE IS INVALID"
NVD0874E	"POOL ADJUST: - NUMBER OF ELEMENTS IS NON-NUMERIC"
NVD0875E	"POOL ADJUST: - NUMBER OF ELEMENTS IS INVALID"
NVD0876E	"POOL ADJUST: - PERCENT WASTE TOLERATED IS NON-NUMERIC"
NVD0877E	"POOL ADJUST: - PERCENT WASTE TOLERATED <%lu> IS NON-NUMERIC"
NVD0878E	"POOL ADJUST: - COMMAND FAILED - POOL SIZE <%lu> NOT ADJUSTED "
NVD0879I	"POOL ADJUST: - Command successful - Pool size <%lu> adjusted "
NVD0880I	"OTHER: %lu allocs, %lu bytes, %lu largest, %lu expansion"
NVD0881I	"Console Manager received command to shutdown everything"
NVD0882I	"Console Manager ending, shutdown detected"
NVD0883I	"Console Manager received the following MVS modify command:"
NVD0884I	"===> %s"
NVD0885I	"%s: Received command longer than %d"
NVD0886I	"Received command to flush the log"
NVD0887I	"Received command to switch the log"
NVD0888I	"Received command to shutdown everything"
NVD0889I	"Received a display command"
NVD0890I	"Received a pool command"
NVD0891I	"Received command to log version information"
NVD0892I	"Received command to queue a request"
NVD0893I	"Received command to refresh trace options"
NVD0894I	"Debug command, tasklim = %i"
NVD0895I	"Debug command, dumping ztcbg number %i"
NVD0896I	"%s *%. *s"
0900-0999	ZSMTRMGR and ZSMTSMGR

Message Number	Log Message
NVD0901I	"%s task has started"
NVD0902E	"ERROR LOADING TCPIP LIBRARY, RC = <%d>"
NVD0903I	"SMTP Port# <%04d>"
NVD0904E	"ATBALLOC ERROR - <%s>"
NVD0905E	"APIOPEN ERROR - <%s>"
NVD0906E	"APIACCEPT ERROR - <%s>"
NVD0907E	"ERROR ZATTACH INTERNAL FAILED - RC = <%d>"
NVD0908I	"Requesting all SMTP ZSMTPRCVs to shutdown"
NVD0909I	"Terminating Method <%d> of Client Automation Configuration Server Task <%d>"
NVD0910I	"Requesting Client Automation Configuration Server Task <%d> to shutdown"
NVD0911I	"Terminating Client Automation Configuration Server Task <%d>"
NVD0912I	"SMTP Send Task has started"
NVD0913E	"COULD NOT VALIDATE SPOOL DIRECTORY"
NVD0914E	"MESSAGE QUEUE WAS NOT INITIALIZED, UNABLE TO CONTINUE"
NVD0915E	"ERROR CREATING MAIL SEMAPHORE"
NVD0916E	"ERROR CREATING MAIL EVENT SEMAPHORE"
NVD0917I	"Requesting SMTP Spooler & Queuer to shutdown"
NVD0918E	"ERROR GENERATING MAIL_ID"
NVD0919E	"ERROR LOCKING MAIL SEMAPHORE"
NVD0920E	"ERROR SPOOLING MAIL TO <%s>"
NVD0921E	"ERROR UNLOCKING MAIL SEMAPHORE"
NVD0922E	"ERROR CREATING MAIL EVENT SEMAPHORE"
NVD0923I	"Mail cannot be spooled <%s>"
NVD0924I	"Cannot open mail queue file <%s>"
NVD0925E	"MISSING MAIL MUTEX SEMAPHORE"
NVD0926E	"MISSING MAIL EVENT SEMAPHORE"
NVD0927E	"SPOOL CONTAINS AN INVALID FILE <%s>"

Message Number	Log Message
NVD0928E	"ERROR OPENING SPOOL FILE <%s>"
NVD0929E	"MAIL TO <%s> IS CORRUPTED"
NVD0930I	"Mail spooler task started"
NVD0931E	"ERROR OPENING MAIL SEMAPHORE"
NVD0932E	"ERROR OPENING MAIL EVENT SEMAPHORE"
NVD0933I	"Mail Spooler task has ended"
NVD0934E	"INIT_ERROR : <%d> <%s> "
NVD0935E	"OPEN_ERROR : <%d> <%s> "
NVD0936E	"CONNECTION REFUSED <%s>"
NVD0937E	"HELO ERROR <%s>"
NVD0938E	"MAIL FROM ERROR <%s>"
NVD0939E	"RCPT TO ERROR <%s>"
NVD0940E	"MAIL DATA ERROR <%s>"
NVD0941I	"Attachment file: <%s> not found"
NVD0942I	"Routing mail via: <%s> "
NVD0943E	"INCORRECT SEQUENCE OF COMMANDS"
NVD0944E	"INVALID FORWARD PATH <%s>"
NVD0945E	"INVALID FORWARD PATH LENGTH <%s>"
NVD0946W	"Message Relaying not enabled <%s>"
NVD0947E	"CANNOT ACCEPT MAIL FOR <%s>"
NVD0948E	"ERROR GENERATING TEMPORARY FILE NAME"
NVD0949E	"ERROR CREATING DIRECTORY <%s>"
NVD0950E	"ERROR CREATING FILE <%s>"
NVD0951I	"e-mail received in <%s>"
1000-1099	Allocation and Expansion
NVD1001E	"ERROR, UNABLE TO ALLOCATE MEMORY FOR OBJECT<%s>"
NVD1002E	"ERROR, UNABLE TO ALLOCATE STORAGE FOR BUFFER"

Message Number	Log Message
NVD1003E	"ERROR, UNABLE TO REALLOCATE MEMORY"
NVD1004E	"ERROR, UNABLE TO EXPAND"
NVD1005E	"ERROR, UNABLE TO ALLOCATE STORAGE FOR TEMPLATE"
NVD1006E	"ERROR, UNABLE TO ALLOCATE STORAGE FOR HEAP <%lu>"
NVD1007E	"ERROR, UNABLE TO ALLOCATE STORAGE FOR INSTANCE <%s.%s.%s.%s>"
NVD1008E	"ERROR, UNABLE TO ALLOCATE STORAGE FOR BASE INSTANCE <%s.%s.%s>"
NVD1009E	"ERROR, UNABLE TO ALLOCATE STORAGE FOR CLASS <%s.%s.%s>"
NVD1010E	"ERROR, UNABLE TO EXPAND TEMPLATE"
NVD1011E	"ERROR, UNABLE TO ALLOCATE MEMORY FOR <%s>"
NVD1012E	"ERROR, CANNOT CREATE A VALID POINTER"
NVD1013E	"ERROR, UNABLE TO ALLOCATE <%d> BYTES OF MEMORY FOR <%s>"
NVD1014E	"ERROR OBTAINING SHARED MEMORY COPY"
NVD1015I	"Expand DIRECTORY from <%d> to <%d>"
NVD1016I	"Directory Old: <%p> New: <%p>"
NVD1017E	"ERROR ALLOCATING STORAGE FOR <%d> BYTES"
NVD1018IE	"M_FREE=<%p> in <%s-%d> num_bytes is <%d>"
NVD1019E	"INSUFFICIENT MEMORY-SIZE <%s %d %d>"
NVD1020E	"MACHINE MIGHT BE OUT OF SWAP SPACE"
NVD1021E	"RLIMIT_DATA (HEAP) (SOFT) <%10.10d> (HARD) <%10.10d>"
NVD1022I	"M_MALLOC=<%p> from <%s-%d> num_bytes is <%d>"
NVD1023I	"Increasing storage size from <%d> to <%d> bytes"
NVD1024I	"Last block of record length <%d> bytes"
NVD1025I	"Received record of length <%d> bytes"
NVD1026E	"VARIABLE SERVER - STORAGE UNAVAILABLE"
NVD1027E	"INSUFFICIENT BUFFER SIZE"
NVD1028E	"ERROR IN LOCATING HEAP <%s>"

Message Number	Log Message
NVD1029I	"Initial heap index allocated <%lu> bytes @ <%08x>"
NVD1030I	"Expanded heap index allocated <%lu> bytes @ <%08x>"
NVD1031I	"Old heap index @ <%08x> freed"
NVD1032I	"Heap allocated <%lu> bytes @ <%08x>"
NVD1033I	"Heap storage @ <%08x> freed"
NVD1034I	"Heap index storage @ <%08x> freed"
NVD1035I	"%s: index @ <%08x>, index_i <%ld> @ <%08x>, index_j <%ld> @ <%08x>"
NVD1036I	"%s: index_i->zheapno <%ld>, index_i->zhpaddr <%08x>"
NVD1037I	"Heap index allocated <%ld> bytes @ <%08x>"
NVD1038E	"ERROR ALLOCATING NEW INDEX STORAGE FOR OBJECT <%s>"
NVD1039E	"ERROR ALLOCATING EXPANDED HEAP STORAGE FOR OBJECT <%s>"
NVD1040W	"Below 16M storage limit exceeded, limit = <%lu>"
NVD1041W	"Above 16M storage limit exceeded, limit = <%lu>"
NVD1042I	"Total available storage < 16M is <%lu>, largest fragment is <%lu>"
NVD1043I	"Total available storage > 16M is <%lu>, largest fragment is <%lu>"
NVD1044I	"Total available storage is <%lu>, largest fragment is <%lu>"
NVD1045I	"Allocating index cache shared memory <%lu> bytes"
NVD1046E	"ERROR CREATING INDEX CACHE SHARED MEMORY"
NVD1047E	"ERROR ALLOCATING INDEX CACHE SHARED MEMORY"
NVD1048I	"Allocating index cache memory; fixed area=<%lu> bytes; entry_table=<%lu> bytes"
NVD1049E	"ERROR UNABLE TO ALLOCATE MEMORY FOR PROCESSING TEMPLATE"
NVD1050E	"ERROR ALLOCATING STORAGE MEMORY FOR QUEUE ENTRY"
NVD1051I	"Total available storage > 16M is <%llu>, largest fragment is <%llu>, credit = <%llu>"
NVD1052I	"Total available storage is <%llu>, total is <%llu>, credit = <%llu>"
NVD1053E	"BELOW 16MB STORAGE LIMIT EXCEEDED, LIMIT = <%llu>"
NVD1054E	"ABOVE 16MB STORAGE LIMIT EXCEEDED, LIMIT = <%llu>"

Message Number	Log Message
NVD1055E	"STORAGE LIMIT EXCEEDED, LIMIT = <%llu>"
NVD1056E	"SHORT ON STORAGE, NEEDED <%llu> BYTES ABOVE THE LINE BUT HAVE ONLY <%llu-%llu = %llu>"
NVD1057E	"SHORT ON STORAGE, NEEDED <%llu> BYTES BUT HAVE ONLY <%llu-%llu = %llu>"
NVD1058E	"SHORT ON STORAGE, NEEDED <%llu> CONTIGUOUS BYTES ABOVE THE LINE BUT HAVE ONLY <%llu>"
NVD1059E	"SHORT ON STORAGE, NEEDED <%llu> CONTIGUOUS BYTES BUT HAVE ONLY <%llu>"
NVD1060E	"ERROR, CALL TO zsys_querySystemMemory() FAILED"
NVD1061E	"ERROR COPYING TEMPLATE ENTRIES"
NVD1062E	"#ENTRIES <%d> #COPIED <%d> "
NVD1063I	"Total number of local memory allocations: <%i>"
NVD1064I	"Total number of local memory frees: <%i>"
NVD1065I	"Total number of local buffers reuse: <%i>"
NVD1066I	"Number of template memory reuse: <%i>"
NVD1067I	"Number of base instance memory reuse: <%i>"
NVD1068I	"Number of instance memory reuse: <%i>"
NVD1069I	"Max size of local memory allocated: <%i>"
NVD1070E	"OBJECT <%0.8s> TEMPLATE DIRECTORY EXPANSION SIZE <%lu> IS GREATER THAN ALLOWED MAX SIZE <%lu>"
NVD1071W	"Unable to create area for call to ZLOGSWCH/ZLOGWRAP REXX, skipping REXX call"
NVD1072E	"EXCEPTION <%s> IN <%s> AT <%X>"
NVD1073E	"ACCESS VIOLATION: ATTEMPTED TO %s <%X>"
1100-1199	Heap
NVD1100E	"ERROR, EXPANSION ERROR FOR <%s.%s>"
NVD1101E	"ERROR LOCATING HEAP <%lu>"
NVD1102E	"ERROR, HEAP <%lu> NOT FOUND FOR <%s>"
NVD1103E	"ERROR, HEAP SIZE <%d> EXCEEDS ALLOCATED BUFFER SPACE <%d>"

Message Number	Log Message
NVD1104E	"ERROR INITIALIZING HEAPS"
NVD1105I	"<%s> Heap queue allocated at <%p>"
NVD1106I	"<%8.8s> Heap queue allocated at <%p>"
NVD1107E	"ERROR <%s> HEAP <%d>"
NVD1108E	"ERROR INITIALIZING HEAP FOR OBJECT <%s>"
NVD1109E	"ERROR ALLOCATING HEAP FOR OBJECT"
NVD1110E	"ERROR SORTHEAP - VARIABLE <%s> NOT FOUND"
NVD1111I	"Heap Sent Before Template (O) <%8.8s> (T) <%8.8s> (S) <%d>"
NVD1112I	"Heap Length <%d> bytes"
NVD1113I	"Heap <%lu>"
NVD1114E	"ERROR, TEMPLATE VARIABLE <%.0s> DATA OUTSIDE OF HEAP BUFFER"
NVD1115E	"ERROR, HEAP SIZE <%d> LESS THAN TEMPLATE OFFSET <%d>, LENGTH <%d>"
NVD1116E	"ERROR SETTING OBJECT QUEUE <%s>"
NVD1117E	"ERROR SORTHEAP - PATCHALL NOT FOUND"
NVD1118E	"ERROR ALLOCATING STORAGE FOR HEAP"
NVD1119E	"ERROR INVALID OBJECT NAME"
NVD1120I	"Heap <%lu> @ <%08x> added to index @ <%08x> offset <%08x>"
NVD1121E	"ERROR HEAP SIZE <%d> GREATER THAN MAX HEAP SIZE ALLOWED <%lu>"
NVD1122I	"Adding heap <%lu> @ <%08x> to multi-heap index"
NVD1123I	"Heap <%lu> copied from @ <%08x> to @ <%08x>"
NVD1124I	"Heap <%d> not found in multi-heap index"
NVD1125I	"%s: ZPCB @ <%08x>, sort list @ <%08x>, sort order <%ld>"
NVD1126I	"%s: varentry @ <%08x>, name <%8.8s>, dspl <%04x>, len <%04x>"
NVD1127E	"ERROR, HEAP NUMBER <%d> NOT FOUND IN OBJECT <%s>"
NVD1128E	"ERROR, NO HEAP DATA POINTER FOR HEAP NUMBER <%lu> OBJECT <%s>"

Message Number	Log Message
NVD1129E	"ERROR, NO HEAPS ALLOCATED FOR OBJECT <%s>"
NVD1130E	"ERROR, HEAP EXPANSION NOT NEEDED FOR OBJECT <%s>"
NVD1131I	"%s returns heap <%lu> @ <%08x>"
NVD1132I	"%s returns heap queue @ <%08x>"
NVD1133E	"ERROR RE-INITIALIZING HEAP FOR <%s> WITH CONFLICTING SIZES OLD=<%d> NEW=<%d>"
NVD1134I	"Heap queue for object <%s> located @ <%08x>"
NVD1135E	"HEAP NUMBER <%lu>, CANNOT EXCEED <%d> FOR OBJECT <%s>"
NVD1136E	"%s REQUESTED HEAP NUMBER IS ZERO AND CURRENT HEAP IS UNDEFINED"
NVD1137E	"SORTHEAP - OBJECT NOT FOUND <%s>"
NVD1138I	"Requested heap number (%lu) higher than current last heap: (%lu)"
NVD1139I	"Object heap will be...: (%lu)"
NVD1140E	"INVALID HEAP NUMBER PASSED =<%lu>, MAXIMUM IS <%lu>"
NVD1141I	"Heap number <%lu> invalid for <%s>; maximum heaps=<%lu>"
NVD1142E	"ERROR ALLOCATING MEMORY FOR PROCESSING THE HEAP"
NVD1143E	"INSTANCE NUMBER > MAXHEAP + 1 %s %s %lu (Max) %lu"
NVD1144E	"ERROR LOCATING HEAP FOR <%s>"
NVD1145E	"ERROR IN MALLOC OF <%s>"
NVD1146E	"ERROR ALLOCATING HEAP <%s>"
NVD1147I	"s *%.*s"
1500-1599	Cache
NVD1500I	"Loading Cache..."
NVD1501E	"ERROR INITIALIZING CACHE IPC FOR CLIENT AUTOMATION AGENT"
NVD1502E	"ERROR INITIALIZING CACHE FOR CLIENT AUTOMATION AGENT TASK"
NVD1503E	"INSUFFICIENT MEMORY FOR CACHE"
NVD1504E	"ERROR ON GETMAIN FOR CACHE MEMORY SEGMENT <%d>"
NVD1505E	"ERROR ON CREATING CACHE SHARED MEMORY SEGMENT <%d>"

Message Number	Log Message
NVD1506E	"ERROR ON ATTACHING CACHE SHARED MEMORY <%d> (PARENT)"
NVD1507E	"ERROR ON ATTACHING CACHE SHARED MEMORY <%d> (CHILD)"
NVD1508E	"ERROR ON DETACHING CACHE SHARED MEMORY SEGMENT <%d> <%d> (PARENT)"
NVD1509E	"ERROR ON REMOVING CACHE SHARED MEMORY SEGMENT <%d> (PARENT)"
NVD1510E	"ERROR ON DETACHING CACHE SHARED MEMORY SEGMENT <%d> <%d> (CHILD)"
NVD1511I	"Cache Page <%d> Free Space <%d> Reclaimed <%d>"
NVD1512W	"Free block <%d> overlaps with <%d>"
NVD1513E	"ERROR ON REMOVING INDEX CACHE ENTRY FOR <%s.%s.%s.%s>"
NVD1514I	"Invalidated cache entry for <%s.%s.%s.%s>"
NVD1515W	"Cached size <%d> differs from new file size <%d>"
NVD1516E	"CACHED SIZE <%d> DIFFERS FROM CLASS HEAP LENGTH <%d> OFFSET <%d>"
NVD1517E	"CACHE NOT UPDATED FOR INSTANCE <%s.%s.%s.%s>"
NVD1518I	"Could not find <%s.%s.%s> in content cache"
NVD1519E	"ERROR ON LOCKING <%s.%s.%s> NOT CACHED"
NVD1520E	"UNABLE TO REFRESH <%s.%s.%s> CONTROL INFORMATION, REMOVED FROM CACHED"
NVD1521W	"Class <%s.%s.%s> removed from cache"
NVD1522W	"Unable to open edmpref file -- caching disabled"
NVD1523W	"No cache list specified or cache list is empty"
NVD1524I	"Cache Rebuilt"
NVD1525W	"Unable to cache <%s.%s.%s.%s>, cache filled"
NVD1526E	"ERROR ON CACHING INSTANCE FOR <%s.%s.%s>"
NVD1527I	"Updated cache entry for <%s.%s.%s.%s>"
NVD1528E	"INSUFFICIENT CACHE SIZE FOR <%d> CONTENT CACHE ENTRIES"
NVD1529I	"Added Class <%s.%s.%s> to Content Cache"
NVD1530E	"ERROR ON ADDING CLASS <%s.%s.%s> TO CONTENT CACHE"

Message Number	Log Message
NVD1531I	"Added Instance <%s.%s.%s.%s> to Content Cache"
NVD1532E	"ERROR ON ADDING INSTANCE <%s.%s.%s.%s> TO CONTENT CACHE"
NVD1533E	"ERROR ON LOCKING CACHE SEMAPHORE"
NVD1534I	"***** Page # %d *****"
NVD1535I	"Key Offset Size PID T S TD_Stamp Hits"
NVD1536I	"FREE LIST"
NVD1537I	"# OFFSET FREE LENGTH"
NVD1538I	"%d %ld %ld"
NVD1539I	"%5.5d %10.10lu %10.10lu,%10.10d %10.10d %10.10d %c %c %12.12s %10.10d "
NVD1540I	"Index cache age pools freed <%ld> bytes @ %08X"
NVD1541I	"Index cache disabled, icache_size = <%ld>"
NVD1542E	"ERROR ON ATTACHING INDEX CACHE SHARED MEMORY (CHILD)"
NVD1543I	"Index cache initialized, icache_prefix @ <%08X>"
NVD1544I	"Index cache cleanup not required, icache_ptr = <%08X>"
NVD1545E	"ERROR ON DETACHING INDEX CACHE SHARED MEMORY (PARENT)"
NVD1546E	"ERROR ON REMOVING INDEX CACHE SHARED MEMORY SEGMENT (PARENT)"
NVD1547E	"ERROR ON DETACHING INDEX CACHE SHARED MEMORY SEGMENT (CHILD)"
NVD1548W	"Index cache deleted directory failed, directory not found for <%s.%s.%s>"
NVD1549I	"Index cache deleted directory invalidated class <%s.%s.%s>"
NVD1550E	"INDEX CACHE DELETE ENTRY FAILED, DIRECTORY NOT FOUND FOR CLASS <%s.%s.%s>"
NVD1551E	"INDEX CACHE DELETE ENTRY FAILED, ENTRY NOT FOUND FOR INSTANCE <%s.%s.%s.%s>"
NVD1552I	"Index cache delete entry <%ld> successful for <%s.%s.%s.%s>"
NVD1553I	"Index cache freed <%ld> bytes @ <%08X>"
NVD1554E	"INDEX CACHE EXPAND ENTRY ARRAY FROM <%ld> TO <%ld> BYTES FAILED FOR INSTANCE <%s.%s.%s.%s>"

Message Number	Log Message
NVD1555I	"Index cache entry array expanded from <%ld> to <%ld> for instance <%s.%s.%s.%s>"
NVD1556I	"Index cache bsearch entry for <%s.%s.%s.%s>"
NVD1557I	"Index cache bsearch entry try <%ld> min <%ld> mid <%ld> max <%ld> cmp <%ld> found <%s.%s.%s.%s> @ <%08X>"
NVD1558I	"Index cache bsearch entry <%ld> @ <%08X> try <%ld> found <%s.%s.%s.%s>"
NVD1559W	"Index cache add entry <%s> not cached, type <%c> invalid for <%s.%s.%s>"
NVD1560W	"Index cache add entry <%s> not cached, directory not found for <%s.%s.%s>"
NVD1561W	"Index cache add entry <%s> not cached, directory invalidated for <%s.%s.%s>"
NVD1562I	"Index cache add entry <%ld> found @ <%08X> for <%s.%s.%s.%s>"
NVD1563I	"Index cache add entry <%ld> created @ <%08X> for instance <%s.%s.%s.%s>"
NVD1564E	"ERROR, INDEX CACHE GET ENTRY CELL NOT FOUND FOR INSTANCE <%s.%s.%s.%s>"
NVD1565I	"Index cache get entry cell <%ld> found @ %08X for instance <%s.%s.%s.%s>"
NVD1566I	"Index cache get entry cell not found for instance <%s.%s.%s.%s>"
NVD1567I	"Index cache find directory cell for <%s.%s.%s>"
NVD1568I	"Index cache find directory cell, found @ <%08X> for <%s.%s.%s>"
NVD1569I	"Index cache get dir cell <%ld> allocated @ <%08X>"
NVD1570I	"Index cache find first entry type <%c>, p_objpkt <%08X>, p_ich <%08X>"
NVD1571E	"ERROR, INDEX CACHE FIND FIRST ENTRY INVALID ARGS"
NVD1572E	"ERROR, INDEX CACHE FIND FIRST ENTRY, DIRECTORY NOT FOUND FOR <%s.%s.%s>"
NVD1573E	"ERROR, INDEX CACHE FIND FIRST ENTRY, NO INSTANCES FOUND FOR <%s.%s.%s>"
NVD1574I	"Index cache find next entry <%ld> EOF"
NVD1575I	"Index cache find first entry <%ld> found @ <%08X> for <%s.%s.%s.%s>"
NVD1576I	"Index cache find first entry directory <%ld> found @ <%08X> for

Message Number	Log Message
	<%s.%s.%s>“
NVD1577E	“ERROR, INDEX CACHE FIND FIRST ENTRY CALLED WITH INVALID REQUEST TYPE <%c>“
NVD1578E	“ERROR, INDEX CACHE FIND NEXT ENTRY CALLED BEFORE FIND FIRST ENTRY”
NVD1579E	“ERROR, INDEX CACHE FIND NEXT ENTRY DETECTED CHANGE FOR CLASS <%s.%s.%s>“
NVD1580E	“ERROR, INDEX CACHE FIND NEXT ENTRY, NO INSTANCES FOUND FOR CLASS <%s.%s.%s>“
NVD1581I	“Index cache next instance entry <%ld> @ <%08X> is <%s.%s.%s.%s>“
NVD1582I	“Index cache find next entry directory <%ld> found @ <%08X> for <%s.%s.%s>“
NVD1583I	“Index cache find next entry directory <%ld> found @ <%08X> for <%s.%s>“
NVD1584I	“Index cache find next entry directory <%ld> found @ <%08X> for <%s>“
NVD1585E	“ERROR, INDEX CACHE FIND NEXT ENTRY CALLED WITH INVALID REQUEST TYPE <%c>“
NVD1586I	“Index cache find next entry type <%c> EOF”
NVD1587E	“ERROR, INDEX CACHE DID NOT FIND ANY CLASSES IN DOMAIN <%s.%s>“
NVD1588E	“ERROR, INDEX CACHE DID NOT FIND ANY DOMAINS IN FILE <%s>“
NVD1589E	“INDEX CACHE ADD <%s.%s.%s> FAILED, INSUFFICIENT DIRECTORY CELLS ARE AVAILABLE”
NVD1590I	“Initializing index cache directory @ <%08X> for class <%s.%s.%s>“
NVD1591I	“Index cache initialization found <%ld> FDC database objects in <%-s>“
NVD1592I	“Index cache allocating <%ld> FDC directory entries”
NVD1593I	“Allocating index cache prefix and directory <%ld> bytes”
NVD1594E	“ERROR ATTACHING INDEX CACHE SHARED MEMORY (PARENT)”
NVD1595I	“Initialized index cache prefix memory <%ld> bytes @ <%08X>“
NVD1596I	“Found index cache prefix memory @ <%08X>“
NVD1597W	“Index cache cannot access class control information for class <%s.%s.%s>“
NVD1598W	“Index cache cannot read class control information for class <%s.%s.%s>“

Message Number	Log Message
NVD1599W	"Index cache cannot validate DMA instance count for class <%s.%s.%s>"
1600-1699	Cache
NVD1600E	"INDEX CACHE CAN NOT FIND CLASS <%s.%s.%s>"
NVD1601I	"Building index cache for class <%s.%s.%s>, loading <%ld> instances"
NVD1602I	"Index cache entry <%ld> added @ <%08X> for instance <%s.%s.%s.%s>"
NVD1603W	"Class <%s.%s.%s> DMA TotalInstanceCount <%ld> does not match actual instance count"
NVD1604I	"Class <%s.%s.%s> added to Index Cache, loaded <%ld> instances"
NVD1605I	"Initialized index cache memory; directory cells=<%ld>; entry cells=<%ld>"
NVD1606W	"Delete Directory from Index Cache failed because (directory not found): crc=<%IX> type=<%c>"
NVD1607I	"Deleted Entry <%ld> from Index Cache: name=<%s>"
NVD1608W	"Deleted Directory <%ld> from Index Cache failed because (directory not found): crc=<%IX> type=<%c>"
NVD1609W	"Delete Entry from Index Cache failed because (directory not found): crc=<%IX> type=<%c> name=<%s>"
NVD1610I	"Deleted first Entry <%ld> from Index Cache: crc=<%IX> type=<%c> name=<%s>"
NVD1611I	"Deleted last Entry <%ld> from Index Cache: crc=<%IX> type=<%c> name=<%s>"
NVD1612I	"Deleted Entry <%ld> from Index Cache: crc=<%IX> type=<%c> name=<%s>"
NVD1613W	"Delete Entry from Index Cache failed because (entry not found): crc=<%IX> type=<%c> name=<%s>"
NVD1614W	"Add to Index Cache failed because (this category is not cached): crc=<%IX> type=<%c> name=<%s>"
NVD1615W	"Add to Index Cache failed because (free entry cell not found): crc=<%IX> type=<%c> name=<%s>"
NVD1616I	"Index Cache Entry <%ld> added first for crc=<%IX> type=<%c> name=<%s>"
NVD1617I	"Index Cache entry <%ld> added at end after <%ld> for crc=<%IX> type=<%c> name=<%s>"
NVD1618I	"Index Cache entry <%ld> added at top before <%ld> for crc=<%IX>

Message Number	Log Message
	type=<%c> name=<%s>“
NVD1619I	“Index Cache entry <%ld> added for crc=<%IX> after <%ld> type=<%c> name=<%s>“
NVD1620I	“Index Caching children of <%s>“
NVD1621I	“Unable to Index Cache for <%s.%s.%s>“
NVD1622E	“ADD TO INDEX CACHE FAILED BECAUSE (FREE DIRECTORY CELL NOT FOUND): CRC=<%IX> TYPE=<%c> NAME=<%s>“
NVD1623I	“Adding instances of <%s.%s.%s> to Index Cache”
NVD1624W	“Unable to add entry to Index Cache (memory shortage) -- removing all entries in <%s.%s.%s>“
NVD1625I	“First entry in Index Cache not found for <%s>“
NVD1626I	“First entry <%d> found in Index Cache for <%s>, name=<%s>“
NVD1627I	“Next entry (to %ld) in Index Cache not found”
NVD1628I	“Next entry (to %ld) in Index Cache is <%ld> name=<%s>“
NVD1629E	“<%s.%s.%s.%s> NO LONGER FITS IN INDEX CACHE. PERFORMANCE MAY BE ADVERSELY AFFECTED.”
NVD1630I	“ICACHE: Using full walk algorithm for Class <%s.%s.%s>“
NVD1631I	“ICACHE: Using load by component prefix algorithm for Class <%s.%s.%s>“
NVD1632I	“ICACHE: Using chunky load non-component algorithm for Class <%s.%s.%s>“
NVD1633I	“Using prefix <%s> to search in class <%s.%s.%s>“
NVD1634I	“Loaded package <%ld> of <%ld>; running instance total is <%ld>“
NVD1635I	“Loaded <%lu> instances”
NVD1636I	“ICACHE: full walk algorithm loaded <%lu> instances”
NVD1637I	“Package <%s.%s.%s.%s> has OID <%12.12s>“
NVD1638E	“ICACHE LOAD ERROR: INVALIDATING CLASS <%s.%s.%s>“
NVD1639W	“ICACHE: Manager will shutdown. Reason: ICACHE_COUNT_ERROR = SHUTDOWN”
NVD1640W	“ICACHE: Manager will continue running. Reason: ICACHE_COUNT_ERROR = WARN”
NVD1641W	“ICACHE Load Error for Class <%s.%s.%s>, contents will be retained due to

Message Number	Log Message
	ICACHE_COUNT_ERROR setting”
NVD1650I	“Content Cache statistics for page segment <%lu>“
NVD1651I	“Type, State, Add YMDHMS, Hit YMDHMS, Hits, D.C.I”
NVD1652I	“%c, %c, %12.12s, %12.12s, %9lu, %s.%s.%s”
NVD1653I	“Page segment entries in use <%lu>, unused <%lu>, stale <%lu>, incomplete <%lu>, cache hits <%lu>“
NVD1654I	“Free area offset, size: bytes”
NVD1655I	“0x%08X, 0x%08X:%lu”
NVD1656I	“Page segment free areas <%lu>, total size <0x%08X, %lu> “
NVD1657I	“Content Cache status is %s, %s”
NVD1658I	“Total entries in use <%lu>, unused <%lu>, stale <%lu>, incomplete <%lu>, cache hits <%lu>“
NVD1659I	“Total free areas <%lu>, total available size <0x%08X:%lu>“
NVD1660E	“ERROR IN INITIALIZING CACHE FOR <%s>”
NVD1661I	“Success in Initializing Cache for <%s>”
1700-1799	Semaphores and IPC
NVD1701E	“ERROR, UNABLE TO CREATE <%s> EVENT SEMAPHORE, RC<%d>“
NVD1702E	“ERROR WAITING FOR <%s> EVENT SEMAPHORE <%d>, RC<%d>“
NVD1703E	“ERROR RESETTING <%s> EVENT SEMAPHORE <%d>, RC<%d>“
NVD1704E	“ERROR REMOVING <%s> EVENT SEMAPHORE <%d>, RC<%d>“
NVD1705E	“ERROR IN ZSYS_SETEVENTSEMAPHORE”
NVD1706E	“ERROR, COULD NOT INITIALIZE IPC”
NVD1707E	“ERROR, COULD NOT REINITIALIZE IPC”
NVD1708E	“ERROR EXCEPTION <%s> IN CLIENT AUTOMATION AGENT TASK AT <%X>“
NVD1709E	“ERROR, FUNCTION <%s> FAILED, RC<%d>“
NVD1710E	“SEMAPHORE IS NOT FOUND FOR EXISTING CLIENT AUTOMATION AGENT <%s>“
NVD1711E	“ERROR CREATING CACHING SEMAPHORE”

Message Number	Log Message
NVD1712E	"ERROR, UNABLE TO LOCK SYSTEM SEMAPHORE"
NVD1713E	"ERROR, UNABLE TO UNLOCK SYSTEM SEMAPHORE"
NVD1714E	"ERROR CREATING INDEX CACHING SEMAPHORE"
NVD1715E	"ERROR CREATING INDEX CACHE ENTRY SEMAPHORE"
NVD1716E	"ERROR CREATING INDEX CACHE AGING SEMAPHORE"
NVD1717E	"COULD NOT FIND ZCVT"
NVD1718E	"COULD NOT FIND ZTCBG"
1900-1999	ZPOOLS
NVD1967I	"%s"
NVD1972I	"%s"
NVD1982E	"%s"
NVD1983I	"%s"
NVD1984D	"%s"
2101-2201	Client Automation Agent Connection
NVD2101E	"ERROR BAD <%s> RECORD"
NVD2102E	"ERROR RETRIEVING RECORD BUFFER. RC<%d>"
NVD2103E	"ERROR, UNKNOWN <%s> RECORD"
NVD2104E	"ERROR <%s> RECORD UPDATE FAILURE"
NVD2105E	"ERROR SWABBING <%s> RECORD FROM NETWORK"
NVD2106E	"ERROR SWABBING <%s> RECORD TO NETWORK"
NVD2107E	"ERROR LOADING <%s> LIBRARY RC<%d>"
NVD2108E	"ERROR COMMUNICATING (ATBALLOC) - <%s>"
NVD2109E	"ERROR COMMUNICATING <%s>, ERROR MESSAGE <%s>"
NVD2110E	"ERROR SENDING TCP_MSG TO CLIENT AUTOMATION AGENT"
NVD2111E	"ERROR TERMINATING COMMUNICATION"
NVD2112I	"Terminating Connection"
NVD2113I	"EDMLOCTP retry interval is <%s> minutes"
NVD2114E	"REJECTING NON-CLIENT AUTOMATION DISTRIBUTED"

Message Number	Log Message
	CONFIGURATION SERVER CONNECTION BECAUSE DATABASE IS HARD LOCKED"
NVD2115I	"Connection Terminated"
NVD2116I	"Sending Final Ack"
NVD2117I	"Sent Final Ack"
NVD2118E	"ERROR ACK-ING FINAL ACK"
NVD2119I	"Sending EDMLOCTP response to Client Automation Agent"
NVD2120E	"ERROR SENDING EDMLOCTP RESPONSE TO CLIENT AUTOMATION AGENT"
NVD2121I	"Client Automation Agent Task Started, active tasks <%d>"
NVD2122E	"ERROR IN CLINIT, ERRMSG <%s>"
NVD2123I	"Host name for New Connection is <%s>"
NVD2124I	"Unable to get Host name for New Connection, Errmsg <%s>"
NVD2125E	"ERROR POSTING COMMUNICATION MANAGER"
NVD2126I	"Session Started"
NVD2127I	"Session Ended "
NVD2128I	"Client Automation Agent Task Ended, reason <%s>, active tasks <%d>"
NVD2129I	"Simulation Task Started, active tasks <%d>"
NVD2130I	"Simulation Task Ended, active tasks <%d> "
NVD2131E	"ERROR, INVALID LU TYPE OF <%c>"
NVD2132E	"ERROR, ADDRESS <%s> IN USE"
NVD2134I	"Terminating previous Connection PID<%d>"
NVD2135E	"ERROR, ADDRESS <%s> STILL IN USE"
NVD2136W	"Wrong TPN <%8.8s> received from LU <%17.17s> EDMTRANS expected."
NVD2137W	"Wrong TPN <%8.8s> received from LU <%17.17s> EDMLOCTP expected."
NVD2138I	"Setting DRAIN flag for the Client Automation Agent <%s>"
NVD2139E	"REJECTING INBOUND REQUEST FROM <%s>"
NVD2140I	"Conversation -- ID <%04X%04X> established"

Message Number	Log Message
NVD2141I	"Client Automation Agent INIT successful. Peer name is <%s>"
NVD2142I	"Expected TPN <%8.8s> received from LU <%17.17s>"
NVD2143I	"Task detected a short on storage condition - Client Automation Agent will be asked to try later"
NVD2144E	"ERROR <%d> IN PUTREC ON TEMPLATE"
NVD2145E	"ERROR <%d> IN PUTREC ON HEAP"
NVD2146E	"ERROR <%d> IN PUTREC ON EOB"
NVD2147E	"ERROR <%d> IN PUTREC ON FLUSH"
NVD2148E	"ERROR <%d> IN PUTREC ON DATA"
NVD2149E	"ERROR <%d> IN GETREC"
NVD2150W	"Session Loss has been detected for <%s>, Task Ending"
NVD2151W	"Wrong TPN <%8.8s> received on LU <%17.17s> EDMLOSLU expected."
NVD2152I	"Sending zero bytes across"
NVD2153E	"INVALID MESSAGE START CHARACTER IN <%s>"
NVD2154E	"INVALID MESSAGE END CHARACTER IN <%s>"
NVD2155E	"INVALID MESSAGE IN <%s>"
NVD2156I	"%s TH: TLEN=%4X TCODE=%4X OAT=%4X DAT=%4X"
NVD2157I	"%s BH: BLEN=%4X BCODE=%4X BOBJ=%8s FLAG1=%2X FLAG2=%2X FLAG3=%2X FLAG4=%2X"
NVD2158I	"%s RH: RLEN=%4X RCODE=%4X RCLN=%4X FLAG1=%2X FLAG2=%2X TYPE=%c COMPRESS=%1x SPANREC=%1x"
NVD2159E	"ERROR IN ZTPRECV"
NVD2160I	"rec @ is 0"
NVD2161E	"DECOMPRESSED RECORD LENGTH <%d> GREATER THAN MAX ALLOWED <%d>"
NVD2162E	"RECORD DECOMPRESSION BUFFER CANNOT BE EXPANDED FROM <%d> TO <%d> BYTES"
NVD2163E	"SPANNED RECORD LENGTH <%d> GREATER THAN MAX ALLOWED <%d>"
NVD2164E	"SPANNED RECORD BUFFER CANNOT BE EXPANDED FROM <%d> TO <%d> BYTES"

Message Number	Log Message
NVD2165I	"Spanned record processing"
NVD2166I	"Last record in block"
NVD2167E	"RECORD LEN TOO SMALL"
NVD2168I	"End of spanned record, length=<%d>"
NVD2169E	"ERROR, ZERO LENGTH RECORD"
NVD2170E	"ERROR <%d> IN PROCESSING SPANNING RECORD"
NVD2171E	"DECOMPRESS ERROR UNCOMPRESSED LENGTH=<%d>; EXPECTED LENGTH=<%d>"
NVD2172I	"Decompressed Length <%d>"
NVD2173I	"Returned Length <%d>"
NVD2174E	"UNEXPECTED ERROR, RECORD LENGTH <%d> < RECORD HEADER SIZE <%d>"
NVD2175I	"Record Mismatch: record expected is <%s>"
NVD2176I	"End-of-Block"
NVD2177I	"Compression from <%d> to <%d>"
NVD2178I	"No compression using original length of <%d>"
NVD2179E	"SPANNED RECORD ERROR"
NVD2180I	"AUTOFLUSHING Send Buffer"
NVD2181E	"ERROR IN ZTPSEND"
NVD2182E	"ERROR IN AUTO-FLUSH RESPONSE ERROR"
NVD2183I	"putrec ADDR <%08x> LEN <%d> SEQNO <%d> TYPE <%d>"
NVD2184I	"Template copied to <%08X>, length <%d>"
NVD2185I	"Flushing Send Buffer"
NVD2186E	"UNDEFINED PUTREC REQUEST TYPE"
NVD2187I	"putrec(%08x,%d,%d,%s,%d,%08x) complete"
NVD2188I	"putrec TH=<%d> BH=<%d> RH=<%d> BLKS=<%d> PUTS=<%d>"
NVD2189I	"Freeing saved template @ <%08X>, length <%d>"
NVD2190E	"ERROR <%s> REQUEST TYPE <%d>, RC <%d>, Msg <%s>, SessionAddress <%p>, Result <%d>"

Message Number	Log Message
NVD2191E	"ERROR <%s> RC <%d>, Msg <%s>, SessionAddress <%p>, Result <%d>"
NVD2192I	"%s RC <%d>, Msg <%s>, SessionAddress <%p>, Result <%d>"
NVD2193E	"PUTREC-FORMAT ERROR RC <%d>"
NVD2194E	"PUTREC-FLUSH ERROR RC <%d>"
NVD2195E	"PUTREC-DATA ERROR RC <%d>"
NVD2196I	"ZTCPMGR: Simulation discontinued due to short on storage condition"
NVD2197I	"ZTCPMGR: Simulation discontinued due to task attach error"
NVD2198I	"Client Automation Agent Disconnected"
NVD2199I	"Communications Timed Out"
NVD2200I	"Communications error rc=<%d - %s>"
2201-2207	Pipe
NVD2201E	"ERROR, UNABLE TO OPEN PIPE"
NVD2202E	"ERROR, UNABLE TO READ FROM PIPE"
NVD2203E	"ERROR, UNABLE TO WRITE TO PIPE"
NVD2204E	"ERROR, UNABLE TO ATTACH TASK"
NVD2205E	"TASK ABENDED WHILE PROCESSING OBJECT <%s>"
NVD2206I	"Transferred record of size <%d> bytes over pipe"
NVD2207E	"ERROR SENDING RECORD OF SIZE <%d> BYTES OVER PIPE"
2208-2249	Compress
NVD2208E	"COMPRESS: INVALID INPUT ADDRESS <%08x> OR LENGTH <%d>"
NVD2209E	"COMPRESS: INVALID OUTPUT ADDRESS <%08x> OR LENGTH <%d>"
NVD2210E	"COMPRESS: OUTPUT BUFFER <%d> SMALLER THAN INPUT BUFFER <%d>"
NVD2211I	"COMPRESS: Input buffer @ <%08x> length <%d>"
NVD2212I	"COMPRESS: ran out of room"
NVD2213I	"COMPRESS: Output <%d> larger than input <%d>"
NVD2214I	"COMPRESS: Output buffer @ <%08x> length <%d>"
NVD2215E	"DECOMPRESS: INVALID INPUT ADDRESS <%08x> OR LENGTH <%d>"

Message Number	Log Message
NVD2216E	"DECOMPRESS: INVALID OUTPUT ADDRESS <%08x> OR LENGTH <%d>"
NVD2217E	"DECOMPRESS: OUTPUT BUFFER LENGTH <%d> SMALLER THAN INPUT BUFFER LENGTH <%d>"
NVD2218I	"DECOMPRESS: Input buffer @ <%08x> length <%d>"
NVD2219E	"DECOMPRESS: OUTPUT BUFFER @ <%08x> LENGTH <%d> TOO SMALL"
NVD2220E	"DECOMPRESS: OUTPUT BUFFER LENGTH EXCEEDED OUTLEN=<%d> INLEN=<%d> IN_OFFSTE=<%d> OUT_OFFSET=<%d> EXP_SIZE=<%d>"
NVD2221I	"DECOMPRESS: Output buffer @ <%08x> length <%d>"
NVD2222I	"%s *%. *s"
2250-2299	FTP
NVD2250E	"REQUESTED OBJECT ID BUFFER IS <%d>"
NVD2251E	"SEND FILES FOR HEAP <%d>"
NVD2252E	"MISSING OBJECT ID LIST FOR HEAP <%d>"
NVD2253I	"File Transfer Complete"
2300-2399	Miscellaneous
NVD2300I	"EOB not received"
NVD2301E	"ERROR FORCING CLIENT AUTOMATION AGENTS TO END!"
NVD2302E	"ERROR ALLOCATING TCLBLK"
NVD2303E	"ERROR IN TASK TCL INITIALIZATION"
2600-2699	File Input/Output
NVD2601E	"ERROR OPENING FILE <%s>"
NVD2602E	"ERROR CLOSING FILE <%s>"
NVD2603E	"ERROR READING FILE <%s>"
NVD2604E	"ERROR WRITING FILE <%s>"
NVD2605E	"ERROR, ZERO-LENGTH RECEIVED BY GETREC(T_DATA) FOR FILE <%s>"
NVD2606I	"Flushing Data"
NVD2607E	"ERROR, UNABLE TO OPEN CONTROL FILE"

Message Number	Log Message
NVD2608E	"UNABLE TO SET CONTROL FILE POINTER FOR FILE CONTROL UPDATE"
NVD2609E	"UNABLE TO READ CONTROL FILE FOR FILE CONTROL UPDATE"
NVD2610E	"UNABLE TO UPDATE FILE CONTROL INFORMATION"
NVD2611E	"UNABLE TO SET CONTROL FILE POINTER"
NVD2612E	"UNABLE TO READ CONTROL RECORD"
NVD2613E	"UNABLE TO UPDATE CONTROL RECORD"
NVD2614E	"UNABLE TO WRITE CONTROL ENTRY"
NVD2615E	"ERROR, CONTROL FILE NOT FOUND <%s>"
NVD2616I	"READFILE rc <%d>"
NVD2617E	"ERROR, UNABLE TO CREATE FILE <%s>"
NVD2618I	"Writing %d Bytes to File <%s>"
NVD2619E	"ERROR WRITING %d BYTES TO FILE <%s>"
NVD2620I	"Bytes Written <%d>"
NVD2621I	"File Read <%s>"
NVD2622I	"File Size <%d>"
NVD2623E	"ERROR, UNABLE TO READ ENTIRE FILE <%s> SIZE <%d> BYTES READ <%d>"
NVD2624E	"ERROR READING CONTROL FILE: #RECS = <%d>, REC# = <%d>"
NVD2625E	"ERROR UPDATING CONTROL FILE: FILE <%s> DOMAIN <%s> CLASS <%s> INSTANCE <%s>"
NVD2626E	"ERROR DELETING FILE <%s>"
NVD2627E	"ERROR IN PUTREC ON <%s> RC <%d>"
NVD2628E	"ERROR SENDING AUXILIARY DATA <%d>"
NVD2629I	"Opening <%s>"
NVD2630E	"ERROR OPENING <%s>, ERRNO <%s> "
NVD2631E	"FSTAT : <%s>"
NVD2632W	"File sizes differ (Actual) <%d> : (Req.) <%d>"
NVD2633I	"Bytes Read <%d>"

Message Number	Log Message
NVD2634E	"ERROR READING CLASS <%s.%s.%s>, RC<%d>"
NVD2635E	"ERROR IN PUTREC FOR CLASS <%s.%s.%s>, RC<%d>"
NVD2636E	"ERROR IN PUTREC OF INSTANCE <%s.%s.%s.%s>, RC<%d>"
NVD2637E	"ERROR IN PUTREC OF OBJECT <%s.%s.%s.%s>, SEQNO<%d>, RC<%d>"
NVD2638I	"Sending DELTA <%s>"
NVD2639E	"ERROR <RC=%d> OPENING PATCH <%s>"
NVD2640E	"ERROR IN PUTREC OF DELTA FILE <%s>, SEQNO<%d>, RC<%d>"
NVD2641E	"ERROR IN PUTREC ON OBJECT IDs <%d>"
NVD2642E	"ERROR OPENING INSTANCE <%s.%s.%s.%s>"
NVD2643I	"Resource transfer for <%s.%s.%s.%s> begins, <%lu> bytes starting at offset <%lu>"
NVD2644I	"Resource transfer for <%s.%s.%s.%s> in progress, <%lu> of <%lu> bytes sent in <%1.1f> seconds"
NVD2645I	"Resource transfer for <%s.%s.%s.%s> complete, <%lu> bytes sent in <%1.1f> seconds"
NVD2646I	"File transfer for <%s> begins, <%lu> bytes starting at offset <%lu>"
NVD2647I	"File transfer for <%s> in progress, <%lu> of <%lu> bytes sent in <%1.1f> seconds"
NVD2648I	"File transfer for <%s> complete, <%lu> bytes sent in <%1.1f> seconds"
NVD2649E	"ERROR UNABLE TO FIND DELTA OBJECT"
NVD2650E	"ERROR CREATING FILE FROM RESOURCE <%s.%s.%s.%s>"
NVD2651I	"Deleting Patch file <%s> ResourceSize <%d> PatchSize <%d>"
NVD2652E	"ERROR CREATING PATCH FILE FROM <%s> TO <%s> BECAUSE OF <%s>"
NVD2653E	"ERROR EXPLODING THE FILE <%s> BECAUSE OF <%s>"
NVD2654E	"ERROR OPENING TEMPORARY DELTA FILE <%s>"
NVD2655E	"ERROR CREATING RESOURCE <%s.%s.%s.%s>"
NVD2656E	"ERROR INVALID DATABASE PATH, <%s>"
NVD2657E	"ERROR CHECK DATABASE PATH <DBPATH> IN THE CLIENT AUTOMATION CONFIGURATION SERVER PROFILE FILE"

Message Number	Log Message
NVD2658E	"ERROR INVALID DMA DATABASE PATH, <%s>"
NVD2659E	"ERROR CHECK DMA DATABASE PATH <DMA_STAGE_PATH> IN THE CLIENT AUTOMATION CONFIGURATION SERVER PROFILE FILE"
NVD2660E	"ERROR WHILE OPENING TRACE FILE: <%s>"
NVD2661E	"UNKNOWN RECORD TYPE <%s>"
NVD2662E	"ERROR OPENING QUEUE FILE <%s>"
NVD2663E	"ERROR READING LIST FILENAME FROM QUEUE FILE <%s>"
3000-3100	Object Resolution
NVD3001E	"ERROR, OBJECT PROCESSING FOR <%s.%s> FAILED RC=<%d>"
NVD3002E	"ERROR, OBJECT RESOLUTION ABORTED - RESOLUTION LIMIT EXCEEDED"
NVD3003E	"ERROR, PROCESSING ABORTED"
NVD3004E	"ERROR, SUBSTITUTION FAILED"
NVD3005E	"ERROR, MASTER PROCESSING ABORTED"
NVD3006I	"Unable to resolve Object <%s>, rc=<%d>"
NVD3007E	"ERROR, STOPPING CURRENT PATH OF OBJECT RESOLUTION"
NVD3008E	"ERROR, INVALID ENTRY TYPE <%s>"
NVD3009I	"Resolving <%s.%s.%s.%s>"
NVD3010I	"Object Resolved <%s>"
NVD3011I	"'%8.8s' ZPCB at <%p> directory at <%p>"
NVD3012I	"'%8.8s' ZPCB at <%p> directory at <%p>"
NVD3013W	"Object ID list not found for <%s>"
NVD3014I	"Requested Obj. <%s> Buflen <%d>"
NVD3015I	"Taking type <%c> connection: <%s.%s.%s(%s)> to class existing in resolution tree"
NVD3016I	"%s<%s> depth: <%i>"
NVD3017E	"ERROR, FAILED FINDING FIRST DOMAIN IN <%s>"
NVD3018E	"ERROR, FAILED FINDING FIRST CLASS IN <%s.%s>"
NVD3019E	"ERROR, FAILED FINDING FIRST INSTANCE IN <%s.%s>"

Message Number	Log Message
NVD3020E	"ERROR, FAILED FINDING NEXT DOMAIN IN <%s>"
NVD3021E	"ERROR, FAILED FINDING NEXT CLASS IN <%s.%s>"
NVD3022E	"ERROR, FAILED FINDING NEXT INSTANCE IN <%s.%s.%s>"
NVD3023E	"ERROR, NO INSTANCES FOUND IN INDEX CACHE FOR CLASS <%s.%s.%s>"
NVD3024E	"ERROR, GETTING INSTANCE -- REMOVING ALL ENTRIES IN <%s.%s.%s>"
NVD3025I	"GET_OBJECT"
NVD3026E	"ERROR, GETTING TEMPLATE FROM GETREC, RC = <%d>"
NVD3027E	"ERROR, GETTING TEMPLATE BUFFER"
NVD3028E	"ERROR, NO RECORDS IN OBJECT"
NVD3029E	"ERROR, UNABLE TO RESOLVE OBJECT, RC = <%d>"
NVD3030E	"ERROR, UNABLE TO STORE DATA RECORD, RC = <%d>"
NVD3031E	"ERROR, UNABLE TO RESOLVE OBJECT, RC = <%d>"
NVD3032E	"ERROR, UNKNOWN STATUS FROM GETREC, RC = <%d>"
NVD3033W	"Single Service Instance Name not specified"
NVD3034I	"Single Service Instance Name should be specified in ZOBJSSIN variable"
NVD3035W	"Single Service Class Name not specified; defaulting to ZSERVICE"
NVD3036W	"Single Service Domain Name not specified; defaulting to SYSTEMX"
NVD3037E	"UNABLE TO PROCESS OBJECT, RC = <%d>"
NVD3038E	"BAD STATUS FROM zrodo(), RC = <%d>, OBJECT = <%s>"
NVD3039E	"ERROR, UNKNOWN RECORD TYPE, RC = <%d>"
NVD3040E	"ERROR, OBJECT NOT FOUND <%8.8s>"
NVD3041I	"Bandwidth Reset to Unlimited"
NVD3042I	"Bandwidth Max=<%lu>, Percent=<%u>, Effective=<%lu>"
NVD3043E	"OBJECT RESOLUTION ABORTED - DUE TO RETURN CODE, RC = <%d>"
NVD3044I	"Resolving <%s>"
NVD3045I	"Message Changes %s %s <%s> <%8.8s>"

Message Number	Log Message
NVD3046E	"RECURSIVE TREE RESOLUTION ABORTED"
NVD3047I	"RESOLUTION Begins: <%s.%s (%s)>"
NVD3048E	"RESOLUTION LIMIT <%d> CURRENT RESOLUTION COUNT <%d>"
NVD3049I	"RESOLUTION Ends: <%s.%s> CRC: <%s>"
NVD3050I	"Total time for MASTER resolution is <%d> seconds"
NVD3051I	"Total number of objects resolved <%d>"
NVD3052I	"Resolution cannot cache objects created by <%s.%s>"
NVD3053I	"Resolution can cache objects created by <%s.%s>"
NVD3054E	"%8.8s FIELD LENGTH <%d> EXCEEDS MAX FIELD SIZE(%d) - FIELD IGNORED"
NVD3055I	"Ignoring Expressions"
NVD3056I	"EXPRESSION STARTED:"
NVD3057I	"Resolution cache encountered expression for class <%s>, <%8.8s> = <%0.*s>"
NVD3058I	"Resetting Object Resolution STOP Flag"
NVD3059I	"Resolution cache encountered method for class <%s>, <%8.8s> = <%0.*s>"
NVD3060E	"ERROR DEFINING VARIABLE -- ABORTING RESOLUTION"
NVD3061W	"Ignoring duplicate instance <%-s.%-s.%-s> OBJECTID <%s>"
NVD3062E	"MISSING OR BLANK CLASS NAME IN <%-s>"
NVD3063E	"PARENT F.D.C IS <%-s.%-s.%-s>"
NVD3064E	"MISSING OR BLANK INSTANCE NAME IN <%-s>"
NVD3065I	"Resolution message changed <%-s.%-s (%s)>"
NVD3066I	"DATE/TIME PARENT: <%s %s>"
NVD3067I	"DATE/TIME CHILD: <%s %s>"
NVD3068E	"PARENTS ZOCC NOT FOUND"
NVD3069I	"DATE/TIME PERCOLATION: PARENT: <%s> CHILD: <%s>"
NVD3070I	"Migrate <%8.8s> -> <%8.8s>"
NVD3071I	"Resolved Connects: _ALWAYS_ (%d)"

Message Number	Log Message
NVD3072I	"Executed Methods: _ALWAYS_ (%d)"
NVD3073I	"Resolved Connects: _ALWAYS_ (%d), %s (%d)"
NVD3074I	"Executed Methods: _ALWAYS_ (%d), %s (%d)"
NVD3075I	"UNKNOWN Processing Begins"
NVD3076E	"ERROR SETTING OBJECT QUEUE <%s>"
NVD3077E	"ERROR RETRIEVING HEAP # %d FOR OBJECT <%s>"
NVD3078E	"ERROR RETRIEVING OBJECT <%s>"
NVD3079E	"MISSING INSTANCE LIST FOR CLASS <%s>"
NVD3080E	"SINGLE SERVICE RESOLUTION SKIPPING <%s.%s.%s.%s>"
NVD3081I	"Processing Requested Service <%s.%s.%s.%s>"
NVD3082I	"ZBASE resolution begins: %s.%s <%s>"
NVD3083E	"ERROR, MASTER RESOLUTION ABORTED"
NVD3084E	"ERROR, UNABLE TO RESOLVE OBJECT RC = <%d>, NAME: <%s>, "
NVD3085E	"TREE RESOLUTION CALLED RECURSIVELY"
NVD3086E	"INVALID HANDLE SPECIFIED <%s>"
NVD3087E	"ERROR CREATING TREEVAR OBJECT"
NVD3088E	"ERROR RESOLVING <%s>"
NVD3089E	"TREE NOT INITIALIZED"
NVD3090E	"HANDLE NOT FOUND OR INVALID"
NVD3091E	"HANDLE MISMATCH ZTCBG <%s> ZTREHAND <%s>"
NVD3092E	"INVALID TYPE"
NVD3093I	"Expression (%s) Len <%d> "
NVD3094I	"Result (%s) Len <%d>"
NVD3095E	"ERROR DURING OBJECT RESOLUTION, RC = <%d>, OBJECT NAME: <%s>"
NVD3096I	"Object <%s> expanded. Old heap size used/allocated: <size_old_used/size_old>, new heap size used/allocated: <size_new_used/size_new>, number of heaps: <heaps>."
NVD3097I	"Persistent Systems object <%s> created, heap size used/allocated: <size_

Message Number	Log Message
	used/size_allocated>, number of heaps: <number_of_heaps>.”
NVD3098I	“Object <%s>, heap size: <size>, heap index: expanded. Old index size: <size_old>, new index size: <size_new>, max heap: <index_slots>.”
NVD3099I	“Removing object <%s>: heap size used/allocated: <size_used/size_allocated>, number of heaps used/allocated: <number_of_used_heaps/number_of_allocated_heaps>.”
NVD3100E	“RESOLUTION ABORTED: <%s> HAS BEEN ALREADY RESOLVED IN THE RESOLUTION LEG”
3101-3199	User ID Errors
NVD3101E	“USERID NOT FOUND IN INBOUND OBJECT”
NVD3102E	“USERID NOT IN PROFILE ADMIN_LIST”
NVD3103E	“ERROR IN VARPUT OF ZMTHPRMS, UNABLE TO VERIFY USERID/PASSWORD”
NVD3104E	“USERID/PASSWORD VERIFICATION ERROR.”
NVD3105E	“ERROR RETRIEVING USERID”
3200-3299	Object Resolution
NVD3200I	“Persistent Systems object <%s> will be treated as transient with PCB name <%s>”
NVD3201E	“RESOLUTION ABORTED: TASK_STACK_SIZE IS ABOUT TO OVERFLOW, USED <%d> of <%d>, DEPTH <%d>, CHECK FOR CIRCULAR REFERENCES”
NVD3202I	“<%s.%s.%s> for %s agent was filtered out by <%s.%s = %s> filter”
NVD3203I	“<%s.%s.%s> for %s agent was allowed through <%s.%s = %s> filter”
NVD3204E	“ERROR: ZO CB %08X HAS ZERO ->PCB; IGNORED FOR TREE RES; proces_vars=%d”
NVD3205E	“ERROR: ZO CB %08X HAS ZERO ->PCB; FAILS NON-TREE RES”
NVD3206I	“Resolution cache encountered substitution for class <%s>, variable <%8.8s> = <%s>”
NVD3207I	“Variable <%0.8s> deleted from object <%0.8s>”
NVD3208I	“All variables deleted from object <%0.8s>, directory @ <%08x>, size <%ld>, reset”
NVD3210W	“Resolving <%s.%s.%s.%s>, Cached Template Size Mismatch, Expected Size <%lu>, Actual Size <%lu>”

Message Number	Log Message
NVD3211W	"Refreshing Cache for Class <%s.%s.%s.%s>"
NVD3212W	"Resolving <%s.%s.%s.%s>, Cached Instance Size Mismatch, Expected Size <%lu>, Actual Size <%lu>"
NVD3213W	"Refreshing Cache for Instance <%s.%s.%s.%s>"
NVD3214E	"RESOLVING <%s.%s.%s.%s>, DATABASE TEMPLATE SIZE MISMATCH, EXPECTED SIZE <%lu>, ACTUAL SIZE <%lu>"
NVD3215E	"RESOLVING <%s.%s.%s.%s>, DATABASE INSTANCE SIZE MISMATCH, EXPECTED SIZE <%lu>, ACTUAL SIZE <%lu>"
3300-3399	Tree Resolution
NVD3300E	"TREE RESOLUTION CALLED RECURSIVELY"
NVD3301I	"Make Tree starting"
NVD3302E	"ERROR CREATING TREEVAR OBJECT"
NVD3303E	"TREE NOT INITIALIZED"
3500-3599	Metaclass Resolution
NVD3501E	"ERROR, METACLASS PROCESSING STOPPED"
NVD3502E	"ERROR, METACLASS <%s> PROCESSING FAILED RC=<%d>"
NVD3503E	"ERROR, METACLASS PROCESSING ABORTED"
NVD3520E	"ERROR CREATING DOMAINLIST FOR CLASS <%s>"
NVD3521E	"ERROR, DOMAIN <%s> NOT FOUND IN DOMAINLIST"
NVD3522E	"ERROR, DOMAIN <%s> COULD NOT BE ADDED TO DOMAINLIST FOR CLASS <%s>"
NVD3523I	"MetaClass Resolution not required for <%s.%s.%s>"
NVD3524I	"MetaClass not found to reset priorities"
NVD3525I	"MetaClass Set Info For <%s>"
NVD3526I	"MetaClass Resolution for <%s>"
3600-3699	Class Resolution
NVD3600I	"Class not processed - length is zero <%8.8s>"
NVD3601I	"Connection <%-s> ignored - floor class <%8.8s> reached"
NVD3602I	"Class not processed - blank field <%8.8s>"

Message Number	Log Message
NVD3603I	"Class not processed - _NULL_ field <%8.8s>"
NVD3604I	"Message changed back from <%8.8s> <%8.8s>"
NVD3605I	"Dormant connection <%8.8s %8.8s>"
NVD3606I	"Remote class field <%8.8s>"
NVD3607I	"Class field <%8.8s> not selected - generic MSG <%8.8s>"
NVD3608I	"Message changed from <%8.8s> to <%8.8s>"
NVD3609I	"Class field <%8.8s> selected - generic MSG <%8.8s>"
NVD3610I	"CLASS DMN <%s> CLS <%s>"
NVD3611E	"CLASS DOES NOT EXIST"
NVD3612I	"<%s>.<%s>.<%s>"
NVD3613I	"Processing connection <%-s> for floor class <%8.8s>"
NVD3614E	"ERROR READING CLASS"
4000-4099	Security
NVD4001W	"Client Automation Configuration Server License expired on <%s>"
NVD4002W	"Client Automation Configuration Server License will expire on <%s>"
NVD4003W	"Client Automation Configuration Server Not Running on a Licensed Platform"
NVD4004I	"Client Automation Agent Information <%s>"
NVD4005E	"REJECTING CONNECTION, LICENSE LIMIT <%d> EXCEEDED"
NVD4006E	"REJECTING CONNECTION, NOT A LICENSED PLATFORM"
NVD4007I	"Client Automation Agent is Year2000 Compliant"
NVD4008I	"Client Automation Agent is NOT Year2000 Compliant"
NVD4009I	"Client Automation Agent is Non Unicode"
NVD4010E	"REJECTING CLIENT AUTOMATION AGENT BECAUSE IT IS NOT YEAR2000 AWARE"
NVD4011I	"%s Task Limit Exceeded -- sending <retry in %d min> response"
NVD4012I	"Logons are disabled -- sending <retry in %d min> response"
NVD4013I	"Hard Task Limit Exceeded -- closing connection"
NVD4014I	"Logon disabled -- closing connection"

Message Number	Log Message
NVD4015I	"Short On Storage -- sending <retry in %d min> response"
NVD4016I	"Client Automation Distributed Configuration Server security method was not specified in edmprof. Security check disabled"
NVD4017I	"Soft Task Limit Exceeded -- closing connection"
NVD4018I	"Short on Storage condition reached -- closing connection"
NVD4019I	"Shutdown has been detected -- closing connection"
NVD4020I	"Soft Task Limit Exceeded and LIMIT=CLOSE set -- closing connection"
NVD4021I	"Short on Storage condition reached and LIMIT=CLOSE set -- closing connection"
NVD4022E	"INBOUND OBJECT <%s>, NOT IDENTIFY. CONNECTION TERMINATED"
NVD4024E	"CONNECTION REJECTED: PRODUCT <%s> IS NOT LICENSED"
NVD4025E	"CONNECTION OF <%s> REJECTED: EXCEEDED LICENSE LIMIT"
NVD4026I	"License is taken from <%s> "
NVD4027I	"License1 string found <%s> "
NVD4028I	"License2 string found <%s> "
4100-4199	License and Profile
NVD4100E	"LICENSE VIOLATION -- 90 DAYS GRACE PERIOD HAS EXPIRED"
NVD4101E	"LICENSE VIOLATION -- CLIENT AUTOMATION CONFIGURATION SERVER WILL SHUTDOWN IN <%d> DAYS, ON <%i/%i/%i>"
NVD4102E	"LICENSE VIOLATION -- CLIENT AUTOMATION CONFIGURATION SERVER WILL SHUTDOWN IN <90> DAYS, ON <%i/%i/%i>"
NVD4103W	"<%s> file is not found, unable to verify product consistency"
NVD4104W	"[RCS_VERSION] section not found or empty in <%s> file. No version verification will be performed"
NVD4105I	"version.nvd: <%s>"
NVD4106I	"Verifying product consistency according to <%s> file"
NVD4107W	"Timestamp for <%s> is <%4.4d/%2.2d/%2.2d %2.2d:%2.2d:%2.2d> and newer than required <%4.4d/%2.2d/%2.2d %2.2d:%2.2d:%2.2d>"
NVD4108E	"TIMESTAMP FOR <%s> IS <%4.4d/%2.2d/%2.2d %2.2d:%2.2d:%2.2d> AND OLDER THAN REQUIRED <%4.4d/%2.2d/%2.2d %2.2d:%2.2d:%2.2d>"

Message Number	Log Message
NVD4109W	"File size for <%s> is <%d> and does not match required <%d>"
NVD4110W	"MD5 for <%s> is <%s> and does not match required <%s>"
NVD4111E	"REQUIRED FILE <%s> NOT FOUND IN <%s>. FILE VERSION VERIFICATION FAILED."
NVD4112I	"Identified License for SSL"
NVD4113I	"Identified License for test environment"
NVD4114W	"Datestamp for <%s> is <%4.4d/%2.2d/%2.2d> and NEWER than required <%4.4d/%2.2d/%2.2d>"
NVD4115E	"DATESTAMP FOR <%s> IS <%4.4d/%2.2d/%2.2d> AND OLDER THAN REQUIRED <%4.4d/%2.2d/%2.2d>"
4200-4299	Database and Y2K
NVD4200I	"Starting database verification"
NVD4201I	"Database verification completed successfully"
NVD4202I	"Database verification failed"
NVD4203I	"Verification: Processing File<%s>"
NVD4204E	"Y2K VERIFICATION FAILED FOR <%s>.<%s>"
NVD4205I	"Verification: Processing Domain <%s>.<%s>"
NVD4206E	"Y2K VERIFICATION FAILED FOR DOMAIN <%s>.<%s>"
NVD4207I	"Verification: Processing class <%s>.<%s>.<%s>"
NVD4208E	"Y2K VERIFICATION FAILED FOR CLASS <%s>.<%s>.<%s>"
NVD4209E	"Y2K VERIFICATION FAILED FOR CLASS CONTROL INFO FOR <%s>.<%s>.<%s>"
NVD4210E	"Y2K VERIFICATION FAILED FOR INSTANCE <%s>.<%s>.<%s>.<%s>"
NVD4211E	"Y2K VERIFICATION FAILED FOR CLASS <%s>.<%s>.<%s>; FIELD = LastUpdatedDate"
NVD4212E	"Y2K VERIFICATION FAILED FOR CLASS <%s>.<%s>.<%s>; FIELD = LastSyncDate"
NVD4213E	"Y2K VERIFICATION FAILED FOR CLASS <%s>.<%s>.<%s>; FIELD = LastCreateDate"
NVD4214E	"Y2K VERIFICATION FAILED FOR CLASS <%s>.<%s>.<%s>; FIELD = LastClassUpdateDate"

Message Number	Log Message
NVD4215E	"Y2K VERIFICATION FAILED FOR CLASS <%s>.<%s>.<%s>; FIELD = LastInstanceUpdateDate"
NVD4216E	"Y2K VERIFICATION FAILED FOR CLASS <%s>.<%s>.<%s>; FIELD = LastDataUpdateDate"
NVD4217I	"Verify Instance <%s.%s.%s.%s> Size changed to match class template"
NVD4218E	"VERIFY INSTANCE <%s.%s.%s.%s> ERROR UPDATING INSTANCE, AUTOFIX FAILED"
NVD4219W	"Verify Instance <%s.%s.%s.%s> Autofix will not change instance size"
NVD4220I	"Verify Instance <%s.%s.%s.%s> Dates updated by autofix"
NVD4221E	"VERIFY INSTANCE <%s.%s.%s.%s> ERROR UPDATING INSTANCE, AUTOFIX FAILED"
NVD4222W	"Class <%s.%s.%s> ZRSCDATE not found but resource flag set in control info"
NVD4223W	"Verify Class <%s.%s.%s> cannot read DMA class control information"
NVD4224E	"CLASS <%s.%s.%s> CLASS PREFIX SIZE LESS THAN EXPECTED"
NVD4225I	"Actual class size <%d>, expected size <%d>"
NVD4226W	"Class <%s.%s.%s> prefix contains invalid dates"
NVD4227W	"To correct these dates, restart Client Automation Configuration Server with MGR_DB_VERIFY.DB_AUTOFIX set to YES"
NVD4228E	"VERIFY CLASS <%s.%s.%s> ERROR VALIDATING CLASS PREFIX DATES"
NVD4229E	"CLASS <%s.%s.%s> CLASS PREFIX SIZE <%d> LESS THAN EXPECTED <%d>"
NVD4230E	"DATABASE VERSION MISMATCH FOR CLASS <%s.%s.%s>"
NVD4231E	"CLASS <%s.%s.%s> TEMPLATE FIRST ENTRY NOT FOUND"
NVD4232E	"CLASS <%s.%s.%s> TEMPLATE FIRST ENTRY NAME <%s> DOES NOT MATCH CLASS NAME"
NVD4233E	"CLASS <%s.%s.%s> TEMPLATE SIZE MISMATCH"
NVD4234E	"EXPECTED TEMPLATE SIZE <%d>, ACTUAL SIZE <%d>"
NVD4235E	"VERIFY CLASS <%s.%s.%s> ERROR VALIDATING CLASS TEMPLATE"
NVD4236E	"DATABASE CLASS <%s.%s.%s> MAY BE CORRUPT"
NVD4237I	"Verifying Class <%s.%s.%s>"

Message Number	Log Message
NVD4238E	"CLASS <%s.%s.%s> CONTAINS INVALID INSTANCE NAME (SEE STDERR)"
NVD4239E	"VERIFYING CLASS <%s.%s.%s> TERMINATED BY FindNextInstance(...) RC=%d"
NVD4240E	"VERIFY CLASS <%s.%s.%s> INSTANCE FOUND <%ld>, VERIFIED <%ld>, ERRORS <%ld>"
NVD4241I	"Verifying Domain <%s.%s>"
NVD4242E	"DOMAIN <%s.%s> CONTAINS INVALID CLASS NAME (SEE STDERR)"
NVD4243E	"VERIFY DOMAIN <%s.%s> TERMINATED BY FindNextClass(...) RC=<%d>"
NVD4244E	"VERIFY CLASS <%s.%s.%s> INSTANCE COUNT IS ZERO"
NVD4245E	"VERIFY CLASS <%s.%s.%s> AUTOFIX UNABLE TO DELETE INVALID CLASS"
NVD4246E	"VERIFY CLASS <%s.%s.%s> AUTOFIX DELETED INVALID CLASS"
NVD4247E	"VERIFY CLASS <%s.%s.%s> CANNOT VALIDATE DMA INSTANCE COUNT"
NVD4248E	"VERIFY CLASS <%s.%s.%s> DMA INSTANCE COUNT DOES NOT EQUALACTUAL INSTANCE COUNT"
NVD4249E	"VERIFY CLASS <%s.%s.%s> DMA TotalInstanceCount IS <%ld>, ACTUAL INSTANCE COUNT IS <%ld>"
NVD4250E	"VERIFY DOMAIN <%s.%s> CLASSES FOUND <%ld>, VERIFIED <%ld>, ERRORS <%ld>"
NVD4251E	"VERIFY DOMAIN <%s.%s> ERROR READING ZBASE CLASS"
NVD4252E	"VERIFY DOMAIN <%s.%s> ERROR READING VALIDATING CLASS <%s>"
NVD4253E	"DATABASE CLASS <%s.%s.%s> MAY BE CORRUPT"
NVD4254E	"DOMAIN <%s.%s> CLIENT AUTOMATION CONFIGURATION SERVER ID DOES NOT MATCH WHAT IS IN PROFILE"
NVD4255I	"Client Automation Configuration Server ID in Profile ==> <%s>"
NVD4256I	"Client Automation Configuration Server ID in Database ==> <%s>"
NVD4257I	"Please change the Client Automation Configuration Server ID in the Profile"
NVD4258E	"MANAGING CLIENT AUTOMATION CONFIGURATION SERVER ID <%s> AND CLIENT AUTOMATION CONFIGURATION SERVER ID <%s>"

Message Number	Log Message
NVD4259E	"SHOULD BE DIFFERENT FOR THE SERVER-TYPE OF CLIENT AUTOMATION CONFIGURATION SERVER"
NVD4260E	"SHOULD BE THE SAME FOR STANDALONE CLIENT AUTOMATION CONFIGURATION SERVER"
NVD4261W	"Verify Domain <%s.%s.%s> cannot validate DMA instance count"
NVD4262E	"VERIFY DOMAIN <%s.%s> DMA INSTANCE COUNT DOES NOT EQUAL ACTUAL INSTANCE COUNT"
NVD4263E	"VERIFY DOMAIN <%s.%s> TotalInstanceCount IS <%ld>, actual instance count is <%ld>"
NVD4264W	"Verify Domain <%s.%s.%s> cannot validate DMA instance count"
NVD4265E	"VERIFY DOMAIN <%s.%s> CLASSES COUNT DOESN'T EQUAL ACTUAL CLASS COUNT"
NVD4266I	"Verify Domain DMA TotalClassCount is <%ld>, actual class count is <%ld>"
NVD4267I	"Verifying File <%s>"
NVD4268E	"FILE <%s> CONTAINS INVALID DOMAIN NAME (SEE STDERR)"
NVD4269E	"VERIFY FILE <%s.%s> TERMINATED BY FindNextDomain(...) RC=<%d>"
NVD4270E	"ERROR READING <%-s> CONTROL CLASS FOR FILE <%-s>"
NVD4271I	"Please use ZEDMAMS to convert the <%-s> file"
NVD4272E	"VERIFY FILE <%s> CHECKED <%ld> DOMAINS, FOUND <%ld> ERRORS"
NVD4273I	"Verify File <%s> checked <%ld> domains, found <%ld> errors"
NVD4274I	"Verify File <%-s> autofix <%c> using ZEDMAMS <%-s> in progress..."
NVD4275I	"Verify File <%-s> autofix ZEDMAMS REFRESH_DMA ended with rc <%ld>"
NVD4276E	"INVALID FILE NAME (SEE STDERR)"
NVD4277E	"VERIFY FILE <%s.%s> TERMINATED BY FindNextFile(...) RC=%d"
NVD4278E	"DATABASE HAS <%ld> FILES WITH ERRORS"
NVD4279E	"VERIFY DATABASE FILES - FOUND <%ld>, CHECKED <%ld>"
NVD4280I	"Verify Database Files - Found <%ld>, checked <%ld>"
NVD4281E	"VERIFY DATABASE FILES - FILE SPECIFIC ERRORS <%ld>, AUTOFIXED <%ld>"
NVD4282I	"Verify Database Files - file specific errors <%ld>, autofixed <%ld>"

Message Number	Log Message
NVD4283E	"VERIFY DATABASE DOMAINS - FOUND <%ld>, CHECKED <%ld>"
NVD4284I	"Verify Database Domains - Found <%ld>, checked <%ld>"
NVD4285E	"VERIFY DATABASE DOMAINS - DOMAIN SPECIFIC ERRORS <%ld>, AUTOFIXED <%ld>"
NVD4286I	"Verify Database Domains - Domain specific errors <%ld>, autofixed <%ld>"
NVD4287E	"VERIFY DATABASE CLASSES - FOUND <%ld>, CHECKED <%ld>"
NVD4288I	"Verify Database Classes - Found <%ld>, checked <%ld>"
NVD4289E	"VERIFY DATABASE CLASSES – CLASS SPECIFIC ERRORS <%ld>, AUTOFIXED <%ld>"
NVD4290I	"Verify Database Classes - Class specific errors <%ld>, autofixed <%ld>"
NVD4291E	"VERIFY DATABASE INSTANCES - FOUND <%ld>, CHECKED <%ld>"
NVD4292I	"Verify Database Instances - Found <%ld>, checked <%ld>"
NVD4293E	"VERIFY DATABASE INSTANCES - DOMAIN SPECIFIC ERRORS <%ld>, AUTOFIXED <%ld>"
NVD4294I	"Verify Database Instances - Domain specific errors <%ld>, autofixed <%ld>"
NVD4295W	"Database Verification rc <%ld>, autofix retry in progress"
NVD4296E	"DATABASE VERIFICATION AUTOFIX RETRY ENDED WITH RC <%ld>"
NVD4297I	"Database Verification autofix retry ended with rc <%ld>"
NVD4298E	"DATABASE VERIFICATION ENDED WITH RC <%ld>"
NVD4299I	"Database Verification ended with rc <%ld>"
4300-4399	Database
NVD4300I	"Verify Class <%s.%s.%s> instances found <%ld>, verified <%ld>, errors <%ld>"
NVD4301I	"Verify Domain <%s.%s> classes found <%ld>, verified <%ld>, errors <%ld>"
NVD4302W	"Verify Instance <%s.%s.%s.%s> mismatched Client Automation Configuration Server ID with instance objid <%12.12s>"
NVD4303W	"Verify Instance <%s.%s.%s.%s> mismatched instance objcid <%12.12s> and class objid <%12.12s>"
NVD4304I	"%s *%.*s"
5000-5099	Variable Substitution

Message Number	Log Message
NVD5001E	"SUBSTITUTION ERROR RC = <%d>"
NVD5002E	"SUBSTITUTION VALUE TRUNCATE <%s.%s> <%d> ACTUAL <%d> ALLOCATED <%d> <%s>"
NVD5003I	"Substituted Value <%8.8s.%8.8s> <%d> ACTUAL <%d> ALLOCATED <%d> <%s>"
NVD5004I	"Substitution Closing Brace not found: <%s>"
NVD5005I	"Passing to Substitution....: <%s>"
NVD5006I	"Back from Substitution....: %d <%s>"
NVD5007E	"SUBSTITUTION BUFFER OVERFLOW"
NVD5008E	"SUBSTITUTION BUFFER TRUNCATED"
NVD5009I	"Substitution needed again, depth %d <%s>"
NVD5010I	"After Substitution <%s>"
NVD5011I	"Passed to Substitution ..: <%s>"
NVD5012E	"INVALID SUBSTITUTION SYNTAX <%s>"
NVD5013E	"ERROR IN SUBSTITUTION <%s>"
NVD5014E	"SUBSTITUTION BUFFER OVERFLOWS <%s>"
NVD5015I	"<%s>"
NVD5016I	"Variable not found <%s>"
NVD5017I	"Truncated Value during Substitution"
NVD5018E	"INVALID SUBSTITUTION VARIABLE <%s> FOR OBJECT <%s>"
NVD5019I	"Variable not found <%s> <%s>"
NVD5020I	"Variable <%8.8s> not found in <%8.8s>"
NVD5021I	"GET <%8.8s.%8.8s> <%d> <%d> <%s>"
NVD5022E	"VALUE TRUNCATED <%8.8s.%8.8s> <%d> ACTUAL <%d> ALLOCATED <%d> <%s>"
NVD5023W	"SBUFFER <%s> length = <%d>"
NVD5024W	"VALUE <%s> length = <%d>"
NVD5025E	"SUBSTITUTION FAILED <%s>"
NVD5026I	"Substituting <%s>"

Message Number	Log Message
NVD5027I	"Substituted value <%s>"
NVD5028E	"UNRESOLVED VARIABLE LIST <%s>"
NVD5029E	"SUBSTITUTION VALUE TRUNCATED ACTUAL <%d>ALLOCATED <%d>"
NVD5030E	"SUBSTITUTION VALUE TRUNCATED <%8.8s.%8.8s> <%d> ACTUAL <%d> ALLOCATED <%d> <%s>"
NVD5031I	"Protected field <%s> will not be returned"
NVD5032I	"Protected field <%s> will not be unloaded"
NVD5033E	"SUBSTITUTION FAILED BECAUSE RECURSION LIMIT OF <%d> REACHED"
NVD5034E	"INVALID SUBSTITUTION OBJECT <%s>"
NVD5035E	"INVALID SUBSTITUTION HEAP <%s> FOR OBJECT <%s> VARIABLE <%s>"
NVD5036E	"UNABLE TO PROCESS CLASS CONNECT OPTIONS <%s>, CONNECTION REJECTED"
NVD5037E	"CLASS CONNECT CONDITIONS <%s> ARE NOT WELL-FORMED, CONNECTION REJECTED"
NVD5038E	"SUBSTITUTION ERROR RC %d %s"
NVD5039E	"SUBSTITUTION VALUE TRUNCATED I <%d> O <%d>"
NVD5040I	"Reset DOMAIN after Method D= %s M= %s"
5100-5199	Instances
NVD5101I	"Dynamic INSTANCE Length <%d> at <%x>"
NVD5102I	"BASE INSTANCE No Inherit Requested"
NVD5103E	"INSTANCE <%s.%s.%s.%s> IS 0 BYTES LONG"
NVD5104I	"Using Base Instance Date/Time"
NVD5105I	"Instance not found <%s>"
NVD5106I	"%s. Domain.Instance = <%s.%s>"
NVD5107I	"'%s' INSTANCE <%d> at <%p>"
NVD5108I	"'%8.8s' INSTANCE <%d> at <%p>"
NVD5109E	"ERROR PROCESSING INSTANCES"
NVD5110E	"ERROR GETTING INSTANCE (%d BYTES) FROM <%s.%s.%s> RC=<%d>"

Message Number	Log Message
NVD5111E	"ERROR <RC=%d> BUILDING PATCH INSTANCE <%s.%s.%s.%s>"
NVD5112E	"ERROR <RC=%d> CREATING RESOURCE INSTANCE <%s.%s.%s.%s>"
NVD5113E	"ERROR <RC=%d> CREATING INSTANCE <%s.%s.%s.%s>"
NVD5114E	"BASE INSTANCE ALREADY EXISTS <%s.%s.%s.%s>"
NVD5115E	"ERROR <RC=%d> WRITING BASE INSTANCE <%s.%s.%s.%s>"
NVD5116E	"INSTANCE ALREADY EXISTS <%s.%s.%s.%s>"
NVD5117E	"ERROR OPENING INSTANCES <%s.%s.%s.%s>"
NVD5118E	"ERROR UNMATED INSTANCE <%s.%s.%s.%s>"
NVD5119E	"ERROR INSTANCE <%s.%s.%s.%s> HAS INVALID DATA; THEREFORE UNMATED"
NVD5120E	"ERROR INSTANCE <%s.%s.%s.%s> SIZE LESS THAN MINIMUM REQUIRED"
NVD5121I	"Actual instance size <%d>, minimum instance size <%d>"
NVD5122E	"INSTANCE <%s.%s.%s.%s> INSTANCE PREFIX TIME <%-8.8s> IS INVALID"
NVD5123E	"INSTANCE <%s.%s.%s.%s> DOES NOT MATCH CLASS TEMPLATE"
NVD5124I	"Actual instance size <%d>, expected instance size <%d>"
5200-5299	Objects and Variables
NVD5201E	"ERROR, UNABLE TO FIND OBJECT <%s>"
NVD5202E	"ERROR, UNABLE TO FIND PARENT OBJECT"
NVD5203E	"ERROR, UNABLE TO PROCESS OBJECT <%s>"
NVD5204E	"VARGET ERROR FOR <%s> IN <%s>"
NVD5205E	"ERROR, UNABLE TO FIND VARIABLE <%s.%s>"
NVD5206E	"ERROR, UNABLE TO GET VARIABLE <%s.%s>"
NVD5207E	"ERROR, UNABLE TO GET TEMPLATE"
NVD5208E	"ERROR, UNABLE TO GET HEAP <%d> FOR OBJECT <%s>"
NVD5209E	"ERROR, UNABLE TO GET OBJECT <%s>"
NVD5210E	"ERROR, UNABLE TO PUT VARIABLE <%s.%s>"
NVD5211E	"ERROR, UNABLE TO PUT OBJECT <%s>"

Message Number	Log Message
NVD5212E	"ERROR, INVALID OBJECT NAME <%s>"
NVD5213E	"ERROR, INVALID VAR NAME <%s>"
NVD5214E	"VARIABLE NOT FOUND <%s.%s>"
NVD5215E	"COMPOBJ - OBJECT NOT FOUND"
NVD5216E	"ERROR INITIALIZING OBJECT <%s>"
NVD5217I	"OBJECT Variable <%s> Length is 0 - Ignored"
NVD5218I	"Delete variable <%0.8s.%0.8s>"
NVD5219E	"DELETE VARIABLE - OBJECT <%0.8s> NOT FOUND"
NVD5220E	"DELETE VARIABLE - VARIABLE <%0.8s.%0.8s> NOT FOUND"
NVD5221I	"Starting OBJECT: <%s>"
NVD5222E	"VARGET ERROR DMATIME NOT FOUND"
NVD5223E	"VARGET ERROR DMAQTYPE NOT FOUND"
NVD5224E	"VARGET ERROR DMAMGRID NOT FOUND"
NVD5225E	"ERROR DMAMGRID DOES NOT VERIFY"
NVD5226E	"VARGET ERROR ZOBJFILE NOT FOUND FOR <%s>"
NVD5227E	"VARGET ERROR ZOBJDOMN NOT FOUND FOR <%s>"
NVD5228E	"VARGET ERROR ZOBJCLAS NOT FOUND FOR <%s>"
NVD5229E	"VARGET ERROR ZACTION NOT FOUND FOR <%s>"
NVD5230E	"VARGET ERROR ZEDMTYPE NOT FOUND FOR <%s>"
NVD5231E	"VARGET ERROR ZOBJNAME NOT FOUND FOR <%s>"
NVD5232E	"VARGET ERROR ZOBJID NOT FOUND FOR <%s>"
NVD5233E	"ERROR ZEDMTYPE <%s> INVALID FOR ACTION <%s>"
NVD5234I	"Starting OBJ: <%8.8s>"
NVD5235E	"VARGET ERROR ZRSCMFIL NOT FOUND FOR <%8.8s>"
NVD5236E	"VARGET ERROR <%s> NOT FOUND FOR <%8.8s>"
NVD5237E	"INVALID ZEDMTYPE <%s>"
NVD5238E	"INVALID CHARACTER IN VAR NAME <%s>"

Message Number	Log Message
NVD5239E	"ZTVARQ = NULL"
NVD5240I	"Variable Server - <%8.8s> OBJECT not found"
NVD5241E	"MISSING <%8.8s> VARIABLE IN <%s>"
NVD5242I	"%s: ztvarq <%08X>, zqhead <%08X>, zqtail <%08X>"
NVD5243E	"%s: TAIL ZPCB NOT FREED <%08X>"
NVD5244E	"%s: HEAD ZPCB NOT FREED <%08X>"
NVD5245E	"VARIABLE LENGTH <%d> EXCEEDS LIMIT <%d>"
NVD5246E	"OBJECT <%s> VARIABLE <%s>"
NVD5247E	"ERROR INITIALIZING OBJECT <%s>"
NVD5248E	"UNABLE TO PACK OBJECTID <%s> FOR <%s>.<%s>"
NVD5249E	"UNABLE TO PACK CRC <%s> FOR <%s>.<%s>"
NVD5250I	"%s started at <%2.2d:%2.2d:%2.2d:%3.3d>"
NVD5251I	"%s completed at <%2.2d:%2.2d:%2.2d:%3.3d>"
NVD5252I	"%s (V4.0 or newer) started at <%2.2d:%2.2d:%2.2d:%3.3d>"
NVD5253I	"%s (Pre V4.0) started at <%2.2d:%2.2d:%2.2d:%3.3d>"
NVD5254I	"Object exchange ended"
NVD5255I	"ADD <%8.8s.%8.8s> <%d> <%d> <%s>"
NVD5256I	"MOVE <%8.8s.%8.8s> <%d> <%d> <%s>"
NVD5257I	"Template size mismatch A<%d> Allocated <%d>"
NVD5258E	"TEMPLATE SIZE EXCEEDS ALLOCATED SPACE ACTUAL <%d> ALLOC <%d>"
NVD5259I	"Template copied to <%08X>, length <%d>"
NVD5260I	"Freeing saved template @ <%08X>, length <%d>"
NVD5261I	"Recd. <%s>"
NVD5262I	"Template Length <%d> bytes"
NVD5263I	"Template <%d>"
NVD5264E	"ZPCB NOT FOUND FOR OBJECT <%s>"
NVD5265I	"%s.%s Variable Value _UNDEF_ - Variable not Put"

Message Number	Log Message
NVD5266I	"<%8.8s.%8.8s> Variable Value _UNDEF_ - Variable not Put "
NVD5267I	"*****NOT USED *****"
NVD5268I	"PUT <%8.8s.%8.8s> <%d> <%d> <%s>"
NVD5269I	"Delete all requires lists starting"
NVD5270E	"DELETING REQUIRES LIST FROM <%s>"
NVD5271W	"NONDIGIT in cumulative variable <%8.8s> - treating as regular variable"
NVD5272W	"Overflow Condition on cumulative variable <%8.8s>"
NVD5273E	"ERROR ADDING NEW VARIABLE <%s> TO OBJECT <%s> HEAP <%d>"
NVD5274E	"VARPUT ERROR IN <%s> <%s>"
NVD5275E	"VARGET ERROR ZRSCSTYP NOT FOUND FOR <%s>"
NVD5276E	"VARGET ERROR ZRSCCFIL NOT FOUND FOR <%s>"
NVD5277E	"ERROR GETTING VARIABLES FOR OBJECT <%s>"
NVD5278E	"ERROR REQUIRED VARIABLES NOT FOUND IN TEMPLATE <%s>"
NVD5279E	"VARGET ERROR ZADMCLAS NOT FOUND FOR <%s>"
NVD5280E	"VARGET ERROR ZADMINST NOT FOUND FOR <%s>"
NVD5281E	"ERROR, MISSING KEY VARIABLE 'ZRSCSTYP' FROM CLASS <%s>.<%s>.<%s>"
NVD5282E	"VARGET ERROR <%s> NOT FOUND IN <%s>"
NVD5283E	"VARPUT ERROR RC<%d> PUTTING VARIABLE <%8.8s> INTO OBJECT <%8.8s> IN HEAP <%d>"
NVD5284E	"ERROR GETTING VARIABLE FROM OBJECT <%s>"
NVD5285E	"ERROR RC<%d> GETTING VARIABLE <%8.8s> FROM OBJECT <%8.8s>"
NVD5286E	"VARPUT ERROR FOR <%8.8s> IN <%s>"
NVD5287E	"VARPUT ERROR ZLUNAME INTO ZMASTER"
NVD5288E	"VARPUT ERROR ZIPNAME INTO ZMASTER"
NVD5289W	"Missing <%s> variable in <%8.8s>"
NVD5290E	"<%s> VARIABLE IS ALL BLANKS IN <%8.8s> INSTANCE #<%d>"
NVD5291E	"INVALID <%s> VALUE IN <%8.8s> INSTANCE #<%d>"

Message Number	Log Message
NVD5292E	"PUT_VAR ERROR O<%s> V<%s>"
NVD5293E	"ERROR DURING VARPUT"
NVD5294I	"%s *%. *s"
NVD5295E	"ERROR TRANSLATING DATA FOR INBOUND OBJECT <%8.8s>, ATTRIBUTE <%8.8s>, HEAP <%lu>, FROM <%s> TO <%s>"
NVD5296E	"ERROR, TRANSLATION BUFFER TOO SMALL, SPECIFIED SIZE <%lu>, REQUIRED SIZE <%lu>"
NVD5298E	"ERROR, ACTUAL DATA SIZE <%lu> BYTES EXCEEDS MAXIMUM <%lu>, TRUNCATED DATA LOST <%s>"
NVD5299E	"ERROR, DATA TRUNCATED TO <%lu> BYTES, <%s>"
5300-5399	Object Processing
NVD5301I	"VERSION OBJECT Processing Begins: <%s>"
NVD5302I	"GENERIC OBJECT Processing Begins: <%s>"
NVD5303I	"HIDDEN GROUP OBJECT Processing Begins: <%s>"
NVD5304I	"Termination requested"
NVD5305I	"PUT-OBJECT control object: <%8.8s> Instance: <%08.8d>"
NVD5306I	"PUT-OBJECT: <%8.8s>, id: <%12.12s>, pid: <%12.12s>, name: <%s>"
NVD5307I	"PUT-OBJECT: <%8.8s> Instances Filtered by %s: <%12.12s>"
NVD5308I	"PUT-OBJECT: <%8.8s> All Instance Selected"
NVD5309I	"PUT-OBJECT: No Instances found for parent ID <%s>"
NVD5310I	"PUT-OBJECT: No Instances found for parent ID <%s> that were updated after <%s> <%s>"
NVD5311I	"PUT-OBJECT select objid: <%12.12s> actual: <%12.12s> <%d>"
NVD5312I	"PUT-OBJECT select parent ID: <%12.12s> actual: <%12.12s> <%d>"
NVD5313I	"PUT-OBJECT major objname: <%s> <%d>"
NVD5314I	"PUT-OBJECT hidden resource <%s>"
NVD5315E	"PUT-OBJECT ERROR - REQUESTED OBJECTID NOT FOUND"
NVD5316E	"ERROR IN NADAOBJ <%d>"
NVD5317I	"ZPUTOBJ Confirmed"

Message Number	Log Message
NVD5318E	"PUT-OBJECT - RESTART OFFSET NON-NUMERIC IN RESOURCE RESTART REQUEST"
NVD5319I	"Request file restart offset: <%d>"
NVD5320E	"PUT-OBJECT REQUESTED OBJID NOT FOUND"
NVD5321I	"<ZRSCMMEM> not found for object <%s>"
NVD5322I	"Honoring RESTART %d"
NVD5323I	"Get-Storage starting Object: <%8.8s>"
NVD5324E	"PUT-OBJECT - UNABLE TO PROCESS STORAGE OBJECT"
NVD5325E	"ERROR SENDING DYNAMIC OBJECT <%8.8s>"
NVD5326I	"Get-Database Object Starting"
NVD5327I	"Get-Database Object Ending"
NVD5328E	"GET-DATABASE OBJECT ERROR"
NVD5329E	"PUT-OBJECT OBJECTID FORMAT ERROR <%12.12s>"
NVD5330E	"PUT-OBJECT <%s> - OBJECT NOT FOUND"
NVD5331E	"OBJECT NOT FOUND <%s>"
NVD5332E	"GET OBJECT ERROR - NO HEAPS OR ERROR RECEIVING DATA"
NVD5333E	"MISSING OBJECT <%s>"
NVD5334I	"Deleting %s object <%s>"
NVD5335E	"BLANK OBJECT NAME"
NVD5336I	"%s: Delete All Objects has started"
NVD5337E	"%s: DELETE ALL OBJECTS - NO OBJECTS FOUND"
NVD5338I	"Delete Object Instance starting"
NVD5339I	"OBJECT - <%s> HEAP - <%d>"
NVD5340E	"DELETE OBJECT INSTANCE - OBJECT NOT FOUND"
NVD5341I	"(Obj) <%8.8s> (Typ) <%8.8s> (Len) <%d>"
NVD5342I	"Object <%8.8s> Template <%d> Heap <%d>"
NVD5343E	"GETBLK ERROR"
NVD5344I	"OBJECT ID match not found"

Message Number	Log Message
NVD5345E	"SOURCE OBJECT NOT FOUND <%8.8s>"
NVD5346E	"DESTINATION OBJECT NOT FOUND <%8.8s>"
NVD5347I	"Source OBJECT <%8.8s> @ %x ; Destination OBJECT <%8.8s> @ %x"
NVD5348E	"NO HEAP FOR OBJECT <%8.8s>"
NVD5349W	"Cleanup object not found <%8.8s>"
NVD5350I	"FREE Directory at <%p>"
NVD5351E	"DESTINATION OBJECT ERROR - VALUE NULL"
NVD5352E	"INVALID DESTINATION OBJECT NAME (Null)"
NVD5353E	"INVALID DESTINATION OBJECT NAME (Blanks)"
NVD5354E	"SOURCE OBJECT ERROR"
NVD5355E	"INVALID SOURCE OBJECT NAME (Null)"
NVD5356E	"INVALID SOURCE OBJECT NAME (Blanks)"
NVD5357E	"HEAP NOT FOUND FOR DESTINATION OBJECT"
NVD5358E	"NO HEAPS FOUND FOR SOURCE OBJECT"
NVD5359I	"PUT-OBJECT: No Instances found for object id <%s>"
NVD5360I	"PUT-OBJECT: No Instances found for object id <%s> that were updated after <%s> <%s>"
NVD5361I	"PUT-OBJECT: No Instances found for parent id <%s>, object id <%s>"
NVD5362I	"PUT-OBJECT: No Instances found for parent id <%s>, object id <%s>, that were updated after <%s> <%s>"
NVD5363I	"PUT-OBJECT: No Instances found that were updated after <%s> <%s>"
NVD5364E	"ERROR CREATING OBJECT FOR <%s.%s.%s.%s>"
NVD5365E	"ERROR LOCATING PARENT OBJECT <%8.8s>"
NVD5366E	"ERROR LOCATING ZCLIST FOR <%8.8s>"
NVD5367I	"Listing ILISTs for all heaps of <%8.8s>"
NVD5368I	"%8.8s ILIST for heap:<%d>, parentid:<%12.12s>, date/time:<%8.8s/%8.8s>, children#:<%d>"
NVD5369E	"ERROR LOCATING ZILIST FOR <%8.8s>"
NVD5370E	"ERROR LOCATING ZCLIST FOR PARENT/CHILD: <%8.8s / %8.8s>"

Message Number	Log Message
NVD5371I	"Updating child date/time entry for <%s> <%8.8s>"
NVD5372I	"%8.8s %8.8s"
NVD5373I	"Checking for requires crc"
NVD5374I	"Checking for includes objectid"
NVD5375I	"Cannot add OBJECTID to ZOBJVPID"
NVD5376I	"User: <%s> Connect message: <%s>"
NVD5377E	"ERROR PUTTING OBJECT"
NVD5378I	"Master Object exchange ended"
NVD5379I	"Configuration Unchanged -- Deleted ZRSOURCE object"
NVD5380W	"Configuration Unchanged -- ZRSOURCE object deletion failed, rc = <%d>"
NVD5381I	"Free unused pool elements requested by <%-8.8s>"
NVD5382I	"Filtering by ZOBJPID <%12.12s>"
NVD5383E	"ERROR EXCHANGING OBJECT <%s>"
NVD5384I	"Deleting request object <%s>"
NVD5385I	"Deleting control object <%s>"
NVD5386I	"%8.8s %04d: Child <%8.8s> Count <%8.8s> Crc <%8.8s> Date <%8.8s> Time <%8.8s>"
NVD5387E	"ZOBJVERS ITEM IS ALL BLANK "
NVD5388E	"ZOBJVERS ITEM DOES NOT MATCH RULE <%s>"
NVD5389E	"ZOBJVERS SUCCESS "
NVD5390I	"Configuration Unchanged -- ZERASE=<%s>, ZRSOURCE object was not deleted"
NVD5391I	"Processing <object_name>, heap size: <heap_size>, total number of heaps: <max_heap>"
NVD5392I	"<%s %s(%12.12s) heap(%03d)"
NVD5393E	"ICONV TRANSLATION ERROR <%d - %s>"
NVD5394E	"ERROR, UNABLE TO TRANSLATE CHARACTER <0x%02X> BEGINNING AT OFFSET <%d>"
NVD5395E	"ERROR TRANSLATING DATA FOR OUTBOUND OBJECT <%8.8s>, ATTRIBUTE <%8.8s>, HEAP <%lu>, FROM <%s> TO <%s>"

Message Number	Log Message
NVD5396E	"ERROR, TRANSLATION BUFFER TOO SMALL, SPECIFIED SIZE <%lu>, REQUIRED SIZE <%lu>"
NVD5398E	"ERROR, ACTUAL DATA SIZE <%lu> BYTES EXCEEDS MAXIMUM <%lu>, TRUNCATED DATA LOST <%s>"
NVD5399E	"ERROR, DATA TRUNCATED TO <%lu> BYTES, <%s>"
5400-5499	Resource
NVD5401W	"Warning - RESOURCE Size (ZRSCSIZE) NON-NUMERIC <%s> Length <%d>"
NVD5402W	"Warning - RESOURCE has ZRSCSIZE=0 RESOURCE: <%s>"
NVD5403I	"RESOURCE <%s>"
NVD5404E	"MISSING RESOURCE <%s> DOMAIN <%s> CLASS <%s>"
NVD5405E	"ERROR LOCKING RESOURCE <%s>"
NVD5406W	"Warning - ACTUAL DATA for Resource does NOT match ZRSCSIZE"
NVD5407W	"ZRSCSIZE: <%d>, ACTUAL: <%d> OFFSET: <%d>"
NVD5408E	"ERROR UNLOCKING RESOURCE <%s>"
NVD5409E	"ERROR LOCKING RESOURCE <%s.%s.%s.%s>"
NVD5410E	"ERROR WRITING RESOURCE PREFIX <%s.%s.%s.%s>"
NVD5411E	"ERROR WRITING RESOURCE FILE <%s.%s.%s.%s>"
NVD5412E	"ERROR OPENING RESOURCE <%s.%s.%s.%s>"
NVD5413E	"ERROR READING PREFIX FOR RESOURCE <%s.%s.%s.%s>"
NVD5414E	"RESOURCE <%s.%s.%s.%s> PREFIX SIZE <%lu> IS SMALLER THAN EXPECTED <%lu>"
NVD5415E	"RESOURCE <%s.%s.%s.%s> PREFIX CLASS NAME <%8.8s> DOES NOT MATCH"
NVD5416E	"RESOURCE <%s.%s.%s.%s> PREFIX INSTANCE NAME <%32.32s> DOES NOT MATCH <%32.32s>"
NVD5417E	"ERROR, ASSOCIATED RESOURCE <%s.%s.%s.%s> HAS INVALID SIZE <%lu>"
NVD5418E	"RESOURCE <%s.%s.%s.%s> ACTUAL SIZE <%lu> IS NOT EQUAL TO PREFIX RESOURCE SIZE <%lu>"
NVD5419E	"RESOURCE <%s.%s.%s.%s> %s <%lu> IS NOT EQUAL TO PREFIX RESOURCE SIZE <%lu>"

Message Number	Log Message
NVD5420E	"RESOURCE <%s.%s.%s.%s> CRC <%.*s> IS NOT EQUAL TO PREFIX RESOURCE CRC <%.*s>"
NVD5421I	"Copying file <%s> to <%s.%s.%s.%s>"
NVD5422I	"Bytes read <%lu>, bytes written <%lu>"
NVD5423I	"Copy file <%s> to <%s.%s.%s.%s> completed successfully"
NVD5424E	"COPY FILE <%s> TO <%s.%s.%s.%s> FAILED"
NVD5425I	"Imploding file <%s> to <%s.%s.%s.%s>"
NVD5426I	"Implode file bytes read <%lu>, bytes written <%lu>"
NVD5427I	"Implode file <%s> to <%s.%s.%s.%s> completed successfully"
NVD5428E	"IMPLODE FILE <%s> TO <%s.%s.%s.%s> FAILED"
NVD5429I	"Patch Size <%lu>, 'TO' Resource Size <%lu>, Max Patch Size <%lu>, Max Patch Percent <%lu>"
NVD5430W	"Patch <%s.%s.%s.%s> deleted because it does not meet patch size limiting criteria"
NVD5431I	"Copying <%s.%s.%s.%s> to file <%s>"
NVD5432I	"Bytes read <%lu>, bytes written <%lu>"
NVD5433I	"Copy <%s.%s.%s.%s> to file <%s> completed successfully"
NVD5434E	"COPY <%s.%s.%s.%s> TO FILE <%s> FAILED"
NVD5435I	"Exploding <%s.%s.%s.%s> to file <%s>"
NVD5436I	"Bytes read <%lu>, bytes written <%lu>"
NVD5437I	"Explode <%s.%s.%s.%s> to file <%s> completed successfully"
NVD5438E	"EXPLODE <%s.%s.%s.%s> TO FILE <%s> FAILED"
NVD5439E	"PATCH BUILD FAILED, RC <%d>, DURATION <%1.3f> SECONDS, FOR <%s.%s.%s.%s>"
NVD5440I	"Patch build complete, rc <%d>, duration <%1.3f> seconds, for <%s.%s.%s.%s>"
NVD5441E	"ERROR READING 'FROM' INSTANCE <%s.%s.%s.%s>"
NVD5442E	"ERROR READING 'TO' INSTANCE <%s.%s.%s.%s>"
NVD5443W	"GET INSTANCE <%s.%s.%s.%s>, VARIABLE <%0.*s>, SIZE <%1.3d>, VALUE <%0.*s> TRUNCATED TO <%0.*s>"

Message Number	Log Message
NVD5444I	"Get instance <%s.%s.%s.%s>, variable <%0.*s>, size <%1.3d>, value <%0.*s>"
NVD5445E	"GET INSTANCE <%s.%s.%s.%s>, VARIABLE <%0.*s>, NOT FOUND"
NVD5446W	"PUT INSTANCE <%s.%s.%s.%s>, VARIABLE <%0.*s>, SIZE <%1.3d>, VALUE <%0.*s> TRUNCATED TO <%0.*s>"
NVD5447I	"Put instance <%s.%s.%s.%s>, variable <%0.*s>, size <%1.3d>, value <%0.*s>"
NVD5448E	"PUT INSTANCE <%s.%s.%s.%s>, VARIABLE <%0.*s>, NOT FOUND"
NVD5449E	"ERROR READING INSTANCE <%s.%s.%s.%s>"
NVD5450I	"Patch build started for <%s.%s.%s.%s>"
NVD5451I	"Read Class <%s.%s.%s.%s>, size <%d>, successful"
NVD5452I	"Read Instance <%s.%s.%s.%s>, size <%d>, successful"
NVD5453I	"Write Instance <%s.%s.%s.%s>, size <%d>, successful"
NVD5454E	"CANNOT OBTAIN PATCH 'FROM' FILE INFORMATION FOR <%s>"
NVD5455E	"CANNOT OBTAIN PATCH 'TO' FILE INFORMATION FOR <%s>"
NVD5456I	"FROM file <%s>, size <%lu>"
NVD5457I	"TO file <%s>, size <%lu>"
NVD5458I	"PATCH file <%s>"
NVD5459I	"File contents are identical, no patch built"
NVD5460I	"PATCH file <%s> created, size <%lu>"
NVD5461E	"CANNOT OBTAIN 'PATCH' FILE INFORMATION FOR <%s>"
NVD5462I	"Patch build queue request for <%s.%s.%s.%s>"
NVD5463E	"PATCH BUILD QUEUE REQUEST FAILED, RC <%d>, DURATION <%1.3f> SECONDS, FOR <%s.%s.%s.%s>"
NVD5464I	"Patch build queue request complete, rc <%d>, duration <%1.3f> seconds, for <%s.%s.%s.%s>"
NVD5465E	"PATCH 'TO' FILE IS TOO SMALL, <%lu> BYTES, NO PATCH BUILT FOR <%s>"
NVD5466W	"Warning - RESOURCE Size (ZCMPSize) NON-NUMERIC <%s> Length <%d>"

Message Number	Log Message
NVD5467E	"MISSING FILE NAME <%0.8s.%0.8s>"
NVD5468E	"MISSING DOMAIN NAME <%0.8s.%0.8s>"
NVD5469E	"MISSING CLASS NAME <%0.8s.%0.8s>"
NVD5470E	"MISSING SERVICE OBJECT ID <%0.8s.%0.8s>"
NVD5471E	"MISSING SOURCE INSTANCE NAME <%0.8s.%0.8s>"
NVD5472E	"MISSING DESTINATION INSTANCE NAME <%0.8s.%0.8s>"
NVD5473E	"CANNOT BUILD PATCH FOR THE SAME INSTANCE NAMES <%s>"
NVD5474E	"FILE NOT FOUND <%s>"
NVD5475E	"DOMAIN NOT FOUND <%s.%s>"
NVD5476E	"CLASS NOT FOUND <%s.%s.%s>"
NVD5477E	"INSTANCE NOT FOUND <%s.%s.%s.%s>"
NVD5478E	"ERROR <%d> RENAMING OBJECT <%s> TO <%s>"
NVD5479E	"ERROR <%d> SORTING OBJECT <%s> BY <%s>"
NVD5480E	"%s: ERROR <%d> GETTING ZRSCCFIL FROM PATCH_S OBJECT"
NVD5481E	"%s: ERROR <%d> GETTING ZRSCCFIL FROM PATCH_T OBJECT"
NVD5482E	"%s: ERROR <%d> COPYING VARS FROM PATCH_T TO PATCHDUP OBJECT"
NVD5483I	"Resolving <%s>"
NVD5484E	"%s: ERROR <%d> GETTING OBJECT <%s> HEAP <%d> VARIABLE <%0.8s>"
NVD5485E	"%s: ERROR <%d> SETTING OBJECT <%s> HEAP <%d> VARIABLE <%0.8s> TO <%s>"
5800-5899	PROC_API
NVD5800I	"%s: version <%4.4s>, reqname <%6.6s>, flag 0x%02X, object <%s>, varname <%s>, heapno <%lu>, length <%u>, retcode <%d>, buflen <%u>"
NVD5801E	"INVALID REQUEST <%6.6s>"
NVD5802I	"EDMGET O<%s> VN<%s> (L)<%d> V<%s>"
NVD5803I	"EDMGTR: Object<%s> Varname<%s> Heap<%d>"
NVD5804I	"EDMGTR: retcode <%ld>, actual size <%ld>, heapno <%ld>"

Message Number	Log Message
NVD5805W	"EDMGTR: retcode <%ld>, actual size <%ld>, heapno <%ld>"
NVD5806E	"EDMRFC CACHE CLOSED - CANNOT REFRESH"
NVD5807I	"EDMUPC * Updating %s"
NVD5808I	"EDMCLOSE"
NVD5809I	"EDMZRM <%s>"
NVD5810E	"EDMZRM ZERRORM HEAP NOT PUT"
NVD5811I	"EDMSMT <%s>"
NVD5812I	"EDMOID <%s>"
NVD5813E	"TOO MANY STRINGS PASSED"
NVD5814I	"EDMLGF <%s>"
NVD5815I	"EDMLGS <%s>"
NVD5816I	"EDMGTD <%s>"
NVD5817I	"EDMGTT <%s>"
NVD5818I	"EDMNT0 Class size <%d>"
NVD5819I	"EDMKLT <%s>"
NVD5820I	"EDMGCC <%s>"
NVD5821I	"EDMGCZ <%s>"
NVD5822I	"EDMOID: <%s> <%s>"
NVD5823I	"%s *%. *s"
5900-5999	EDMMAPID
NVD5900I	"EDMMAPID_OPEN Entered"
NVD5901E	"%s: UNABLE TO ALLOCATE MEMORY FOR THE EDMMAPI CONTROL BLOCK"
NVD5902I	"EDMMAPID_CLOSE Entered"
NVD5903I	"EDMDEL Object <%s>"
NVD5904I	"DELALL Entered"
NVD5905I	"DELALL : Deleting <%s>"
NVD5906I	"EDMGET : O <%s>, VN <%s>, L <%d>, V <%s>"

Message Number	Log Message
NVD5907I	"EDMSET : O <%s>, VN <%s>, H <%d>, L <%d>, V <%s>"
NVD5908E	"UNKNOWN RECORD TYPE <%s>"
NVD5910I	"EDMSTR : O <%s>, VN <%s>, H <%d>"
NVD5911I	"EDMGTR : O <%s>, V <%s>, H <%d>"
NVD5912E	"EDMGTR: OBJECT NOT FOUND"
NVD5913I	"EDMGTR: Template size <%d>"
NVD5914E	"EDMGTR: RECORD TOO SMALL. DATASIZE = %ld, MAX_REC_SIZE = %ld"
NVD5915E	"EDMGTR: HEAP NOT FOUND <%d>"
NVD5916I	"EDMGTR: Heap size <%d>"
NVD5917E	"EDMGTR: ERROR <%d> RETRIEVING DATA BUFFER"
NVD5918I	"EDMGTR: Data size <%d>"
NVD5919I	"Sending zero bytes across"
NVD5920I	"EDMGTR: Class size <%d>"
NVD5921I	"EDMGTR : Exit O <%s>, V <%s>, H <%d>, rc = <%d>"
NVD5922I	"EDMLOG : <%s>"
NVD5923E	"EDMERR : <%s>"
NVD5924I	"EDMOBJ : <%s> - find object"
NVD5925E	"EDMOBJ : OBJECT NOT FOUND <%s>"
NVD5926I	"EDMRFC * Refreshing Cache **"
NVD5927I	"EDMUPC * Updating Cache Entry**"
NVD5928I	"EDMUPC * Updating <%s.%s.%s.%s>"
NVD5929E	"ERROR UPDATING CACHED INSTANCE FOR <%s.%s.%s>"
NVD5930E	"ERROR ADDING INDEX CACHE ENTRY FOR <%s.%s.%s.%s>"
NVD5931I	"EDMCIN * Cache Info - print cache info **"
NVD5932I	"EDMSRT : Sorting Object <%s> Order <%d> Variables <%d>"
NVD5933I	"EDMCMP : <%s> - compress object in storage"
NVD5934E	"EDMNTC : ERROR RETRIEVING DATA BUFFER <%d>"

Message Number	Log Message
NVD5935E	"ERROR ALLOCATING TEMP STORAGE"
NVD5936E	"ERROR RE-ALLOCATING TEMP STORAGE"
NVD5937I	"Template copied to <%08X>, length <%d>"
NVD5938I	"Class conversion from network to host order. E <%d>"
NVD5939I	"EDMNTC : Class size <%d>"
NVD5940I	"EDMDHP : <%s> - delete heap"
NVD5941I	"EDMDVR : O <%s>, V <%s> - delete variable"
NVD5942I	"EDMSMQ - send message to queue"
NVD5943I	"EDMRES - resolve object"
NVD5944I	"EDMMIG <%8.8s --> %8.8s>"
NVD5945I	"EDMFYI : <%s> - get object info"
NVD5946I	"EDMFYI : <%s> T <%d> H <%d>"
NVD5947E	"EDMFYI : OBJECT NOT FOUND <%s>"
NVD5948I	"EDMZRM - create zerror instance"
NVD5949I	"EDMOID - create ObjectID"
NVD5950E	"EDMOID ERROR CREATING ObjectID"
NVD5951E	"EDMNTI : ERROR RETRIEVING DATA BUFFER <%d>"
NVD5952I	"EDMNTI : Instance size <%d>"
NVD5953I	"EDMSMT [ADMIN] - set method type"
NVD5954I	"Reset task type to administrator"
NVD5955W	"Could not reset administrator timeout to <%d> seconds"
NVD5956E	"WINDOWS EVENT HANDLE IS NOT OPEN"
NVD5957E	"WINDOWS ReportEvent FAILED, RC = <%d>"
NVD5958I	"Message sent to system log: <%s>"
NVD5959I	"EDMLGF - flush log"
NVD5960I	"EDMLGS - switch/dump log"
NVD5961I	"EDMBPO <%s> - build patch object"

Message Number	Log Message
NVD5962I	"EDMBPF <%s> - build path files"
NVD5963I	"EDMGTD - get directory name, directory_type <%c>, rec_buf @ <%08x>, length <%d>"
NVD5964I	"EDMGTD - directory name is <%s>, rc <%d>"
NVD5965I	"EDMGTT - get trace flag"
NVD5966I	"Recd. <%s>"
NVD5967I	"(Obj) <%8.8s>, (Typ) <%8.8s>, (Len) <%d>"
NVD5968I	"Increasing Storage size from <%d> to <%d> bytes"
NVD5969I	"Recvd. record of length <%d> bytes"
NVD5970I	"Recvd. <%s> of length <%d> bytes"
NVD5971W	"Heap Sent Before Template (O) <%8.8s> (T) <%8.8s> (S) <%d>"
NVD5972I	"Object <%8.8s> Template <%d> Heap <%d>"
NVD5973W	"Limit of <%i> LOG LINES reached for the METHOD, No more messages will be allowed"
NVD5974I	"EDMMKT <%s>"
NVD5975I	"EDMKLT <>"
NVD5976I	"EDMGMI database SHMID=<%i>"
NVD5977I	"EDMGCC <>"
NVD5978I	"EDMGCZ <>"
NVD5979I	"EDMDSC * Disabling Cache **"
NVD5980I	"EDMLDB * Locking Database **"
NVD5981I	"EDMUDB * Unlocking Database **"
NVD5982I	"EDMENC * Enabling Cache **"
NVD5983I	"Deleting old PATCHDUP object"
NVD5984E	"BUILDING PATCH RESOURCE OBJECT"
NVD5985E	"ERROR BUILDING PATCH RESOURCE OBJECT"
NVD5986E	"ERROR, UNABLE TO BUILD PATCH RESOURCE OBJECT"
NVD5987E	"NO PATCHES TO PROCESS"

Message Number	Log Message
NVD5988E	"INCORRECT CALL TO EDMGET; USE: EDMGET(object,number)"
NVD5989I	"%s *%.*s"
6000-6009	ZATTACH
NVD6001I	"Parent TaskID is <%ld>; Command Line is <%s>"
NVD6002E	"ERROR FORKING: <%s>"
NVD6003E	"USER FUNCTION RETURNED TO AN INVALID PLACE"
NVD6004E	"INTERNAL ERROR -- CLIENT AUTOMATION CONFIGURATION SERVER DOES NOT SUPPORT PROCESS-BASED SYSTEMS"
NVD6005E	"INTERNAL ERROR -- INVALID TASK TYPE OF <%c>"
NVD6006I	"Parent TaskID is <%ld>; Command Line is <%s>"
NVD6007E	"USER PROGRAM RETURNED TO AN INVALID PLACE"
NVD6008E	"ERROR STARTING METHOD <%s>"
NVD6009E	"ERROR <%ld> IN CREATE PROCESS FOR <%s>"
NVD6010E	"ERROR SPAWNING METHOD <%s> : BECAUSE <%s>, ERRNO = <%d>"
NVD6011E	"MAY BE TOO MANY ENVIRONMENT VARIABLES IN CONFIG.SYS"
NVD6012E	"CHILD: ERROR <%s> ALLOCATING MEMORY FOR ENVIRONMENT VARIABLE <%s>"
NVD6013E	"ERROR <%s> DURING PUTENV OF <%s>"
NVD6014I	"ZADMIN execution has been forced"
6100-6199	General Method Messages
NVD6100E	"%s: MISSING ZMTHPRMS IN ZMETHOD"
NVD6101E	"ERROR GETTING MAIL PATH"
NVD6102E	"ERROR GETTING DATABASE PATH"
NVD6103E	"ERROR PARSING INPUT PARAMETERS"
NVD6104E	"MISSING SENDERS ADDRESS"
NVD6105E	"MISSING RECIPIENTS ADDRESS"
NVD6106E	"ERROR IN zsys_getCurrentTime(), RC=<%d>"
NVD6107E	"ERROR IN CHECK-SUM, RC=<%d>"

Message Number	Log Message
NVD6108E	"FAILED TRYING TO FIND/CREATE DIRECORY <%s>"
NVD6109E	"ERROR OPENING <%s>"
NVD6110E	"ERROR %s NOTIFY HANDLE"
NVD6111E	"UNABLE TO SCHEDULE EMAIL REQUEST FOR <%s> FROM <%s> BECAUSE OF INTERNAL ERROR"
NVD6112W	"Missing value for keyword <%s>"
NVD6113E	"INCORRECT KEYWORD <%s>"
NVD6114E	"%s: INSUFFICIENT NUMBER OF PARAMETERS <%d>"
NVD6115E	"ONE OF THE MANDATORY INPUT PARAMETERS ARE EMPTY (to,from,mesgfile or message)"
NVD6116E	"%s: INVALID PARAMETERS PASSED"
NVD6117I	"%s Cache"
NVD6118E	"ERROR %s CACHE"
NVD6119I	"Cache %s"
NVD6120E	"%s: INVALID ARGUMENTS PASSED"
NVD6121I	"Option passed = <%s>"
NVD6122E	"OPTION MUST BE ENABLE OR DISABLE"
NVD6123E	"%s: %s"
NVD6124I	"%s Database"
NVD6125E	"ERROR %s DATABASE"
NVD6126I	"Database %s"
NVD6127E	"OPTION MUST BE LOCK OR UNLOCK"
NVD6128E	"%s: NO INPUT PARAMETERS"
NVD6129E	"NUMBER OF PARAMETERS IS <%d>, MUST BE 2."
NVD6130I	"Event log request: type = <%s>, message = <%s>"
NVD6131W	"Event type <%c> is invalid, Type assumed to be informational"
NVD6132E	"EVENT LOGGING REQUEST FAILED"
NVD6133E	"%s: VARIABLE ZUSERID NOT FOUND IN <%s> OBJECT"

Message Number	Log Message
NVD6134W	"NTSRVNAME variable not found in <%s>. Local server will be used"
NVD6135W	"ZOBJREQ variable not found in <%s>. NTGROUPS object will be used"
NVD6136I	"%s: %s"
NVD6137E	"%s: AN ACCESS VIOLATION HAS OCCURRED"
NVD6138W	"%s: %s"
NVD6139E	"ERROR RETRIEVING <%s> GROUPS <%s>"
NVD6140E	"%s: ERROR GETTING MANAGER TYPE"
NVD6141E	"%s: ERROR GETTING DATABASE STATUS"
NVD6142E	"%s: CANNOT PROMOTE TO LOCKED DATABASE"
NVD6143I	"%s: Entered"
NVD6144W	"No parameters passed in ZMETHOD object, will use defaults"
NVD6145E	"%s: ERROR OBTAINING DATABASE LOCATION"
NVD6146E	"%s: ERROR GETTING TRACE SETTINGS"
NVD6147E	"UNABLE TO FIND INBOUND OBJECT <%s>, TERMINATING"
NVD6148I	"%s: Exiting"
NVD6149E	"ERROR GETTING MAIL PATH"
NVD6150E	"%s: %s OBJECT = <%s>, HEAP = <%d>"
NVD6151E	"%s: Error opening <%s>"
NVD6152I	"%s: %s, object = <%s>, heap = <%i>. Notification for %s"
NVD6153W	"DATABASE IS LOCKED"
NVD6154E	"%s: OBJECT <%-s> NOT FOUND"
NVD6155E	"%s: UNABLE TO OBTAIN <%d> BYTES OF STORAGE FOR TEMPLATE BUFFER"
NVD6156E	"%s: ERROR GETTING TEMPLATE FOR <%-s>"
NVD6157E	"%s: UNABLE TO OBTAIN <%d> BYTES OF STORAGE"
NVD6158E	"%s: REQUIRED <%-s.%-s> NOT FOUND IN HEAP <%d>"
NVD6159E	"%s: CANNOT READ CLASS <%-s.%-s.%-s>"
NVD6160E	"%s: CANNOT READ INSTANCE <%-s.%-s.%-s.%-s>"

Message Number	Log Message
NVD6161E	"%s: SYNTAX ERROR IN <%-8.8s> LENGTH <%d> VALUE <%-0.*s>"
NVD6162E	"%s: CMRRTN rptptr->name <%-8.8s> type <%c>"
NVD6163E	"%s; CMRRTN rptptr->bufflen <%d> uselen <%d>"
NVD6164E	"%s: CANNOT FIND ENTRY FOR <%-8.8s> LENGTH <%d> VALUE <%-0.*s> IN <%-s.%-s.%-s.%-s>"
NVD6165I	"%s: Copied <%-0.*s> to <%-8.8s>"
NVD6166E	"%s: CANNOT DELETE DATABASE INSTANCE <%-s.%-s.%-s.%-s>"
NVD6167I	"%s: Deleted database instance <%-s.%-s.%-s.%-s>"
NVD6168E	"%s: FAILED TO WRITE INSTANCE <%-s.%-s.%-s.%-s>"
NVD6169I	"%s: %-s database instance <%-s.%-s.%-s.%-s>"
NVD6170I	"%s: Copied <%s> to <%-8.8s>"
NVD6171W	"%s: Field <%-8.8s> Not Found!"
NVD6172E	"CLASS NOT FOUND <%-s.%-s>"
NVD6173E	"EMPTY CLASS <%-s.%-s>"
NVD6174E	"ERROR LOCKING CLASS <%-s.%-s>"
NVD6175E	"ERROR READING CONTROL INFO <%s>"
NVD6176E	"ERROR READING TEMPLATE ENTRY."
NVD6177E	"ERROR ALLOCATING TEMPLATE SPACE <%d> BYTES."
NVD6178E	"INCOMPLETE TEMPLATE IN <%-s.%-s>"
NVD6179E	"ERROR ALLOCATING <%d> BYTES OF STORAGE FOR INSTANCE <%-s.%-s.%-s>"
NVD6180E	"ERROR LOCKING INSTANCE <%-s.%-s.%-s>"
NVD6181E	"ERROR READING INSTANCE <%-s.%-s.%-s>"
NVD6182E	"UNABLE TO OBTAIN MEMORY"
NVD6183E	"FAILED TO PROCESS THE INPUT OBJECT"
NVD6184E	"kvp_cleanup FAILED RELEASING THE LINK LIST"
NVD6185E	"kvp_findStop FAILED RELEASING STRUCTURE FOR DUPLICATE VARIABLES"
NVD6186E	"NO VCxxx VARIABLES WERE PROCESSED."

Message Number	Log Message
NVD6187E	"ERROR GETTING SQL DATA"
NVD6188W	"Default SQLTOUT = <%d> is used"
NVD6189E	"DYNAMIC SQL STATEMENT IS TOO BIG. MAXIMUM SIZE ALLOWED IS %i"
NVD6190E	"UNABLE TO OBTAIN SQL_column TYPE BY NAME"
NVD6191E	"FAILED TO PREPARE 'SELECT' STATEMENT"
NVD6192E	"FAILED TO EXECUTE 'SELECT' STATEMENT <%s>"
NVD6193E	"PROCESSING FAILED DUE TO SQL_GetInfo-error RC =<%i>"
NVD6194E	"ERROR EXECUTING %s FUNCTION."
NVD6195E	"ERROR EXECUTING edmmapi_put_var FOR <%s> OBJECT <%s> VARIABLE."
NVD6196I	"Calling EdmConnect with params: DSN=<%s>, USERID=<%s>, PWD=<%s>"
NVD6197E	"UNABLE TO FIND SOURCE OBJECT <%s>."
NVD6198I	"InfoValuePtr = <%s>"
NVD6199E	"PROCESSING FAILED DUE TO ERRORS ON DATA LOADING FROM HEAP <%i> OF <%i>"
6200-6299	General Method Messages
NVD6200I	"ExecuteUpdateForCurrentHeap() Before DuplicateSingleQuotes = <%s>"
NVD6201I	"ExecuteUpdateForCurrentHeap() After DuplicateSingleQuotes = <%s>"
NVD6202E	"NO UNIQUE COLUMNS FOUND FOR 'WHERE' CLAUSE. TERMINATING"
NVD6203W	"SQL-request: %s"
NVD6204E	"ERROR LOADING VARIABLE <%s> HEAP <%i> OBJECT <%s>. DEFAULT WILL BE USED"
NVD6205E	"FAILED TO PREPARE 'INSERT' STATEMENT"
NVD6206E	"FAILED TO READ DATA FROM OBJECT <%s> HEAP <%i> FOR 'INSERT' HEAP IGNORED"
NVD6207E	"FAILED TO EXECUTE 'INSERT' STATEMENT FOR OBJECT <%s> HEAP <%i>"
NVD6208E	"PROCESSING FAILED DUE TO SQL ERRORS. OBJECT <%s> HEAP <%i>"

Message Number	Log Message
NVD6209W	"The row requested for REPLACE does not exist in the database. INSERT will be executed. Heap <%i> of <%i>"
NVD6210E	"FAILED TO PREPARE 'INSERT' STATEMENT FOR OBJECT <%s> HEAP <%i>"
NVD6211E	"PROCESSING FAILED DUE TO SQL ERRORS. HEAP <%i> OF <%i>"
NVD6212E	"ERROR GETTING VARIABLE <%s>, HEAP = <%d>"
NVD6213E	"INTERNAL ERROR IN GETTING VARIABLE <%s> FROM LIST OF PARAMETERS"
NVD6214E	"%s: OBJECT NAME WAS NOT SUPPLIED"
NVD6215W	"Variable <%s> was not found"
NVD6216E	"REQUIRED VARIABLE <%s> NOT FOUND IN THE REQUEST OBJECT <%s>."
NVD6217E	"%s, DSN = <%s>, RC = <%d>, SQLState = <%s>"
NVD6218I	"%s: SQLDescribeCol(): pRoot->request->hstmt = %8X, ColumnNumber = %d, ColumnName = %s, sizeof(ColumnName) = %d, NameLength = %d, ColumnSize = %d, DecimalDigits = %d, Nullable = %d"
NVD6219E	"NOT ENOUGH INFO IN VC PAIR DEFINITION"
NVD6220E	"INCORRECT VC PAIR DEFINITION"
NVD6221I	"%s Starting Object: %8.8s"
NVD6222E	"%s VARIABLE ZUSERID NOT FOUND IN <%8.8s> OBJECT"
NVD6223E	"%s VARIABLE ZPWD NOT FOUND IN <%8.8s> OBJECT"
NVD6224E	"%s ZUSERID: <%8.8s> SIGNON FAILED"
NVD6225I	"%s ZUSERID: <%8.8s> Signon Verified"
NVD6226E	"%s TEMPLATE NOT FOUND FOR <%8.8s>"
NVD6227E	"%s - ERROR OPENING FILE %32.32s"
NVD6228E	"%s ERROR IN WRITING PREFIX FOR CLASS %s.%s.%s.%s"
NVD6229E	"%s ERROR IN WRITING CLASS HEADER %s.%s.%s.%s"
NVD6230E	"%s ERROR IN WRITING CLASS TEMPLATE %s.%s.%s.%s"
NVD6231E	"%s: HEAP NOT FOUND FOR <%8.8s>"
NVD6232E	"%s: NO PASSWORD ENTRY FOR USER <%s>"

Message Number	Log Message
NVD6233E	"%s: PASSWORD EXPIRED FOR USER <%s>"
NVD6234W	"%s: Password Will Expire In <%ld> Days For User <%s>"
NVD6235E	"%s ERROR CLOSING LOGN TOKEN"
NVD6236E	"%s ZUSERID: <%s> SIGNON FAILED: <%s>"
NVD6237E	"%s ZUSERID: <%s> SAF: <%s>"
NVD6238E	"ERROR: GetTokenInformation() RC <%ld> GETTING TOKEN INFO LENGTH"
NVD6239E	"ERROR: malloc(%lu) FAILED FOR TOKEN BUFFER"
NVD6240E	"ERROR: GetTokenInformation() RC <%ld> GETTING TOKEN INFO"
NVD6241I	"User <%s> belongs to the following groups:"
NVD6242I	"SID SidNameUse DomainName\\GroupName"
NVD6243I	"%08x %d %-16.16s %-s\\%-s"
NVD6244E	"ERROR: <%ld> GETTING ACCOUNT SID INFO"
NVD6245E	"NOT A VALID ACL. ERROR: <%ld>"
NVD6246E	"ERROR: <%ld> GETTING ACE ADDRESS."
NVD6247I	"No full access."
NVD6248E	"INVALID TIME \"%s\" PASSED TO EDMSLEEP METHOD - DEFAULTING TO 60 SECONDS"
NVD6249I	"Method going to sleep for %d seconds"
NVD6250I	"Sleep method woke up and is now terminating"
NVD6251E	"ERROR CREATING FILE <%s> FOR NOTIFY"
NVD6252E	"ERROR CREATING DOMAIN <%s.%s> FOR NOTIFY"
NVD6253E	"ERROR CREATING CLASS <%s.%s.%s> FOR NOTIFY"
NVD6254E	"ERROR LOCKING CLASS <%s.%s.%s.%s>"
NVD6255E	"ERROR UNLOCKING CLASS <%s.%s.%s.%s>"
NVD6256E	"UNABLE TO WRITE <%s> STATS, BECAUSE <DOMAIN %s DOES NOT EXIST>"
NVD6257E	"UNABLE TO USE <%s> STATS, BECAUSE <CLASS %s DOES NOT EXIST>"

Message Number	Log Message
NVD6258E	"UNABLE TO USE <%s> STATS, BECAUSE <CLASS INFO FOR <%s> IS NOT AVAILABLE>"
NVD6259E	"UNABLE TO WRITE <%s> STATS, BECAUSE <MEMORY ALLOCATION FAILURE FOR THE TEMPLATE BUFFER>"
NVD6260E	"UNABLE TO WRITE <%s> STATS, BECAUSE <CLASS %s IS NOT LOCKING>"
NVD6261E	"UNABLE TO WRITE <%s> STATS, BECAUSE <CLASS TEMPLATE IS NOT AVAILABLE>"
NVD6262E	"UNABLE TO WRITE <%s> STATS, BECAUSE <MEMORY ALLOCATION FAILURE FOR THE INSTANCE BUFFER>"
NVD6263E	"ERROR %s INSTANCE <%s.%s.%s.%s> WHILE STORING NOTIFY RESULTS"
NVD6264E	"UNABLE TO SCHEDULE REQUEST, BECAUSE <NOTIFY.RETRY.NOTIFY CLASS IS NOT FOUND>"
NVD6265E	"UNABLE TO ALLOCATE BUFFER FOR TEMPLATE OF <NOTIFY.RETRY.NOTIFY>"
NVD6266E	"UNABLE TO LOCK <NOTIFY.RETRY.NOTIFY CLASS>"
NVD6267E	"UNABLE TO READ TEMPLATE FOR <NOTIFY.RETRY.NOTIFY CLASS>"
NVD6268E	"UNABLE TO ALLOCATE BUFFER FOR INSTANCE OF <NOTIFY.RETRY.NOTIFY>"
NVD6269E	"%s PRE-PROMOTE OBJECT NOT FOUND"
NVD6270E	"%s - VARIABLE ZPRMOS NOT FOUND IN %s"
NVD6271E	"%s - SETUP INSTANCE CHECK FOR <%s.%s.%s.%s>"
NVD6272E	"%s - INSTANCE FOUND <%s.%s.%s.%s>"
NVD6273E	"%s - INSTANCE NOT FOUND <%s.%s.%s.%s>"
NVD6274E	"%s - INSTANCE CHECK FAILURE FOR <%s.%s.%s.%s>"
NVD6275E	"%s - VARIABLE %s NOT FOUND IN %s"
NVD6276E	"%s - ERROR SORTING PREPROMOTE OBJECT"
NVD6277E	"%s - ERROR GETTING %s"
NVD6278E	"%s - ERROR RESOLVING ZRSOURCE INSTANCES"
NVD6279E	"%s - ZRSOURCE OBJECT NOT FOUND"
NVD6280E	"%s - ERROR SORTING ZRSCOBJ"

Message Number	Log Message
NVD6281E	"%s - ERROR RESOLVING ZRSOURCE INSTANCES"
NVD6282E	"%s - VARIABLE %s NOT FOUND"
NVD6283E	"%s - AGENT LOCATION NOT FOUND"
NVD6284E	"%s - NO %s PROVIDED FOR HEAP %d"
NVD6285I	"%s - +++++ %s TIME %s +++++"
NVD6286I	"%s: Sending Shutdown Request to the Client Automation Configuration Server"
NVD6289E	"%s: ENDING WITH ERRORS"
NVD6290I	"%s: %s logons %s"
NVD6291I	"%s: PID = <%s>"
NVD6292I	"%s: Inbound object name = <%s>"
NVD6293I	"%s: Current object name = <%s>"
NVD6294I	"%s: Message = <%s>"
NVD6295I	"%s: Heaps = <%s>"
NVD6296I	"%s: Resolve objects = <%s>"
NVD6297I	"%s: files to send = <%s>"
NVD6298I	"%s: Method running = <%s>"
NVD6299I	"%s: Files sent = <%s>"
6300-6399	General Method Messages
NVD6300I	"%s: -- new value is <%i>"
NVD6301I	"%s: Flush Log Request issued from Console"
NVD6302I	"%s: Switch Log Request issued from Console"
NVD6303I	"%s: %s domain: <%s> class: <%s> instance: <%s> message: <%s> parent: <%s>"
NVD6304I	"%s: Simulation has started at <%s>"
NVD6305I	"%s: Put Template length - %d name - %s template - %s"
NVD6306I	"%s *%. *s"
NVD6307E	"%s: CANNOT UPDATE LOCKED DATABASE"

Message Number	Log Message
NVD6308E	"%s: INCORRECT NUMBER OF ARGUMENTS PASSED <%d>"
NVD6309E	"%s: CLASS ZBASE CAN NOT BE DELETED"
NVD6310E	"%s: CANNOT UPDATE THE DATABASE OF A CLIENT AUTOMATION CONFIGURATION SERVER TYPE SERVER"
NVD6311E	"%s: OBJECT DATABASE ACCESS ERROR"
NVD6312I	"%s: Class Deleted <%s>"
NVD6313E	"%s: CLASS NOT FOUND <%s>"
NVD6314E	"%s: INSTANCE NOT FOUND: <%s>"
NVD6315I	"%s: Instance Key: <%s>"
NVD6316I	"%s: Instance Deleted <%s>"
NVD6317E	"%s: ERROR DELETING INSTANCE <%s>"
NVD6318E	"%s: NO INSTANCES FOUND"
NVD6319E	"%s: MISSING OBJECT NAME"
NVD6320E	"%s: CANNOT RUN AMS UPDATE OPERATION ON LOCKED DATABASE"
NVD6321E	"%s: NO ARGUMENTS FOUND"
NVD6322E	"%s: CLASS <%s.%s.%s> NOT FOUND"
NVD6323E	"%s: INSTANCE <%s.%s.%s.%s> NOT FOUND"
NVD6324I	"%s: Class <%s.%s.%s> was found"
NVD6325I	"%s: Instance <%s.%s.%s.%s> was found"
NVD6326E	"%s: PROFILE ERROR - ZUSERID NOT FOUND"
NVD6327I	"%s: Get Profile Userid <%s>, From Object <%s>, To Object <%s>, Instance <%s>"
NVD6328I	"%s: Creating Object : <%s>"
NVD6329E	"%s: UNABLE TO USE <%s> STATS, BECAUSE CLASS <%s> DOES NOT EXIST"
NVD6330E	"%s: UNABLE TO USE <%s> STATS, BECAUSE CLASS INFO FOR <%s> IS NOT AVAILABLE"
NVD6331E	"%s: PROFILE ERROR - %s CLASS : <%s>"
NVD6332E	"%s: PROFILE ERROR - ALLOCATING HEAP STORAGE : <%s>"

Message Number	Log Message
NVD6333E	"%s: PROFILE ERROR - %s INSTANCE : <%s>"
NVD6334E	"%s: RODO ERROR (TEMPLATE) <%s>"
NVD6335E	"%s: NO MATCHING INSTANCES FOUND IN CLASS <%s>"
NVD6336E	"%s: CANNOT UPDATE THE DATABASE OF A SERVER"
NVD6337E	"%s: DATABASE IS LOCKED FOR SYNCHRONIZATION. NO UPDATES ALLOWED"
NVD6338E	"%s: CANNOT GET DOMAIN INFORMATION"
NVD6339I	"%s: Getting ZMETHOD ZMTHPARMS"
NVD6340I	"%s: Verifying ZMETHOD ZMTHPARMS"
NVD6341I	"%s: Initializing notify request structure"
NVD6342I	"%s: Creating notify request structure"
NVD6343E	"%s: ERROR CREATING REQUEST STRUCTURE"
NVD6344I	"%s: Storing notify request structure"
NVD6345E	"%s: ERROR STORING REQUEST STRUCTURE"
NVD6346I	"%s: Sending request to the queue"
NVD6347W	"%s: Unable to initiate notify handle, will try to continue"
NVD6348W	"%s: Unable to store notify results, will try to continue"
NVD6349E	"%s: NO OBJECT SPECIFIED -- REJECTED"
NVD6350E	"%s: COMPRESS OBJECT - OBJECT NOT FOUND: <%s>"
NVD6351W	"%s: Delete Object Instance - Object <%s> not found"
NVD6352W	"%s: Bad Heap Number <%d> requested for Deletion. Max Heap Number is <%d> for <%s>"
NVD6353E	"%s: SORT OBJECT - OBJECT NOT FOUND"
NVD6354I	"%s: Sort Object - Single Heap Object"
NVD6355E	"%s: NEW OBJECT <%-s> SAME AS OLD OBJECT <%-s>, COPY NOT DONE"
NVD6356E	"%s: FROM OBJECT <%-s> NOT FOUND"
NVD6357E	"%s: EMPTY OBJECT <%-s>, NO %s"
NVD6358E	"%s: NEW OBJECT <%-s> ALREADY EXISTS, COPY NOT DONE"

Message Number	Log Message
NVD6359E	"%s: UNABLE TO ALLOCATE <%d> BYTES FOR <%-s> %s BUFFER"
NVD6360E	"%s: TEMPLATE NOT FOUND FOR OBJECT <%-s>, RC <%d>"
NVD6361E	"%s: ERROR ON PUT <%-s> TEMPLATE, RC <%d>"
NVD6362E	"%s: HEAP <%d> NOT FOUND IN <%-s>, RC <%d>"
NVD6363E	"%s: ERROR ON PUT <%-s> HEAP <%d>, RC <%d>"
NVD6364E	"%s: DELETE VARIABLE - OBJECT NOT FOUND"
NVD6365E	"%s: PROMOTE TO ANY - OBJECT NOT FOUND"
NVD6366E	"%s: ERROR ALLOCATING %s STORAGE"
NVD6367E	"%s: ERROR LOCKING <%s>"
NVD6368E	"%s: ERROR GETTING %s"
NVD6369E	"%s: %s NOT FOUND IN HEAP <%d>"
NVD6370E	"%s: ERROR DELETING <%s>.<%s>.<%s>"
NVD6371E	"%s: ERROR CREATING <%s>.<%s>.<%s>"
NVD6372E	"%s: ERROR LOCKING <%s>.<%s>.<%s>"
NVD6373E	"%s: ERROR WRITING <%s>.<%s>.<%s>"
NVD6374E	"%s: ERROR UNLOCKING <%s>.<%s>.<%s>"
NVD6375E	"%s: ENTRY <%s>.<%s>.<%s> NOT CACHED: rc: <%d>"
NVD6376E	"%s: ERROR UNLOCKING <%s>"
NVD6377E	"%s: CLASS NOT FOUND <%s>.<%s>"
NVD6378E	"%s: EMPTY CLASS <%s>.<%s>"
NVD6379E	"%s: ERROR LOCKING CLASS <%s>.<%s>"
NVD6380E	"%s: ERROR READING CONTROL INFO <%s>"
NVD6381E	"%s: ERROR READING TEMPLATE ENTRY"
NVD6382E	"%s: ERROR ALLOCATING STORAGE FOR INSTANCE"
NVD6383E	"%s: ERROR READING <%s>.<%s>.<%s>"
NVD6384I	"%s: Processing object <%s>"
NVD6385E	"%s: WRITING DIRECTLY TO VSAM AUDIT DATABASE IS NOT SUPPORTED, OBJNAME <%s>"

Message Number	Log Message
NVD6386E	"%s: ERROR, %s NOT FOUND FOR OBJECT <%s>"
NVD6387E	"%s: ERROR OPENING FILE <%s>"
NVD6388E	"%s: Opening <%s>"
NVD6389E	"%s: ERROR OPENING FILE <%s> : ERRNO <%s>"
NVD6390E	"%s: ERROR WRITING <%d> BYTES TO FILE <%s>"
NVD6391E	"%s: chmod FAILED FOR <%s> ERRNO <%d>:<%s>"
NVD6392E	"%s: ERROR - NO DATA WRITTEN TO FILE <%s>"
NVD6393I	"%s: <%ld> bytes written to file <%s>"
NVD6394E	"%s: ERROR OPENING FILE <%s>, TYPE <%s>, MODE <%s>"
NVD6395I	"%s: Opening <%s> type <%s>"
NVD6396E	"%s: ERROR OPENING FILE <%s> TYPE <%s>:<%s>"
NVD6397E	"%s: INVALID %s NAME <%s> FILE <%s>"
NVD6398E	"%s: chown FAILURE: USER <%s> FILE <%s>"
NVD6399E	"%s: INVALID ZEDMTYPE <%s>"
6400-6499	General Method Messages
NVD6400E	"%s: ERROR <%d> ADDING FILE <%s>"
NVD6401E	"%s: ERROR <%d> ADDING DOMAIN <%s.%s>"
NVD6402E	"%s: ERROR <%d> CREATING CLASS <%s.%s.%s>"
NVD6403E	"%s: ERROR <%d> LOCKING CLASS <%s.%s.%s>"
NVD6404E	"%s: ERROR <%d> LOCKING INSTANCE <%s.%s.%s.%s>"
NVD6405E	"%s: ERROR WRITING <%d> BYTES TO INSTANCE <%s.%s.%s.%s>"
NVD6406E	"%s: ERROR WRITING <%ld> BYTES TO CLASS <%s.%s.%s>"
NVD6407E	"%s: ERROR - NO DATA WRITTEN TO CLASS <%s.%s.%s>"
NVD6408E	"%s: ERROR - NO DATA WRITTEN TO INSTANCE <%s.%s.%s.%s>"
NVD6409I	"%s: <%ld> bytes written to class <%s.%s.%s>"
NVD6410I	"%s: <%ld> bytes written to instance <%s.%s.%s.%s>"
NVD6411W	"%s: Object <%-s> was not found"

Message Number	Log Message
NVD6412I	"%s: From: <%s> To: <%s>"
NVD6413I	"%s: Write History F.D.C <%s.%s.%s>"
NVD6414E	"%s: UNABLE TO OBTAIN <%d> BYTES OF STORAGE FOR %s BUFFER"
NVD6415I	"%s: History T (%d) H (%d)"
NVD6416E	"%s: ERROR WRITING CLASS TEMPLATE FOR <%s.%s.%s>"
NVD6417E	"%s: OBJECT NOT FOUND <%s>"
NVD6418E	"%s: ERROR <%d> WRITING INSTANCE <%s.%s.%s.%s>"
NVD6419I	"%s: Instance <%s.%s.%s.%s> written from object <%s> heap <%ld>"
NVD6420I	"%s: ERROR <%d> UNLOCKING INSTANCE <%s.%s.%s.%s>"
NVD6421I	"%s: Wrote <%ld> instances to <%s.%s.%s>, size <%ld>, from <%s> heaps <%ld>"
NVD6422I	"%s: User: <%s> From: <%s> To: <%s>"
NVD6423E	"%s: UNABLE TO CREATE %s"
NVD6424I	"%s: Profile T (%d) H (%d)"
NVD6425I	"%s: Entered VarName = <%8.8s>, VarLenName = <%8.8s>"
NVD6426I	"%s: Wire pwd = <%s>"
NVD6427E	"%s: UNABLE TO OBTAIN <%ld> BYTE BUFFER FOR CLASS <%s.%s.%s.%s>"
NVD6428E	"%s: ERROR READING CONTROL INFO FOR CLASS <%s.%s.%s.%s>"
NVD6429I	"%s: DB pwd = <%s>, pwdLen = %d, rc = %d"
NVD6430E	"%s: OBJECT DATABASE ACCESS ERROR CLASS <%s> INSTANCE <%s>"
NVD6431E	"%s: OBJECT DATABASE ACCESS ERROR FOR CLASS <%s>"
NVD6432E	"%s: OBJECT DATABASE ACCESS ERROR FOR INSTANCE <%s>"
NVD6433E	"%s: UPDATED RESOURCE <%s>"
NVD6434E	"%s: INVALID ID IN INSTANCE NAME <%s>"
NVD6435I	"%s: Parent instance with object id <%s> could not be located for <%s.%s.%s>"
NVD6436E	"%s: CLASS NOT FOUND <%s.%s.%s>"

Message Number	Log Message
NVD6437E	"%s: ERROR READING INSTANCE <%s.%s.%s.%s>, BYTES READ <%d>"
NVD6438E	"%s: ERROR EXPECTED OBJECT ID <%12.12s>, ACTUAL OBJECT ID <%12.12s> IN <%s.%s.%s.%s>"
NVD6439E	"%s: UNABLE TO OBTAIN STORAGE FOR %s"
NVD6440E	"%s: UNABLE TO GET %s"
NVD6441E	"%s: PUT PROFILE - TEMPLATE OBJECT NOT FOUND"
NVD6442I	"%s: Original CLASS Profile T (%d) H (%d)"
NVD6443I	"%s: Class Size Changed New Profile T (%d) H (%d)"
NVD6444I	"%s: Class Content Changed New Profile T (%d) H (%d)"
NVD6445E	"%s: PROFILE - ERROR OPENING CLASS <%s.%s.%s>"
NVD6446E	"%s: ERROR IN WRITING NEW CLASS TEMPLATE"
NVD6447E	"%s: PUT PROFILE - OBJECT NOT FOUND"
NVD6448E	"%s: PROFILE - ERROR OPENING CLASS <%s.%s.%s>"
NVD6449I	"%s: %s Instance <%s.%s.%s.%s>"
NVD6450E	"%s: ERROR OPENING FILE <%s.%s.%s.%s>"
NVD6451I	"%s: Creating Instance from Class Record T (%d) H (%d)"
NVD6452I	"%s: Varname (%8.8s) To Dspl (%d) From Offset (%d) Len (%d)"
NVD6453I	"%s: Contents (%s)"
NVD6454E	"%s: ERROR IN WRITING INSTANCE DATA FOR <%s.%s.%s.%s>"
NVD6455E	"%s: PROFILE - ERROR OPENING INSTANCE <%s.%s.%s>"
NVD6456E	"%s: PROFILE - ERROR READING INSTANCE <%s.%s.%s>"
NVD6457I	"%s: Remapping Instance (%s) Old T (%d) H (%d), New T (%d) H (%d)"
NVD6458E	"%s: PROFILE - ERROR WRITING INSTANCE <%s.%s.%s>"
NVD6459E	"%s: VARIABLE LOG LIST - <%s> OBJECT NOT FOUND %s"
NVD6460I	"%s: ZVARLOG FOR OBJECT <%s>"
NVD6461I	"%s: VARIABLE LEN INSTANCE %d"
NVD6462I	"%s: %8.8s %03d %s"
NVD6463E	"%s: NOT DONE, OBJECT NOT FOUND"

Message Number	Log Message
NVD6464E	"%s: SUBSTITUTION BUFFER TOO SMALL "
NVD6465E	"%s: SUBSTITUTION ERROR"
NVD6466E	"%s: ITEM DOES NOT FIT IN LIST BUFFER"
NVD6467I	"%s: NO NAMES FOUND"
NVD6468E	"%s: NAME DOES NOT FIT IN BUFFER"
NVD6469E	"%s: ERROR GETTING HEAP FOR <%s>"
NVD6470I	"%s: exited, rc <%d>"
NVD6471D	"%s: %s"
NVD6472E	"%s: %s"
NVD6473A	"%s: %s"
NVD6474E	"%s: INSTANCE NAME <%s> EXCEEDS <%d> CHARACTERS"
NVD6475E	"%s: ERROR LOCKING INSTANCE <%s>"
NVD6476E	"%s: ERROR FINDING RESOURCE FOR INSTANCE <%s>"
NVD6477E	"%s: ERROR LOCKING RESOURCE <%s>"
NVD6478E	"%s: ERROR WRITING RESOURCE <%s>"
NVD6479I	"%s: ztcbg @ %08x"
NVD6480E	"%s: FINDFIRST/NEXTCLASS ERROR: <%s>"
NVD6481E	"%s: ERROR IN CONTROL RESOLUTION"
NVD6482W	"%s: Control Resolution - OBJECT [ZCONTROL] NOT FOUND"
NVD6483E	"%s: ERROR IN TREE RESOLUTION"
NVD6484E	"%s: TreeView - OBJECT [ZTREEOBJ] NOT FOUND"
NVD6485E	"%s: ERROR IN MakeTree RESOLUTION"
NVD6486E	"%s: TREE ERROR OBJECT <%s> NOT CREATED"
NVD6487I	"%s: TREE ERROR object <%s> created"
NVD6488E	"%s: ERROR IN KillTree"
NVD6489E	"%s: GET export_path FAILED"
NVD6490E	"%s: ERROR GETTING OBJECT INFO, ERROR=<%s>"

Message Number	Log Message
NVD6491E	"%s: UNKNOWN INSTANCE NAME, UPDOBJ FAILED, INSTANCE=<%s>"
NVD6492E	"%s: ERROR RETRIEVING INSTANCE DATA, ERROR=<%s>"
NVD6493E	"%s: ERROR IN INSTANCE: OBJECTID IS BLANK"
NVD6494I	"%s: UPDOBJ - getting class"
NVD6495E	"%s: OBJECT TIMESTAMPS WERE NOT EQUAL, RC=<%d>"
NVD6496E	"%s: ERROR LOCKING CLASS, ERROR=<%s>"
NVD6497E	"%s: ERROR RETRIEVING CLASS DATA, RC=<%d>"
NVD6498E	"%s: ENTRY # %d MISMATCH <%s>.<%s>.<%s>"
NVD6499E	"%s: DATABASE <%8.8s> <%d> BROWSER <%8.8s> <%d> <%x>"
6500-6599	General Method Messages
NVD6500E	"%s: RULES STARTING SYSTEM ACCESS LEVEL"
NVD6501E	"%s: ZRULES - NO ZADMIN FUNCTION PROVIDED"
NVD6502E	"%s: ZRULES - ALLOW: no processing for <%s>"
NVD6503E	"%s: ZRULES - NO ADMIN TYPE PROVIDED"
NVD6504I	"%s: ZRULES - NO ADMIN CLASS PROVIDED"
NVD6505E	"%s: ZRULES - NO ADMIN INSTANCE GIVEN"
NVD6506E	"%s: ZRULES - ERROR ALLOCATING STORAGE"
NVD6507E	"%s: LISTFILE - listrc <%d> Items <%lu> Length <%d> Addr <%x>"
NVD6508E	"%s: LISTDOMN - listrc <%d> Items <%lu> Length <%d>"
NVD6509E	"%s: LISTCLAS - listrc <%d> Items <%lu> Length <%d>"
NVD6510I	"%s: LISTINST - listrc <%d> Items <%lu> Length <%d>"
NVD6511I	"%s: ZRULES - No Rules Present - System default Access is (I)GNORE"
NVD6512I	"%s: ZRULES - No Rules Present - System default Access is (A)LLOW"
NVD6513I	"%s: ZRULES - No Rules Present - System default Access is (D)ENY"
NVD6514I	"%s: Rules Object <%s>"
NVD6515I	"%s: ZRULES - zvarsort rc <%d>, list @ <%08x>, item length <%d>, items <%d>"
NVD6516I	"%s: ZRULES - List item <%d> <%-0.*s>"

Message Number	Log Message
NVD6517I	"%s: Rules Ended RC (%d)"
NVD6518E	"%s: ERROR SETTING UP VARIABLES"
NVD6519E	"%s: ERROR INVALID TYPE PROVIDED <%c>"
NVD6520E	"%s: ERROR GETTING DIRECTORY, RC=<%d>"
NVD6521E	"%s: DUPLICATE INSTANCE NAME, ADDOBJ FAILED, INSTANCE=<%s>"
NVD6522I	"%s: ADDOBJ - getting class"
NVD6523E	"%s: ERROR GETTING %s FROM THE WIRE, RC=<%d>"
NVD6524E	"%s: ERROR LOCKING CLASS <%s.%s.%s>, ERROR IS <%s>"
NVD6525E	"%s: ERROR RETRIEVING CLASS DATA FOR <%s.%s.%s>, ERROR IS <%s>"
NVD6526E	"%s: ERROR UNLOCKING CLASS <%s.%s.%s>, ERROR IS <%s>"
NVD6527E	"%s: ENTRY # %d MISMATCH <%s>.<%s>.<%s>"
NVD6528E	"%s: DATABASE <%8.8s> <%d> BROWSER <%8.8s> <%d>"
NVD6529E	"%s: ERROR GETTING DOMAIN, ERROR IS <%s>"
NVD6530E	"%s: DUPLICATE CLASS NAME, ADDOBJ FAILED, INSTANCE=<%s>"
NVD6531E	"%s: ERROR RETRIEVING CLASS %s, RC=<%d>"
NVD6532E	"%s: ERROR CREATING NEW CLASS OBJECT, ERROR IS <%s>"
NVD6533E	"%s: ERROR CREATING NEW CLASS, ERROR IS <%s>"
NVD6534E	"%s: ERROR CREATING NEW BASE INSTANCE, RC=<%s>"
NVD6535E	"%s: ERROR UPDATING DATABASE CONTROL INFORMATION, RC=<%d>"
NVD6536E	"%s: MISSING %s NAME"
NVD6537E	"%s: INVALID (Blank) %s NAME"
NVD6538E	"%s: DOMAIN EXISTS"
NVD6539E	"%s: ERROR CREATING %s"
NVD6540E	"%s: ERROR WRITING ZBASE CLASS"
NVD6541E	"%s: CLASS DOES NOT EXIST, ADD INSTANCE FAILED"
NVD6542E	"%s: ERROR DISINHERITING RC=<%d>"

Message Number	Log Message
NVD6543E	"%s: ERROR CREATING INSTANCE <%s>"
NVD6544E	"%s: ERROR CREATING NEW INSTANCE, RC=<%d>"
NVD6545E	"%s: ERROR DELETING INSTANCE <%s>, RC=<%d>"
NVD6546E	"%s: ERROR UPDATING CLASS CONTROL, ERROR IS <%s>"
NVD6547E	"%s: %s NOT FOUND <%s>"
NVD6548I	"%s: Copy instance <%s> into <%s>"
NVD6549I	"%s: Check existence of <%s> class"
NVD6550E	"%s: INSTANCE <%s> ALREADY EXISTS"
NVD6551E	"%s: RENAMING TO THE SAME INSTANCE NAME <%s>"
NVD6552E	"%s: ERROR ALLOCATING <%d> BYTES OF STORAGE FOR %s <%s>"
NVD6553E	"%s: ERROR <%d> COPYING INSTANCE <%s> INTO <%s>"
NVD6554I	"%s: Successfully copied instance <%s> into <%s>"
NVD6555E	"%s: ERROR CREATING RESOURCE %s <%s>"
NVD6556E	"%s: ERROR LOCKING RESOURCE <%s>"
NVD6557E	"%s: ERROR READING RESOURCE PREFIX FOR <%s>"
NVD6558E	"%s: ERROR WRITING DATA FOR RESOURCE <%s>"
NVD6559E	"%s: SOURCE AND TARGET FILES DO NOT MATCH"
NVD6560E	"%s: TARGET CLASS ALREADY EXISTS, CLASS = <%s>"
NVD6561E	"%s: ERROR CREATING NEW CLASS OBJECT, ERROR IS <%s>"
NVD6562W	"%s: Issue Reading Source, Null instance error is <%s>, ignored"
NVD6563E	"%s: ERROR GETTING CLASS <%s>"
NVD6564E	"%s: CLASS SIZE <%d> vs. <%d> NOT EQUAL FOR CLASS <%s>"
NVD6565E	"%s: ERROR COPYING PACKAGE INSTANCE"
NVD6566E	"%s: CopyPackage FAILED TO COPY ALL PACKAGE INSTANCES, DELETING NEW INSTANCES"
NVD6567E	"%s: CopyPackage FAILED TO DELETE NEW INSTANCES"
NVD6568I	"%s: CopyPackage Successfully cleaned up new instances"
NVD6569E	"%s: ERROR GETTING ENTRY FOR PACKAGE INSTANCE"

Message Number	Log Message
NVD6570E	"%s: ERROR PARSING LIST ENTRY RC = <%d> <%s>"
NVD6571I	"%s: Copy Package Instance <%s> into <%s>"
NVD6572E	"%s: ERROR READING INSTANCE <%s>"
NVD6573I	"%s: Connections found for new instance <%s>"
NVD6574W	"%s: No Connections found for new instance <%s>"
NVD6575I	"%s: Successfully copied instance <%s> into <%s> <%0.*s> <%0.*s> <%0.*s>"
NVD6576I	"%s: Deleting instance <%s>"
NVD6577I	"%s: Instance Not Found <%s>, No need to delete"
NVD6578I	"%s: Instance Deleted <%s>"
NVD6579E	"%s: ERROR DELETING INSTANCE <%s>"
NVD6580E	"%s: COPYING TO THE SAME INSTANCE NAME <%s>"
NVD6581E	"%s: ERROR: INSTANCE <%s> HAS A 0 RESOURCE SIZE"
NVD6582E	"%s: INSTANCE <%s> DOES NOT EXIST"
NVD6583E	"%s: ERROR LOCKING INSTANCE, ERROR IS <%s>"
NVD6584E	"%s: UNABLE TO DELETE RESOURCE PRIOR TO COPYING NEW RESOURCE <%s> INTO <%s> RC = <%d>"
NVD6585E	"%s: ERROR <%d> COPYING RESOURCE FROM <%s> INTO <%s>"
NVD6586E	"%s: ERROR <%d> UPDATING INSTANCE FOR RESOURCE COPY FROM <%s> INTO <%s>"
NVD6587I	"%s: Successfully updated instance for resource copy from <%s> into <%s>"
NVD6588E	"%s: %s ERROR GETTING FUNCTION, RC <%d>, HEAP <%d> WILL BE SKIPPED"
NVD6589E	"%s: %s WRONG FUNCTION <%s>. HEAP <%d> WILL BE SKIPPED"
NVD6590E	"%s: %s TYPE NOT PROVIDED. HEAP <%d> WILL BE SKIPPED"
NVD6591E	"%s: %s INVALID TYPE <%c> PROVIDED, HEAP <%d> WILL BE SKIPPED"
NVD6592E	"%s: ERROR GETTING INSTANCE INFORMATION"
NVD6593E	"%s: ATTEMPT TO DELETE THE BASE INSTANCE"
NVD6594E	"%s: RULES DENIED ON %s DELOBJ"

Message Number	Log Message
NVD6595E	"%s: ERROR RETRIEVING MASKS, RC=<%d>"
NVD6596I	"%s: Checking Access before deletion"
NVD6597E	"%s: UNKNOWN INSTANCE NAME, <%s> FAILED, INSTANCE=<%s>"
NVD6598E	"%s: ERROR ON INSTANCE FIND, ERROR IS <%s>"
NVD6599E	"%s: ERROR GETTING INSTANCE DIRECTORY, RC=<%d>"
6600-6699	General Method Messages
NVD6600E	"%s: ERROR DELETING CLASS <%s.%s.%s> AND INSTANCES, RC <%d>"
NVD6601E	"%s: ERROR RETRIEVING INSTANCE DATA, ERROR IS <%s>"
NVD6602E	"%s: ASSOCIATED RESOURCE OBJECT: <%s>"
NVD6603E	"%s: BAD INSTANCE NAME, DELOBJ FAILED, INSTANCE=<%s>, ERROR IS <%s>"
NVD6604E	"%s: MULTI-RESOURCE DELETE, RESOURCE NOT FOUND IN DOMAIN: <%s>, CHECKING DEFAULT DOMAIN"
NVD6605I	"%s: Deleting Resource <%s.%s.%s.%s>"
NVD6606E	"%s: RESOURCE DELOBJ FAILED, PROBABLE DATALESS RESOURCE, RESOURCE ID: <%s>"
NVD6607I	"%s: Successfully deleted instance"
NVD6608E	"%s: ERROR GETTING DOMAIN INFORMATION, RC=<%d>"
NVD6609E	"%s: ERROR DELETING DOMAIN <%s>, RC=<%d>"
NVD6610E	"%s: ADMIN EXECUTE - OBJECT <ZADMIN> NOT FOUND"
NVD6611E	"%s: ERROR GETTING FUNCTION, RC=<%d>"
NVD6612E	"%s: INTERNAL ERROR: UNABLE TO ALLOCATE MEMORY"
NVD6613E	"%s: COMMAND ARGUMENT IS INCORRECT"
NVD6614E	"%s: ERROR RETRIEVING CLASS, ERROR IS <%s>"
NVD6615E	"%s: ERROR RETRIEVING INSTANCE, PLEASE VERIFY THE DATABASE. ERROR=<%s>"
NVD6616W	"%s: Problem getting %s Name from ADMIN Object, rc=<%d>. Setting to Default."
NVD6617E	"%s: ERROR GETTING THE USER CLASS, RC=<%d>"

Message Number	Log Message
NVD6618E	"%s: ERROR RETRIEVING INSTANCE= <%s> CLASS= <%s> DOMAIN= <%s>. PLEASE VERIFY DB AND CHECK INSTANCE SIZE"
NVD6619W	"%s: Unable to notify <%s>, because <no user info in PROFILE available>."
NVD6620W	"%s: Unable to notify user <%s>, because <PROFILE.%s.ZMASTER.OBJECT is not available>."
NVD6621E	"%s: ERROR ADDING <%s> VARIABLE"
NVD6622E	"%s: ERROR GETTING INSTANCE OBJECT LOCATION, RC=<%d>"
NVD6623I	"%s: Unknown Instance name, Instance=<%s>. Setting Instance name to Base Instance"
NVD6624E	"%s: ERROR GETTING CLASS, RC=<%d>"
NVD6625E	"%s: ERROR PUTTING %s ON THE WIRE, RC=<%d>"
NVD6626E	"%s: EMPTY INSTANCE <%s> FOUND PROCESSING IS TERMINATED"
NVD6627E	"%s: BASE INSTANCE NOT PRESENT, CLASS=<%s>"
NVD6628E	"%s: ERROR STORING CLASS %s, RC=<%d>"
NVD6629E	"%s: ERROR STORING SEQUENCE_SENSITIVE FLAG TYPE, RC=<%d>"
NVD6630E	"%s: NO CLASS RETRIEVED, ERROR IS <%s>"
NVD6631I	"%s: Putting Class"
NVD6632I	"%s: Buffer Size is <%d>"
NVD6633I	"%s: Putting Instance"
NVD6634E	"%s: UNABLE TO GET RESOURCE FILE NAME"
NVD6635E	"%s: UNABLE TO GET RESOURCE FILE INFORMATION FOR <%s>, ERROR IS <%s>"
NVD6636E	"%s: ERROR RETRIEVING RESOURCE DATA, ERROR IS <%s>"
NVD6637I	"%s: GETOBJ - putting resource %s"
NVD6638E	"%s: ERROR STORING VARIABLE, VARIABLE=<%8.8s>, OBJECT=<%8.8s>, RC=<%d>"
NVD6639E	"%s: ERROR GETTING %s NAME FROM ADMIN OBJECT, RC=<%d>. SETTING TO DEFAULT"
NVD6640I	"%s: No User Mask"
NVD6641I	"%s: Mask <%s> Length <%d>"

Message Number	Log Message
NVD6642E	"%s: ERROR %s <%s> DATA, ERROR IS <%s>"
NVD6643E	"%s: ERROR RETRIEVING BASE INSTANCE DATA, ERROR IS <%s>"
NVD6644E	"%s: ERROR GETTING DOMAIN DIRECTORY, RC <%d>"
NVD6645E	"%s: ERROR GETTING CLASS, ERROR IS <%s>"
NVD6646E	"%s: ERROR GETTING INSTANCE NAMES, ERROR IS <%s>"
NVD6647E	"%s: ERROR GETTING CLASS, ERROR <%d>"
NVD6648E	"%s: UNABLE TO RETRIEVE %s DATA FOR <%s>, IGNORED"
NVD6649I	"%s: Class Name <%s.%s.%s>, class type <%c>"
NVD6650E	"%s: INSTANCE DATA LENGTH READ OF <%d> BYTES IS LESS THAN EXPECTED LENGTH OF <%d> FOR INSTANCE <%s.%s.%s.%s>, IGNORED"
NVD6651E	"%s: INSTANCE DATA LENGTH READ OF <%d> BYTES IS GREATER THAN EXPECTED LENGTH OF <%d> FOR INSTANCE <%s.%s.%s.%s>, IGNORED"
NVD6652E	"%s: UNABLE TO RETRIEVE CLASS DATA FOR <%s.%s.%s>, IGNORED"
NVD6653E	"%s: INVALID REQUEST TYPE <%c> SPECIFIED IN LIST REQUEST FOR <%s.%s.%s>"
NVD6654I	"%s: Rulechk - rule <%s> name <%s> size <%d>"
NVD6655I	"%s: InstanceNameFilter - rule <%s> name <%s>"
NVD6656I	"%s: Building component rules filters for <%s.%s.%s>"
NVD6657I	"%s: Added component rules filter <%s> <%d> <%s>"
NVD6658I	"%s: Built <%lu> component rules filters for <%s.%s.%s>"
NVD6659I	"%s: Received from varsort, rc=<%d>"
NVD6660E	"%s: ERROR FROM edmmapi_put_var, OBJ=<%s>, VAR=<%s>, HEAP=<%d>"
NVD6661I	"%s: Sending Default Mask"
NVD6662I	"%s: heap count=<%d>"
NVD6663E	"%s: UNABLE TO DELETE OBJECT=<%s>"
NVD6664I	"%s: Object <%s> deleted"
NVD6665I	"%s: %s - getting class"

Message Number	Log Message
NVD6666I	"%s: Number of heaps %d"
NVD6667I	"%s; Getting instance"
NVD6668E	"%s: OBJECT <%s> NOT FOUND"
NVD6669E	"%s: BUILD PATCH ERROR, ATTRIBUTE <%0.8s.%0.8s> WAS NOT FOUND"
NVD6670E	"%s: ERROR CLEARING <%s> VARIABLE"
NVD6671I	"%s: PROMOTE started for <%s>"
NVD6672I	"%s: PROMOTE ended for <%s>, rc <%d>, duration <%1.3f> seconds"
NVD6673I	"%s: DPROMOTE started for <%s>"
NVD6674I	"%s: DPROMOTE ended for <%s>, rc <%d>, duration <%1.3f> seconds"
NVD6675W	"%s: Promote cannot retrieve instance name from ZADMINST, rc=<%d>, trying ZOBJNAME"
NVD6676W	"%s: Promote cannot retrieve instance name from ZOBJNAME, rc=<%d>"
NVD6677E	"%s: PROMOTE CANNOT RETRIEVE INSTANCE NAME, BOTH ZADMINST AND ZOBJNAME ARE UNDEFINED"
NVD6678I	"%s: ZPROMANY started for <%s>"
NVD6679I	"%s: ZPROMANY ended for <%s>, rc <%d>, duration <%1.3f> seconds"
NVD6680E	"%s: ERROR UPDATING INSTANCE VIA ZPROMANY, RC=<%d>"
NVD6681I	"%s: end ZPROMANY"
NVD6682E	"%s: ERROR RETRIEVING OBJECT ID, RC=<%d>"
NVD6683I	"%s: PROMOTE - Object ID=<%s>"
NVD6684E	"%s: ERROR GETTING CLASS INFO, RC=<%d>"
NVD6685I	"%s: PROMOTE - fdci <%s.%s.%s.%s>"
NVD6686I	"%s: PROMOTE - PromoteObj begin"
NVD6687E	"%s: ERROR GETTING RESOURCE DIRECTORY, RC=<%d>"
NVD6688I	"%s: PromoteObj started for <%s>"
NVD6689E	"%s: ERROR PROMOTING <%s.%s.%s.%s>, RC <%d>"
NVD6690E	"%s: ERROR WRITING %s DATA, ERROR IS <%s>"
NVD6691E	"%s: ERROR UNLOCKING %s, ERROR IS <%s>"

Message Number	Log Message
NVD6692E	"%s: ERROR GETTING <%ld> BYTE BUFFER FRO PROMOTE"
NVD6693E	"%s: ERROR LOCKING CLASS FOR <%s.%s.%s.%s>, ERROR IS <%s>"
NVD6694E	"%s: ERROR %s CLASS DATA FOR <%s.%s.%s.%s>, ERROR IS <%s>"
NVD6695E	"%s: ERROR UNLOCKING CLASS FOR <%s.%s.%s.%s>, ERROR IS <%s>"
NVD6696I	"%s: No Patches to be processed"
NVD6697I	"%s: %0.8s.BLDPATCH = <%s>, building PATCHES"
NVD6698I	"%s: %0.8s Building PATCHES for <%s> and <%s>"
NVD6699I	"%s: %0.8s.ZADMDINS = <%s>, no patches will be built for <%s.%s.%s.%s>"
6700-6799	General Method Messages
NVD6700I	"%s: ObjInfo <%s.%s.%s.%s>"
NVD6701I	"%s: PromotePkt <%s.%s.%s.%s>"
NVD6702I	"%s: Promoting New Resource <%s.%s.%s.%s>"
NVD6703E	"%s: PROMOTE ERROR VALIDATING RESOURCE <%s.%s.%s.%s>, <%s>"
NVD6704I	"%s: Re-Promoting Existing Resource <%s.%s.%s.%s>"
NVD6705I	"%s: Old Resource prefix <%8.8s><%8.8s><%8.8s><%32.32s>"
NVD6706E	"%s: PROMOTE ERROR, EXISTING RESOURCE PREFIX C.I <%-8.8s.%-32.32s> DOES NOT MATCH <%-8.8s.%-32.32s>"
NVD6707E	"%s: PROMOTE ERROR, EXISTING RESOURCE PREFIX SIZE <%lu> IS LESS THAN EXPECTED SIZE <%lu>"
NVD6708E	"%s: PROMOTE ERROR, EXISTING RESOURCE <%s.%s.%s.%s> WILL NOT BE REPLACED"
NVD6709I	"%s: New Resource prefix <%8.8s><%8.8s><%8.8s><%32.32s>"
NVD6710E	"%s: PROMOTE ERROR %s RESOURCE <%s.%s.%s.%s>, ERROR <%s>"
NVD6711E	"%s: PROMOTE RESTART ERROR READING <%s.%s.%s.%s>, ERROR <%s>"
NVD6712E	"%s: PROMOTE RESTART ERROR READING <%s.%s.%s.%s>, PREFIX BYTES READ <%lu>"
NVD6713E	"%s: PROMOTE RESTART NOT POSSIBLE FOR <%s.%s.%s.%s>"
NVD6714E	"%s: OLD RESOURCE PREFIX IS <%0.*s><%0.*s><%0.*s><%0.*s>"

Message Number	Log Message
NVD6715E	"%s: NEW RESOURCE PREFIX IS <%0.*s><%0.*s><%0.*s><%0.*s>"
NVD6716E	"%s: PROMOTE RESTART ERROR READING <%s.%s.%s.%s>, EOF OFFSET <%lu>, RESTART OFFSET <%lu>"
NVD6717E	"%s: PROMOTE ERROR, RESETTING RESOURCE <%s.%s.%s.%s> SIZE TO <%lu>, ERROR <%s>"
NVD6718E	"%s: PROMOTE ERROR ON MAP FILE %s, RESOURCE <%s.%s.%s.%s>, ERROR <%d>"
NVD6719I	"%s: File mapping handle <%08X>, view <%08X>, offset <%08X%08X>, bytes <%08X>"
NVD6720W	"%s: Promote data size is zero, rc <%d>"
NVD6721E	"%s: ERROR RECEIVING PROMOTE DATA FOR <%s.%s.%s.%s>, RC <%d>, LENGTH <%d>"
NVD6722I	"%s: Promoted <%lu> bytes of <%lu> in <%1.3f> seconds, current Kbps <%1.3f>"
NVD6723I	"%s: Promoted <%lu> bytes in <%1.3f> seconds, average Kbps <%1.3f>"
NVD6724E	"%s: PROMOTE ERROR ON UNMAP FILE %s, RESOURCE <%s.%s.%s.%s>, ERROR <%d>"
NVD6725I	"%s: Promote complete for <%s.%s.%s.%s>, size <%lu>, duration <%1.3f> seconds"
NVD6726W	"%s: Missing %s Variable"
NVD6727W	"%s: ZRSCDATE contains an invalid date <%s>"
NVD6728E	"%s: ERROR SETTING RESOURCE DATE TIME <%s> <%s>"
NVD6729E	"%s: <%s>.<%s>.<%s>.<%s>"
NVD6730I	"%s: Multiple Instance PROMANY - start"
NVD6731E	"%s: MULTIPLE INSTANCE PROMANY ERROR GETTING %s FROM THE WIRE, RC=<%d>"
NVD6732I	"%s: Multiple Instance PROMANY - getting instance"
NVD6733E	"%s: MULTIPLE INSTANCE PROMANY ERROR PUTTING %s TO THE WIRE, RC=<%d>"
NVD6734E	"%s: ERROR UPDATING INSTANCE VIA MULTIPLE INSTANCE PROMANY, RC=<%d>"
NVD6735E	"%s: BUILD PATCH ERROR, ATTRIBUTE <%0.8s.%0.8s> IS BLANK"

Message Number	Log Message
NVD6736E	"%s: ERROR UPDATING INSTANCE, RC=<%d>"
NVD6737E	"%s: ERROR UPDATING CLASS DATA, ERROR=<%s>"
NVD6738E	"%s: ERROR UNLOCKING CLASS, ERROR=<%s>"
NVD6739E	"%s: ERROR RETRIEVING CLASS DATA, ERROR=<%s>"
NVD6740E	"%s: ERROR RETRIEVING CLASS NAME, RC=<%d>"
NVD6741E	"%s: ERROR RETRIEVING CLASS TYPE, RC=<%d>"
NVD6742I	"%s: Template from wire with prefix from Class Record(_CLASS_):"
NVD6743I	"%s: Heap from wire (_BASE_INSTANCE_):"
NVD6744E	"%s: ERROR UPDATING CLASS FILES"
NVD6745E	"%s: ERROR WRITING ZBASE CLASS FOR DM UPDATE, ERROR=<%s>"
NVD6746I	"%s: Time/Date stamps are equal"
NVD6747I	"%s: Backup F.D.C.I = %s.%s.%s.%s"
NVD6748I	"%s: Update F.D.C.I = %s.%s.%s.%s"
NVD6749E	"%s: ERROR RETRIEVING OLD CLASS, RC=<%d>"
NVD6750I	"%s: Old Class:"
NVD6751I	"%s: New Class:"
NVD6752E	"%s: ERROR CREATING TEMPORARY DIRECTORY, RC=<%d>"
NVD6753E	"%s: ERROR RETRIEVING INSTANCE FROM CLASS=<%s>"
NVD6754E	"%s: ERROR RETRIEVING OLD INSTANCE, RC=<%d>"
NVD6755E	"%s: BAD FLAG VALUE - INTERNAL ERROR, FLAG=<%d>"
NVD6756E	"%s: ERROR UPDATING INSTANCE FROM CLASS=<%s>"
NVD6757I	"%s: Old Instance:"
NVD6758I	"%s: New Instance:"
NVD6759I	"%s: Update Class - Deleted Fields Detected"
NVD6760E	"%s: ERROR UPDATING NEW CLASS <%s.%s.%s>, RC=<%d>"
NVD6761I	"%s: PutObj of New Class <%s.%s.%s> Successful"
NVD6762E	"%s: ERROR RENAMING CLASS <%s> TO CLASS <%s>; ERROR <%s>"

Message Number	Log Message
NVD6763I	"%s: Rename of Class <%s> to Class <%s> Successful"
NVD6764E	"%s: ERROR DELETING OLD CLASS <%s.%s.%s>, ERROR <%s>"
NVD6765I	"%s: Delete of Old Class <%s> successful"
NVD6766I	"%s: Deleting Old Class <%s.%s.%s> Successful"
NVD6767E	"%s: ERROR SAVING NEW CLASS <%s.%s.%s>, ERROR <%s>"
NVD6768I	"%s: PutObj of New Class <%s.%s.%s> Successful"
NVD6769E	"%s: ERROR RETRIEVING <%s.%s.%s> INSTANCE DATA, ERROR <%s>"
NVD6770I	"%s; VEROBJ - Start"
NVD6771E	"%s: ERROR GETTING OBJECT INFORMATION, RC=<%d>"
NVD6772E	"%s: RULES DENIED ON %s VEROBJ"
NVD6773W	"%s: VEROBJ rc=[%d], Domain exists"
NVD6774E	"%s: UNKNOWN OBJECT, VEROBJ FAILED, RC=<%d>"
NVD6775I	"%s: Admin Method"
NVD6776I	"%s: %72.72s"
NVD6777E	"%s: UNABLE TO ALLOCATE DI CONTROL BLOCK"
NVD6778W	"%s: Unable to delete object <%8.8s>, rc =<%d>"
NVD6779I	"%s: Delete of object <%8.8s> successful"
NVD6780E	"%s: ERROR SETTING ZOBJID TO 000000000000, RC=<%d>"
NVD6781E	"%s: TYPE NOT PROVIDED - SETTING TO DEFAULT, RC=<%d>"
NVD6782E	"%s: ERROR GETTING FUNCTION, RC=<%d>"
NVD6783I	"%s: Processing Request <%s>"
NVD6784E	"%s: UNKNOWN ADMIN FUNCTION <%s>"
NVD6785I	"%s: Request=<%s>, DB Object=<%.8s.%.32s.%.8s.%.32s>"
NVD6786I	"%s: Request=<%s>, DB Object=<%.8s.%.32s.%.8s.%.32s>, DB Object To=<%.8s.%.32s.%.8s.%.32s>"
NVD6787I	"%s: Domain <%s.%s> was not found"
NVD6788I	"%s: Class <%s.%s.%s> was not found"
NVD6789I	"%s: Instance <%s.%s.%s.%s> was not found"

Message Number	Log Message
NVD6790W	"%s: Domain <%s.%s> was not found"
NVD6791W	"%s: Class <%s.%s.%s> was not found"
NVD6792W	"%s: Instance <%s.%s.%s.%s> was not found"
NVD6793E	"%s: DOMAIN <%s.%s> WAS NOT FOUND"
NVD6794E	"%s: CLASS <%s.%s.%s> WAS NOT FOUND"
NVD6795E	"%s: INSTANCE <%s.%s.%s.%s> WAS NOT FOUND"
6800-6899	General Method Messages
NVD6800W	"%s: Missing package name filter for component instance <%s.%s.%s.%s>"
NVD6801E	"%s: RULES DENY <%s> <%s> ACCESS TO INSTANCE <%s.%s.%s.%s>"
NVD6802E	"%s: RULES DENY <%s> <%s> ACCESS TO CLASS <%s.%s.%s>"
NVD6803E	"%s: RULES DENY <%s> <%s> ACCESS TO DOMAIN <%s.%s>"
NVD6804E	"%s: RULES DENY <%s> <%s> ACCESS TO FILE <%s.%s>"
NVD6805E	"%s: RULES DENY <%s> <%s> ACCESS TO PACKAGE <%s.%s.%s.%s>"
NVD6806E	"%s: RULES DENY <%s> <%s> ACCESS TO RESOURCE <%s.%s.%s.%s>"
NVD6807E	"%s: RULES DENY <%s> <%s> ACCESS TO UPDATE CLASS <%s.%s.%s.%s>"
NVD6808E	"%s: RULES DENY <%s> <%s> ACCESS TO <%s.%s.%s.%s>"
NVD6809E	"%s: PACKAGE RULES DENY <%s> <%s> ACCESS TO COMPONENT <%s.%s.%s.%s>"
NVD6810I	"%s: Package rules allow <%s> <%s> access to component <%s.%s.%s.%s>"
NVD6811I	"%s: Rules allow <%s> <%s> access to instance <%s.%s.%s.%s>"
NVD6812I	"%s: Rules allow <%s> <%s> access to class <%s.%s.%s>"
NVD6813I	"%s: Rules allow <%s> <%s> access to domain <%s.%s>"
NVD6814I	"%s: Rules allow <%s> <%s> access to file <%s.%s>"
NVD6815I	"%s: Rules allow <%s> <%s> access to package <%s.%s.%s.%s>"
NVD6816I	"%s: Rules allow <%s> <%s> access to resource <%s.%s.%s.%s>"
NVD6817I	"%s: Rules allow <%s> <%s> access to update class <%s.%s.%s.%s>"
NVD6818I	"%s: Rules allow <%s> <%s> access to <%s.%s.%s.%s>"

Message Number	Log Message
6900-6999	Package Consolidation
NVD6900I	"Package consolidation verification from <%s> to <%s> begins"
NVD6901I	"Package consolidation preview from <%s> to <%s> begins"
NVD6902I	"Package consolidation copy from <%s> to <%s> begins"
NVD6903I	"Pkgcons replacing existing package <%s.%s.%s.%s>"
NVD6904I	"Pkgcons resolving <%s> for generic <%s>"
NVD6905I	"Pkgcons resolving <%s>"
NVD6909I	"Pkgcons resolution complete for <%s>"
NVD6910I	"Pkgcons expression <%8.8s> <%c> <%s>"
NVD6911E	"PACKAGE CONSOLIDATION UNABLE TO PROCESS SUBSTITUTIONS IN EXPRESSION <%s>"
NVD6913E	"PACKAGE CONSOLIDATION FOUND INVALID OR UNSUPPORTED TYPE <%c> IN EXPRESSION <%s>"
NVD6914E	"PACKAGE CONSOLIDATION EXPRESSION MISMATCH WITH PARENT <%s>"
NVD6915I	"Pkgcons will substitute <%s> with <%s>"
NVD6916I	"Pkgcons has substituted <%s> with <%s>"
NVD6920I	"Pkgcons connection <%8.8s> <%c> <%s>"
NVD6921E	"PACKAGE CONSOLIDATION UNABLE TO PROCESS SUBSTITUTIONS IN CONNECTION <%s>"
NVD6923E	"PACKAGE CONSOLIDATION FOUND INVALID OR UNSUPPORTED TYPE <%c> IN CONNECTION <%s>"
NVD6925I	"Pkgcons will substitute <%s> with <%s>"
NVD6926I	"Pkgcons has substituted <%s> with <%s>"
NVD6930I	"Pkgcons method <%8.8s> <%c> <%s>"
NVD6933E	"PACKAGE CONSOLIDATION FOUND INVALID OR UNSUPPORTED TYPE <%c> IN METHOD <%s>"
NVD6934E	"PACKAGE CONSOLIDATION METHOD MISMATCH WITH PARENT <%s>"
NVD6935I	"Pkgcons will substitute <%s> with <%s>"
NVD6936I	"Pkgcons has substituted <%s> with <%s>"

Message Number	Log Message
NVD6940I	"Pkgcons noop <%8.8s> <%c> <%s>"
NVD6945I	"Pkgcons will substitute <%s> with <%s>"
NVD6946I	"Pkgcons has substituted <%s> with <%s>"
NVD6950I	"Pkgcons variable <%8.8s> <%c> <%s>"
NVD6953E	"PACKAGE CONSOLIDATION FOUND INVALID OR UNSUPPORTED TYPE <%c> IN VARIABLE <%8.8s>"
NVD6954E	"PACKAGE CONSOLIDATION VARIABLE DATA MISMATCH WITH PARENT <%s>"
NVD6955I	"Pkgcons will substitute <%s> with <%s>"
NVD6956I	"Pkgcons has substituted <%s> with <%s>"
NVD6960W	"Pkgcons found resource <%s.%s.%s.%s> already exists, trying new object ID"
NVD6961E	"PKGCONS ERROR, CANNOT LOCK (EXCLUSIVE) <%s.%s.%s.%s>"
NVD6962E	"PKGCONS ERROR, CANNOT LOCK (SHARE) <%s.%s.%s.%s>"
NVD6963E	"PKGCONS ERROR, CANNOT UNLOCK <%s.%s.%s.%s>"
NVD6964E	"PKGCONS ERROR, CANNOT READ <%d> BYTES FROM <%s.%s.%s.%s>"
NVD6965E	"PKGCONS ERROR, CANNOT WRITE <%d> BYTES TO <%s.%s.%s.%s>"
NVD6966E	"PKGCONS ERROR, CANNOT DECOMPRESS <%s.%s.%s.%s> <%s>"
NVD6967E	"PKGCONS ERROR, CANNOT COMPRESS <%s.%s.%s.%s> <%s>"
NVD6968E	"PKGCONS ERROR, CANNOT FIND <%s.%s.%s.%s>"
NVD6969E	"PKGCONS ERROR, COPY FAILED FROM <%s> TO <%s.%s.%s.%s>"
NVD6990I	"Package consolidation verification complete, rc <%d>"
NVD6991I	"Package consolidation preview complete, rc <%d>"
NVD6992I	"Package consolidation copy complete, rc <%d>"
NVD6998E	"PACKAGE CONSOLIDATION FAILURE, RC <%d>, DURATION <%1.3f> SECONDS"
NVD6999I	"Package consolidation successful, rc <%d>, duration <%1.3f> seconds"
7001-7080	Find
NVD7001E	"ERROR: FILE <%s> NOT FOUND"

Message Number	Log Message
NVD7002W	"Warning: File <%s> Not Found"
NVD7003E	"ERROR: DOMAIN <%s.%s> NOT FOUND"
NVD7004W	"Warning: Domain <%s.%s> Not Found"
NVD7005E	"ERROR: CLASS <%s.%s.%s> NOT FOUND"
NVD7006W	"Warning: Class <%s.%s.%s> Not Found"
NVD7007E	"ERROR: INSTANCE <%s.%s.%s.%s> NOT FOUND"
NVD7008W	"Warning: Instance <%s.%s.%s.%s> Not Found"
NVD7009E	"ERROR, CANNOT LOCK (EXCLUSIVE) FILE <%s>"
NVD7010E	"ERROR, CANNOT LOCK (EXCLUSIVE) DOMAIN <%s.%s>"
NVD7011E	"ERROR, CANNOT LOCK (EXCLUSIVE) CLASS <%s.%s.%s>"
NVD7012E	"ERROR, CANNOT LOCK (SHARE) CLASS <%s.%s.%s>"
NVD7013E	"ERROR, CANNOT LOCK (EXCLUSIVE) INSTANCE <%s.%s.%s.%s>"
NVD7014E	"ERROR, CANNOT LOCK (SHARE) INSTANCE <%s.%s.%s.%s>"
NVD7015E	"ERROR, CANNOT UNLOCK FILE <%s>"
NVD7016E	"ERROR, CANNOT UNLOCK DOMAIN <%s.%s>"
NVD7017E	"ERROR, CANNOT UNLOCK CLASS <%s.%s.%s>"
NVD7018E	"ERROR, CANNOT UNLOCK INSTANCE <%s.%s.%s.%s>"
NVD7019E	"ERROR, CANNOT READ <%d> BYTES FROM FILE <%s>"
NVD7020E	"ERROR, CANNOT READ <%d> BYTES FROM DOMAIN <%s.%s>"
NVD7021E	"ERROR, CANNOT READ <%d> BYTES FROM CLASS <%s.%s.%s>"
NVD7022E	"ERROR, CANNOT READ <%d> BYTES FROM INSTANCE <%s.%s.%s.%s>"
NVD7023E	"ERROR, CANNOT WRITE <%d> BYTES TO FILE <%s>"
NVD7024E	"ERROR, CANNOT WRITE <%d> BYTES TO DOMAIN <%s.%s>"
NVD7025E	"ERROR, CANNOT WRITE <%d> BYTES TO CLASS <%s.%s.%s>"
NVD7026E	"ERROR, CANNOT WRITE <%d> BYTES TO INSTANCE <%s.%s.%s.%s>"
NVD7027I	"Using Cached Class <%s.%s.%s>"
NVD7028I	"Using Cached Instance <%s.%s.%s.%s>"

Message Number	Log Message
NVD7029I	"Using Cached Base Instance <%s.%s.%s.%s>"
NVD7030E	"ERROR, CACHED INSTANCE UPDATE FOR <%s.%s.%s.%s> FAILED"
NVD7031E	"ERROR, CANNOT LOCK (EXCLUSIVE) RESOURCE <%s.%s.%s.%s>"
NVD7032E	"ERROR, CANNOT LOCK (SHARE) RESOURCE <%s.%s.%s.%s>"
NVD7033E	"ERROR, CANNOT UNLOCK RESOURCE <%s.%s.%s.%s>"
NVD7034E	"ERROR, CANNOT READ <%d> BYTES FROM RESOURCE <%s.%s.%s.%s>"
NVD7035E	"ERROR, CANNOT WRITE <%d> BYTES TO RESOURCE <%s.%s.%s.%s>"
NVD7036E	"ERROR, RESOURCE <%s.%s.%s.%s> NOT FOUND"
NVD7037W	"WARNING, Resource <%s.%s.%s.%s> Not Found"
NVD7038I	"Class DMN <%s> CLS <%s>"
NVD7039E	"ERROR PROCESSING DELETE: File <%s> Domain <%s> Class <%s> Instance <%s>"
NVD7040E	"ERROR <%d> ADDING FILE F<%s>"
NVD7041E	"ERROR <%d> ADDING DOMAIN D<%s>"
NVD7042E	"ERROR <%d> CREATING CLASS D<%s> C<%s>"
NVD7043E	"ERROR <%d> LOCKING FILE D<%s> C<%s> I<%s>"
NVD7044E	"ERROR DELETING DOMAIN <%s.%s>"
NVD7045E	"ERROR DELETING CLASS <%s.%s.%s>"
NVD7046E	"ERROR DELETING INSTANCE <%s.%s.%s.%s>"
NVD7047E	"ERROR DELETING RESOURCE <%s.%s.%s.%s>"
NVD7048E	"ERROR LOCKING CLASS <%s.%s.%s>, RC<%d>"
NVD7049E	"ERROR UNLOCKING CLASS <%s.%s.%s>, RC<%d>"
NVD7050E	"ERROR LOCKING OBJECT <%s.%s.%s.%s>, RC<%d>"
NVD7051E	"ERROR READING INSTANCE <%s.%s.%s.%s>, RC<%d>"
NVD7052E	"ERROR UNLOCKING OBJECT <%s.%s.%s.%s>, RC<%d>"
NVD7053I	"Instance <%s.%s.%s.%s> not cached"
NVD7054I	"Class <%s.%s.%s.%s> not cached"

Message Number	Log Message
NVD7055I	"Base Instance <%s.%s.%s.%s> not cached"
NVD7056E	"ERROR, UNABLE TO WRITE <%s> STATS, BECAUSE <%s %s DOES NOT EXIST>"
NVD7057E	"ERROR, UNABLE TO WRITE <%s> STATS, BECAUSE <Class %s IS NOT LOCKING>"
NVD7058E	"ERROR, UNABLE TO WRITE <%s> STATS, BECAUSE <Class TEMPLATE IS NOT AVAILABLE>"
NVD7059E	"ERROR, UNABLE TO WRITE <%s> RETRY, BECAUSE <Class %s NOT FOUND>"
NVD7060E	"ERROR READING CLASS HEADER (%d BYTES) OF CLASS <%s.%s.%s>. (RC = <%d>)"
NVD7061E	"ERROR, DB ERROR <%s> WAS DISCOVERED WHILE PROCESSING <%s.%s> CLASS. CLIENT AUTOMATION CONFIGURATION SERVER MAY SHUTDOWN!"
NVD7062E	"ERROR, SHUTTING DOWN CLIENT AUTOMATION CONFIGURATION SERVER BECAUSE OF DATABASE ERROR!"
NVD7063E	"ERROR, INVALID CLASS SIZE <%s.%s.%s>"
NVD7064E	"ERROR READING HEADER RECORD (%d BYTES) OF CLASS <%s.%s.%s>. (RC = <%d>)"
NVD7065E	"ERROR UNLOCKING CLASS TEMPLATE <%s.%s.%s>, RC<%d>"
NVD7066E	"ERROR, UNABLE TO LOCK INSTANCE <%s.%s.%s.%s>, RC <%d>"
NVD7067E	"ERROR, UNABLE TO WRITE INSTANCE <%s.%s.%s.%s>, RC<%d>"
NVD7068E	"ERROR, UNABLE TO UNLOCK INSTANCE <%s.%s.%s.%s>, RC<%d>"
NVD7069E	"ERROR READING CLASS TEMPLATE <%s.%s.%s>, RC <%d>"
7081-7099	Invalid Name
NVD7081E	"ERROR, INVALID INSTANCE NAME <%s>"
NVD7082E	"ERROR, INVALID CLASS NAME <%s>"
NVD7083E	"ERROR, INVALID DOMAIN NAME <%s>"
NVD7084E	"ERROR, INVALID FILE NAME <%s>"
NVD7085E	"ERROR, INVALID INSTANCE <%s.%s.%s.%s> SIZE <%d> TOO SMALL"
7090-7099	Database Path Errors
NVD7090E	"ERROR, DATABASE PATH <%s> IS UNAVAILABLE, REASON <%s>"

Message Number	Log Message
NVD7091E	"ERROR, DATABASE HAS BEEN HARDLOCKED"
NVD7092E	"ERROR, MONITORING FOR DATABASE PATH ACCESSIBILITY TO BE RESTORED"
NVD7093E	"ERROR, DATABASE PATH ACCESSIBILITY HAS NOT BEEN RESTORED, MANAGER WILL BE SHUT DOWN"
NVD7094I	"Database Path Accessibility has been restored, Database Status is <%s>"
7100-7199	Database Prefix
NVD7101E	"ERROR, UNABLE TO GET CONTROL INFORMATION FOR INSTANCE <%s.%s.%s.%s>"
NVD7102E	"ERROR, UNABLE TO GET CONTROL INFORMATION FOR CLASS <%s.%s.%s>"
NVD7103E	"ERROR, UNABLE TO GET CONTROL INFORMATION FOR DOMAIN <%s.%s>"
NVD7104E	"ERROR, UNABLE TO GET CONTROL INFORMATION FOR FILE <%s>"
NVD7105E	"ERROR, UNABLE TO UPDATE CONTROL INFORMATION FOR INSTANCE <%s.%s.%s.%s>"
NVD7106E	"ERROR, UNABLE TO UPDATE CONTROL INFORMATION FOR CLASS <%s.%s.%s>"
NVD7107E	"ERROR, UNABLE TO UPDATE CONTROL INFORMATION FOR DOMAIN <%s.%s>"
NVD7108E	"ERROR, UNABLE TO UPDATE CONTROL INFORMATION FOR FILE <%s>"
NVD7109E	"ERROR, INVALID CONTROL INFORMATION FOR INSTANCE <%s.%s.%s.%s>"
NVD7110E	"ERROR, INVALID CONTROL INFORMATION FOR CLASS <%s.%s.%s>"
NVD7111E	"ERROR, INVALID CONTROL INFORMATION FOR DOMAIN <%s.%s>"
NVD7112E	"ERROR, INVALID CONTROL INFORMATION FOR FILE <%s>"
NVD7113I	"UPDATE_FILE_CONTROL_INFO started"
NVD7114I	"UPDATE_FILE_CONTROL_INFO ended"
NVD7115E	"UNABLE TO CREATE RETRY DOMAIN"
NVD7116E	"UNABLE TO CREATE <NOTIFY.RETRY.NOTIFY> Class"
NVD7117E	"ERROR WRITING <RETRY> Class PREFIX"

Message Number	Log Message
NVD7118E	"ERROR WRITING <RETRY> Class HEADER %s"
NVD7119E	"UNABLE TO CREATE CLASS <%s.%s.%s>"
NVD7120I	"Class <%s.%s.%s> prefix dates updated successfully"
7200-7299	Control
NVD7201I	"SOFTLOCKED! (exclusive)"
NVD7202I	"SOFTLOCKED! shared count=<%d>"
NVD7203E	"ERROR: UNABLE TO OBTAIN TIMESTAMP"
NVD7204W	"Warning, Timed out waiting for ADMIN to end"
NVD7205W	"UNLOCKED! (DMA_TIMEOUT)"
NVD7206I	"SOFTLOCKED! shared count=<%d> (DMA_TIMEOUT)"
NVD7207I	"HARDLOCKED!"
NVD7208I	"SOFTLOCKED! <%s> (hardlock timed out)"
NVD7209I	"UNLOCKED!"
NVD7210W	"Warning, UNLOCK request with count=0 ignored"
NVD7211E	"ERROR, UNABLE TO UPDATE CONTROL BUFFER, RC = <%d>"
NVD7212E	"ERROR, UNABLE TO ADD CONTROL RECORD"
NVD7213E	"UNABLE, TO LOCK CONTROL FILE"
NVD7214E	"ERROR, UNABLE TO READ CONTROL ENTRY SIZE <%d> BYTES READ <%d>"
NVD7215E	"UNKNOWN CONTROL UPDATE TYPE <%s>"
NVD7216E	"ERROR LOCKING <RETRY> Class"
NVD7217E	"ERROR LOCKING CLASS <%s.%s.%s>"
NVD7218I	"Pending HARDLOCK!"
NVD7219E	"ERROR, CLASS <%s.%s.%s> LOCK ERROR WHILE UPDATING CLASS PREFIX"
NVD7220E	"ERROR, CLASS <%s.%s.%s> WRITE ERROR WHILE UPDATING CLASS PREFIX"
NVD7221E	"ERROR, CLASS <%s.%s.%s> UNLOCK ERROR WHILE UPDATING CLASS PREFIX"

Message Number	Log Message
7300-7399	DMA
NVD7300I	"Starting <%s>"
NVD7301E	"VARGET ERROR DMAFUNC NOT FOUND FOR <%s>"
NVD7302I	"Processing Client Automation Distributed Configuration Server Function <%s>"
NVD7303E	"ERROR, INCONSISTENT LOCK STATE/COUNT: STATE=%c COUNT=%d"
NVD7304W	"Could not reset Client Automation Distributed Configuration Server timeout=<%d> secs"
NVD7305I	"Ending <%s>"
NVD7306I	"SoftLock Starting"
NVD7307I	"Creating DMA_Handle"
NVD7308E	"THE DMAHandle <%s> ALREADY EXISTS. INTERNAL ERROR"
NVD7309I	"New DMAHandle created: <%s>"
NVD7310E	"FAILED TO CREATE NEW DMAHandle: <%s>"
NVD7311I	"Restarting Client Automation Distributed Configuration Server staging for handle = <%s>"
NVD7312I	"Starting Client Automation Distributed Configuration Server staging for handle = <%s>"
NVD7313E	"%s <%s>: NOT FOUND IN STAGING AREA <%s>"
NVD7314E	"%s <%s>: STAGE FILE <%s> DELETE FAILED"
NVD7315E	"%s <%s> FROM STAGING AREA <%s>: DIRECTORY DELETE FAILED"
NVD7316I	"Removed Client Automation Distributed Configuration Server Stage Handle -- <%s> from staging area <%s>"
NVD7317E	"ERROR, CLIENT AUTOMATION DISTRIBUTED CONFIGURATION SERVER REQUEST FOR STANDALONE CLIENT AUTOMATION CONFIGURATION SERVER"
NVD7318E	"CLIENT AUTOMATION DISTRIBUTED CONFIGURATION SERVER REQUEST RC=<%d> MSG=<%s>"
NVD7319I	"Create Stage Filename <%s>"
NVD7320I	"Create Stage Filename CRC <%s>"
NVD7321E	"ERROR GETTING STAGE VARIABLES"

Message Number	Log Message
NVD7322E	"ERROR, UNABLE TO CREATE STAGE FILENAME"
NVD7323E	"ERROR WRITING <%d> BYTES TO STAGE OBJECT <%s>"
NVD7324E	"CLIENT AUTOMATION DISTRIBUTED CONFIGURATION SERVER PUTOBJECT STAGE ERROR"
NVD7325I	"DMAStage Successfully ended"
NVD7326E	"ERROR, UNABLE TO WRITE STAGE FILE"
NVD7327E	"ERROR PROCESSING DUPLICATES"
NVD7328E	"ERROR PROCESSING DELETES"
NVD7329E	"ERROR PROCESSING ADDS"
NVD7330W	"Warning, <%8.8s> marked duplicate"
NVD7331I	"ACTION %6.6s"
NVD7332I	"FILETYPE %8.8s"
NVD7333I	"FILENAME %8.8s"
NVD7334I	"DOMAIN %32.32s"
NVD7335I	"CLASS %8.8s"
NVD7336I	"INSTANCE %32.32s"
NVD7337I	"STAGE_FILENAME %8.8s"
NVD7338I	"DMA_BASE <%s>"
NVD7339I	"STAGE_LOC <%s>"
NVD7340I	"FILENAME <%s>"
NVD7341I	"DOMAIN <%s>"
NVD7342I	"CLASS <%s>"
NVD7343I	"INSTANCE <%s>"
NVD7344I	"STAGE_FILENAME <%s>"
NVD7345I	"TYPE <%s>"
NVD7346I	"FILE SIZE <%d>"
NVD7347W	"Warning, Staging Duplicate File: <%s>"
NVD7348E	"ERROR, UNABLE TO OPEN STAGE FILE FOR <%s>"

Message Number	Log Message
NVD7349I	"No DELETES to be PROCESSED"
NVD7350I	"No more DELETES to be PROCESSED"
NVD7351I	"No ADDS to be PROCESSED"
NVD7352I	"No more ADDS to be PROCESSED"
NVD7353E	"ERROR GETTING STAGE OBJECT: FILE <%s> DOMAIN <%s> CLASS <%s> INSTANCE <%s>"
NVD7354E	"ERROR UPDATING DATABASE: FILE <%s> DOMAIN <%s> CLASS <%s> INSTANCE <%s>"
NVD7355E	"ERROR UPDATING CONTROL FILE: FILE <%s> DOMAIN <%s> CLASS <%s> INSTANCE <%s>"
NVD7356E	"ERROR, DELETE FAILED FOR STAGEFILE <%s>"
NVD7357E	"TREECRC FAILED RC <%d>"
NVD7358E	"GetFileCRC <%s> FAILED RC <%d>"
NVD7359I	"Starting TREECRC"
NVD7360I	"Total CRC <%8IX>"
NVD7361I	"Domain <%s> not found, no CRC"
NVD7362E	"%s BASE INSTANCE NOT FOUND, CAN'T LOCK FOR CRC"
NVD7363I	"Locking domain <%s>"
NVD7364I	"Domain <%s> using cached total CRC=<%s>"
NVD7365E	"DOMAIN <%s> CACHED TOTAL CRC NOT FOUND"
NVD7366E	"UNABLE TO COMPRESS CRC FILE <%s>, RC=<%d>"
NVD7367E	"UNABLE TO CREATE TOTAL.CRC FILE <%s>"
NVD7368I	"Domain <%s> Total CRC: <%s>"
NVD7369I	"Unable to create CRC subdir: <%s>"
NVD7370I	"Domain <%s> simultaneous syncs with conflicting CRCLEVEL"
NVD7371I	"Ending <%s>, rc = <%d>"
NVD7372I	"Resource <%s.%s.%s> read failed"
NVD7373I	"Unable to create temp file <%s>"
NVD7374I	"Unable to write temp file <%s>"

Message Number	Log Message
NVD7375E	"UNABLE TO DECOMPRESS %s, rc=%d"
NVD7376I	"Cache Disabled"
NVD7377I	"Cache Enabled"
NVD7378E	"<%s> CRC LOCK ATTEMPT TIMED OUT"
NVD7379I	"Client Automation Distributed Configuration Server request no error RC=<%d> MSG=<%s>"
NVD7380I	"%s, Request <%c>, F.D.C.I <%s.%s.%s.%s>, OID <%s>, old OID <%s>"
7400-7599	Object ID
NVD7400I	"Initialize parms: global <%08X>, caller type <%d>, p_logmsg <%08X>"
NVD7402I	"Object ID cache handle allocated <%d> bytes at <%08X>"
NVD7403E	"UNABLE TO ALLOCATE <%d> BYTES OF STORAGE FOR OID CACHE HANDLE"
NVD7406I	"Attaching shared memory ID <%d>"
NVD7407E	"ERROR, NULL POINTER RETURNED BY zsys_attachSharedMem <%d>"
NVD7409E	"ERROR, OBJECT ID CACHE IS FULL "
NVD7410I	"Cleanup oid_handle at <%08X>"
NVD7411E	"ERROR <%d> RETURNED BY zsys_lockMutexSemaphore <%d>"
NVD7412I	"Detaching shared memory ID <%d> at <%08X>"
NVD7413E	"ERROR <%d> RETURNED BY zsys_detachSharedMem <%d>"
NVD7414I	"Removing shared memory ID <%d>"
NVD7415E	"ERROR <%d> RETURNED BY zsys_removeSharedMem <%d>"
NVD7416E	"ERROR <%d> RETURNED BY zsys_unlockMutexSemaphore <%d>"
NVD7417E	"ERROR <%d> RETURNED BY zsys_removeMutexSemaphore <%d>"
NVD7418I	"Object ID cache handle free <%d> bytes at <%08X>"
NVD7419E	"ERROR, INVALID OBJECT ID <%0.12s>"
NVD7420I	"Getting DMA counts from <%s.%s.%s.%s>"
NVD7421E	"ERROR <%d> LOCKING CLASS <%s.%s.%s.%s>"
NVD7422E	"ERROR <%d> READING CLASS <%s.%s.%s.%s>"

Message Number	Log Message
NVD7423E	"ERROR <%d> UNLOCKING CLASS <%s.%s.%s.%s>"
NVD7424E	"UNABLE TO OBTAIN DMA CLASS CONTROL INFORMATION FROM <%s.%s.%s.%s>"
NVD7425I	"DMA counts from <%s.%s.%s.%s>, Domains <%lu>, Classes <%lu>, Instances <%lu>"
NVD7426I	"Total counts for <%s>, Domains <%lu>, Classes <%lu>, Instances <%lu>"
NVD7427E	"ERROR <%d> READING DOMAINS FOR FILE <%s>"
NVD7428I	"Loaded <%lu> class object id's for domain <%s.%s>"
NVD7429E	"ERROR <%d> CLOSING DOMAIN FIND HANDLE FOR <%s>"
NVD7430E	"ERROR <%d> READING CLASSES FOR DOMAIN <%s.%s>"
NVD7431E	"ERROR <%d> READING CLASS PREFIX FOR <%s.%s.%s.%s>"
NVD7432E	"ERROR READING CLASS PREFIX FOR <%s.%s.%s.%s>, SIZE READ <%d>, EXPECTED <%d>"
NVD7433I	"Loaded <%lu> instance object IDs for class <%s.%s.%s>"
NVD7434E	"ERROR <%d> CLOSING CLASS FIND HANDLE FOR <%s.%s>"
NVD7435E	"ERROR <%d> READING INSTANCES FOR CLASS <%s.%s.%s>"
NVD7436E	"ERROR <%d> READING INSTANCE PREFIX FOR <%s.%s.%s.%s>"
NVD7437E	"ERROR READING INSTANCE PREFIX FOR <%s.%s.%s.%s>, SIZE READ <%d>, EXPECTED <%d>"
NVD7438I	"Loaded oid <%0.12s> for <%s.%s.%s.%s>"
NVD7439E	"ERROR <%d> CLOSING INSTANCE FIND HANDLE FOR <%s.%s.%s>"
NVD7440E	"ERROR, INVALID OBJECT ID <%0.12s> IN <%s.%s.%s.%s>"
NVD7441I	"Database total counts do not match DMA counts of: domains <%lu>, classes <%lu>, instances <%lu>"
NVD7442E	"ERROR, DUPLICATE OBJECT ID <%0.12s>, OCCURS <%d> TIMES"
NVD7443E	"ERROR, DUPLICATE OBJECT ID <%0.12s> FOUND WHILE ADDING <%s.%s.%s.%s>"
NVD7444E	"ERROR, DATABASE CONTAINS <%lu> DUPLICATE OBJECT IDS"
NVD7445W	"Object ID cache load from the database has been disabled"
NVD7446W	"Object ID cache structures have changed since the last checkpoint"

Message Number	Log Message
NVD7447W	"Unable to add object ID <%s> to object ID cache during restart reorganization"
NVD7448I	"Object ID cache restart reorganization in progress"
NVD7449I	"Object ID cache cold start request in progress"
NVD7450W	"Object ID cache restart checkpoint file <%s> was not found"
NVD7451W	"Object ID cache restart in progress"
NVD7452E	"ERROR OPENING OBJECT ID CACHE CHECKPOINT FILE <%s>, <%s>"
NVD7453E	"ERROR READING OBJECT ID CACHE CHECKPOINT FILE <%s>, <%s>"
NVD7454E	"ERROR CLOSING OBJECT ID CACHE CHECKPOINT FILE <%s>, <%s>"
NVD7455I	"Object ID cache checkpoint last DB update date/time <%8.8s> <%8.8s>, domains <%lu>, classes <%lu>, instances <%lu>"
NVD7456I	"Object ID cache DMA ZBASE last DB update date/time <%8.8s> <%8.8s>, domains <%lu>, classes <%lu>, instances <%lu>"
NVD7457W	"Database has been altered since last object ID cache checkpoint"
NVD7458W	"Object ID cache restart from checkpoint is not possible, object ID cache rebuild in-progress"
NVD7459I	"Object ID cache restart completed successfully, DB last updated on <%8.8s> at <%8.8s>"
NVD7460I	"Object ID cache checkpoint not required, DB last updated on <%8.8s> at <%8.8s>"
NVD7461I	"Object ID cache checkpoint in progress, DB last updated on <%s> at <%s>"
NVD7462E	"ERROR OPENING OBJECT ID CACHE CHECKPOINT FILE <%s>, <%s>"
NVD7463E	"ERROR WRITING OBJECT ID CACHE CHECKPOINT FILE <%s>, <%s>"
NVD7464E	"ERROR CLOSING OBJECT ID CACHE CHECKPOINT FILE <%s>, <%s>"
NVD7468E	"OBJECT ID CACHE CHECKPOINT FAILED"
NVD7469I	"Object ID cache checkpoint completed successfully"
NVD7470D	"%s, rc <%d>, <%12.12s> fatal error"
NVD7471D	"%s, rc <%d>, <%12.12s> was created"
NVD7472D	"%s, rc <%d>, <%12.12s> was not created"
NVD7473D	"%s, rc <%d>, <%12.12s> was found"

Message Number	Log Message
NVD7474D	"%s, rc <%d>, <%12.12s> was not found"
NVD7475D	"%s, rc <%d>, <%12.12s> was added"
NVD7476D	"%s, rc <%d>, <%12.12s> was not added, already exists"
NVD7477D	"%s, rc <%d>, <%12.12s> was deleted"
NVD7478D	"%s, rc <%d>, <%12.12s> was not deleted, was not found"
NVD7479D	"%s, rc <%d>, <%12.12s> was found with commit pending"
NVD7480D	"%s, rc <%d>, <%12.12s> was not found with commit pending"
NVD7481D	"%s, rc <%d>, <%12.12s> was added with commit pending"
NVD7482D	"%s, rc <%d>, <%12.12s> was not added, already exists"
NVD7483D	"%s, rc <%d>, <%12.12s> was found with commit pending and committed"
NVD7484D	"%s, rc <%d>, <%12.12s> was not found with commit pending, not committed"
NVD7485D	"%s, rc <%d>, <%12.12s> was found with commit pending and deleted"
NVD7486D	"%s, rc <%d>, <%12.12s> was not found with commit pending, not deleted"
NVD7487D	"%s, rc <%d>, <%lu> entries committed for task <%X>"
NVD7488D	"%s, rc <%d>, <%lu> entries deleted for task <%X>"
NVD7489W	"Spare message"
NVD7490W	"DMA control class <%s.%s.%s.%s> does not exist"
NVD7491I	"Determining actual DB domain, class, and instance counts for file <%s>"
NVD7492I	"Class <%s.%s.%s> contains <%lu> instances"
NVD7493I	"Domain <%s.%s> contains <%lu> classes"
NVD7494I	"Actual DB counts for file <%s>, Domains <%lu>, Classes <%lu>, Instances <%lu>"
NVD7495W	"Find error <%s> while processing Instance <%s.%s.%s.%s>"
NVD7496W	"Find error <%s> while processing Class <%s.%s.%s>"
NVD7497W	"Find error <%s> while processing Domain <%s.%s>"
NVD7498W	"Total DB find errors encountered processing file <%s>, Domain errors <%lu>, Class errors <%lu>, Instance errors <%lu>"
NVD7499W	"Spare message"

Message Number	Log Message
NVD7500I	"Object ID Cache Statistics"
NVD7501I	"Total size <%lu>, Prefix size <%lu>, Entry size <%lu>"
NVD7502I	"Total entries <%lu>, Used entries <%lu>, Active handles <%lu>"
NVD7503I	"Found <%lu>, Not found <%lu>, Adds <%lu>, Deletes <%lu>"
NVD7504I	"Commits <%lu>, Dups <%lu>, Hashed <%lu>, Rehashes <%lu>"
NVD7505I	"DMA Last Update Date/Time <%8.8s> <%8.8s>"
NVD7506I	"DMA Domains <%lu>, Classes <%lu>, Instances <%lu>"
NVD7507I	"Spare message"
NVD7508E	"ERROR, FAILED TO DELETE OBJECT ID <%0.12s> WHILE DELETING INSTANCE <%s.%s.%s.%s>"
NVD7509E	"ERROR, FAILED TO READ OBJECT ID FOR CLASS <%s.%s.%s>"
NVD7510E	"ERROR, FAILED TO DELETE OBJECT ID <%0.12s> WHILE DELETING CLASS <%s.%s.%s>"
NVD7511E	"ERROR, FAILED TO ADD OBJECT ID <%0.12s> WHILE ADDING CLASS <%s.%s.%s>"
NVD7512E	"ERROR, FAILED TO ADD OBJECT ID <%0.12s> WHILE ADDING INSTANCE <%s.%s.%s.%s>"
7900-7999	ZVSAM
NVD7900I	"%s%s: %s rc=%d line=%d fdci=<%s,%s,%s,%s, %d>"
NVD7901I	"%s%s: %s rc=%d line=%d"
NVD7902I	"%s%s: %s rc=%d line=%d dci=<%s> <%s> <%s> type=%c seq=%ld"
NVD7903I	"%s"
NVD7904E	"%s"
8000-8099	Errors
NVD8001E	"ERROR STARTING METHOD <%s>"
NVD8002E	"ERROR, INVALID PRAMETERS PASSED TO METHOD <%s>"
NVD8003E	"ERROR, METHOD <%s> FAILED"
NVD8004E	"ERROR, UNABLE TO GET TRACE FLAGS"
NVD8005E	"MISSING PROGRAM NAME"

Message Number	Log Message
NVD8006E	"MISSING PROGRAM TYPE"
NVD8007E	"MISSING PROGRAM PARMS"
NVD8008E	"PROMOTE FAILED, FLUSHING COMMUNICATIONS BUFFER"
NVD8009E	"CLIENT AUTOMATION CONFIGURATION SERVER METHOD FAILURE PGM: <%s> RC: <%d>, MAX RC ALLOWED <%d>"
NVD8010E	"METHOD <%s> SET OBJECT RESOLUTION ABORT CODE"
NVD8011E	"METHOD <%s> SET OBJECT RESOLUTION STOP CODE"
NVD8012E	"ERROR RECV. ACK FROM METHOD"
NVD8013E	"REXX METHOD <%s> TIMEOUT DUE TO STORAGE SHORTAGE"
NVD8014E	"REXX METHOD <%s>: ERR EXIT ENQ FAILURE RC <%d>"
NVD8015E	"ERROR LOADING THE REXX INTERPRETER...RC IS <%d>"
NVD8016E	"REXX INTERPRETER ERROR...RC IS <%d>"
NVD8017E	"SUBCOM REGISTRATION FOR EDMLINK FAILED RC=<%d>"
NVD8018E	"EXIT REGISTRATION FOR EDMSAY FAILED RC=<%d>"
NVD8019E	"EDMLINK HOST CMD ENV SETUP FAILED"
NVD8020E	"REXX ENVIROMENT SET UP COMPLETED WITH ERRORS <%d>"
NVD8021E	"FUNCTION DEREG FOR <%s> FAILED RC=<%d>"
NVD8022E	"FUNCTION DEREG FOR EDMLINK FAILED RC=<%d>"
NVD8023E	"EXIT DEREG FOR EDMSAY FAILED RC=<%d>"
NVD8024E	"REXX HOST CMD ENV CLEANUP COMPLETED WITH ERRORS <%d>"
NVD8025E	"INCORRECT #args FOR EDMRESO. USE: EDMRESO (RESOLVE STRING)"
NVD8026E	"ARGUMENT RESOLVE STRING HAS INVALID LENGTH...: <%d>"
NVD8027E	"INCORRECT #args TO EDMGETV. USE: EDMGETV (object,variable,heapnumber,resolveflag)"
NVD8028E	"ARGUMENT OBJECT NAME HAS INVALID LENGTH: <%d>"
NVD8029E	"ARGUMENT VARIABLE NAME HAS INVALID LENGTH: <%d>"
NVD8030E	"ARGUMENT OBJECT HEAP HAS INVALID LENGTH: <%d>"
NVD8031E	"ARGUMENT RESOLVEFLAG HAS INVALID LENGTH: <%d>"

Message Number	Log Message
NVD8032E	"VARIABLE <%s> WHICH IS <%d> LONG CANNOT BE STORED IN REXX"
NVD8033E	"INVALID LENGTH FOR ZOBJID <%d>"
NVD8034E	"ERROR IN REXX_SET: RC=<%d> SHVRET=<%d> NAME=<%s> VALUE=<%s>"
NVD8035E	"INCORRECT #args TO EDMSETV. USE: EDMSETV (object,variable,value,heapnumber)"
NVD8036E	"ARGUMENT VARIABLE VALUE HAS INVALID LENGTH: <%d>"
NVD8037E	"INCORRECT #args TO EDMSET. USE: EDMSET (object,heapnumber)"
NVD8038E	"ERROR IN REXX_GETNEXT: SHVRET=<%d> NAME=<%. *s> VALUE=<%. *s>"
NVD8039E	"ERROR REXX_RESET: RC=<%d> SHVRET=<%d> NAME=<%s> VALUE=<%s>"
NVD8040E	"%s: OSDYNALLOC FAILED, RETURN CODE <%ld>, REASON CODE <%08X>"
NVD8041E	"%s: REXX METHOD: ERR EXIT ENQ FAILURE RC <%d>"
NVD8042E	"INTERNAL ERROR - UNKNOWN TYPE <%d> IN ZCVT ACCESS TABLE FOR <%s>"
NVD8043E	"INTERNAL ERROR - UNKNOWN TYPE <%d> IN ZTCBG ACCESS TABLE FOR <%s>"
NVD8044E	"COULD NOT RESET SESSION TIMEOUT ATTRIBUTE <%s> INVALID"
NVD8045E	"FUNCTION REGISTRATION FOR <%s> FAILED RC <%d>"
NVD8046E	"%s: OSDYNALLOC FAILURE, PARM STRING PASSED IN WAS <%s>"
NVD8047I	"ZEDMLINK passed a blank command string <%s>"
NVD8048I	"Invalid number of arguments passed to ZSIMRESO"
NVD8049I	"usage: EDMLINK('ZSIMRESO FILE,DOMAIN,CLASS,INSTANCE,MESSAGE')"
NVD8050I	"EDMLINK Error in varput of ZMTHPRMS"
NVD8051D	"TCL: %s"
NVD8052I	"TCL: %s"
NVD8053W	"TCL: %s"
NVD8054E	"TCL: %s"

Message Number	Log Message
NVD8055E	"TCL: %s"
NVD8056I	"TCL cmd <%s> arg#%d(%-8.8s): <%s>"
NVD8057I	"TCL cmd <%s> arg#%d: <%s>"
NVD8058E	"TCL CMD <%s> ERROR RESULT: <%s>"
NVD8059I	"TCL cmd <%s> result: <%s>"
NVD8060E	"UNABLE TO FIND REQUESTED TCL METHOD <%s>"
NVD8061I	"TCL method <%s> executing from the <%s> directory"
NVD8062I	"TCL method completed, rc=%ld"
NVD8063E	"TCL INTERPRETER NOT CREATED"
NVD8064W	"init.tcl not found"
NVD8065E	"TCL METHOD <%s> RETURNED NON-NUMERIC RESULT <%s>"
NVD8066I	"TCL method <%s> returned rc=%d"
NVD8067E	"TCL METHOD <%s> FAILED: RESULT=<%s>, TRACEBACK FOLLOWS:"
NVD8068I	"instg verb=%s"
NVD8069E	"%s"
NVD8070E	"TCL CMD: %s"
NVD8071I	"heap# %d will be implicitly added to obj %s"
NVD8072I	"using current heap# %d of obj %s"
NVD8073I	"current heap# set =%d of obj %s"
NVD8074I	"heap# %d explicitly added to obj %s"
NVD8075I	"getvar(nosub) %s.%s = <%s>"
NVD8076I	"getvar(sub) %s.%s = <%s>"
NVD8077I	"EDMLINK rc=%d"
NVD8078I	"TCL tracing disabled"
NVD8079I	"resolution rc=%d"
NVD8080I	"setvar %s.%s = <%s>"
NVD8081I	"TCL cmd <%s>: NO args"

Message Number	Log Message
NVD8082E	"COULD NOT FIND %s.%s"
NVD8083E	"INTERNAL ERROR - UNKNOWN TYPE <%d> IN %s ACCESS TABLE FOR <%s>"
NVD8084I	"Value of %s.<%s> is <%s>"
NVD8085I	"REXX EXPR: <%s>"
NVD8086I	"TCL EXPR: <%s>"
NVD8087I	"TCL EVAL: <%s>"
NVD8088I	"RULE %d: <%s>"
NVD8089I	"TCL cmd <%s> result: <%d>"
NVD8090I	"TCL regexp compilation failed"
8100-8199	Informational
NVD8101I	"ZMTHSVR Object: <%s>"
NVD8102I	"ZMTHSVR Program: <%s>"
NVD8103I	"ZMTHSVR Program: <%s>, TYPE: <%s>"
NVD8104I	"ZMTHSVR C Method Completed RC: <%d>"
NVD8105I	"Client Automation Configuration Server Method RC Reset for Pgm: <%s> RC: <%d>, Max RC Allowed <%d>"
NVD8106I	"Ignoring Client Automation Configuration Server Method Return Code Pgm: <%s> RC: <%d>"
NVD8107W	"Unable to find requested REXX method <%s>"
NVD8108I	"start_rexx zsrc <%ld>"
NVD8109I	"REXX Method completed, rc = <%ld>"
NVD8110I	"Method Completed, rc = <%d>"
NVD8111I	"Remote METHOD field <%s>"
NVD8112I	"Dormant METHOD field <%s>"
NVD8113W	"REXX Method <%s> delayed due to storage shortage"
NVD8114W	"REXX Method <%s> with parms <%s> not executed, MGR_TRACE REXXOFF = YES"
NVD8115I	"REXX Method <%s> with parms <%s> started at <%2.2d:%2.2d:%2.2d:%3.3d>"

Message Number	Log Message
NVD8116I	"%s: REXX Method <%s> doing REXX Environment Ownership Check"
NVD8117I	"%s: REXX Method <%s>: RC of <%d>, ENQ for ZICB in use, using Local ZICB"
NVD8118I	"About to execute REXXStart()...."
NVD8119I	"REXX Method <%s> ended at <%2.2d:%2.2d:%2.2d:%3.3d>, rc<%d>"
NVD8120I	"REXX Version number is <%s>"
NVD8121W	"Setup of REXX dll failed, rc = <%d> - ok if not running REXX as a dll"
NVD8122A	"REXX search environment variable <%s>"
NVD8123I	"REXX Environment set up completed successfully"
NVD8124I	"Cmd Environment EDMLINK could not be deleted"
NVD8125I	"REXX host cmd env cleanup completed successfully"
NVD8126I	"%s"
NVD8127W	"Limit of <%i> LOG LINES reached for the REXX METHOD, No more messages will be allowed"
NVD8128I	"Resolution string passed = <%s>"
NVD8129I	"Object Name passed = <%s>"
NVD8130I	"Variable Name passed = <%s>"
NVD8131I	"Heap number was not specified, current assumed"
NVD8132I	"Heap number passed = <%d>"
NVD8133I	"Resolve flag was not specified, no substitution assumed"
NVD8134I	"Resolve flag passed <%s>"
NVD8135W	"Variable <%8.8s> not found in <%8.8s>"
NVD8136I	"Object <%8.8s> was not found"
NVD8137I	"Object name will be...: <%s>"
NVD8138I	"Variable <%s> is <%d> long and is at offset <%d>"
NVD8139I	"Returning <%s>"
NVD8140I	"ZOBJPID Variable Not found in <%8.8s>"
NVD8141I	"Object not found - <%s>"

Message Number	Log Message
NVD8142W	"Skipping Variable <%s> which is (%d) long cannot be stored in REXX"
NVD8143I	"Variable Value passed = <%s>"
NVD8144I	"%s: Variable will be updated...: <%.8s> length=<%d> Data=<%. *s>"
NVD8145I	"Incorrect call to EDMLINK. Use: EDMLINK (command)"
NVD8146I	"EDMLINK <%s>"
NVD8147I	"Incorrect call to EDMMALLO. Use: EDMMALLO (parms)"
NVD8148I	"Incorrect call to EDMMALLO. No \"DSN=\" in parms"
NVD8149I	"%s: Parms <%s>"
NVD8150I	"%s: osdynalloc parms <%s>"
NVD8151I	"ddname <%s> assigned"
NVD8152I	"Incorrect call to EDMMDALO. Use: EDMMDALO (DDN=ddname)"
NVD8153I	"%s: <%s> freed"
NVD8154I	"%s: REXX Method doing REXX Environment Ownership Check"
NVD8155I	"%s: REXX Method: RC of <%d> ENQ for ZICB in use, using local ZICB"
NVD8156W	"%s: Truncation in REXX_GETNEXT: SHVRET=<%d> NAME=<%. *s> VALUE=<%. *s>"
NVD8157W	"%s: Attribute name <%. *s>, length <%d>, is greater than max allowed <%d>, attribute cannot be set"
NVD8158I	"REXX Expression <%s> started at <%2.2d:%2.2d:%2.2d:%3.3d>"
NVD8159I	"REXX Expression <%s> completed at <%2.2d:%2.2d:%2.2d:%3.3d>"
NVD8160W	"Request to set Soft Task Limit to <%d>"
NVD8161W	"Reset Soft Task Limit to <%d>"
NVD8162W	"Could not find ZCVT variable of <%s>"
NVD8163W	"Value of ZCVT.<%s> is <%s>"
NVD8164W	"Attempting to get unknown ZTCBG variable of <%s>"
NVD8165E	"Set Ignore Expression Flag ON!"
NVD8166E	"Set Ignore Expression Flag OFF!"
NVD8167W	"Attempting to set unknown ZTCBG variable of <%s>"

Message Number	Log Message
NVD8168W	"Value of ZTCBG.<%s> is <%s>"
NVD8169W	"Unable to reset timeout, CliBlk unavailable"
NVD8170W	"Could not reset Session Timeout to <%d> seconds"
NVD8171I	"Session Timeout reset to <%d> seconds"
NVD8172I	"ZOBJID Value = <%s>"
NVD8173A	"REXX search environment variable value <%s>"
NVD8174W	"%s: irxctl(IRXCTL_NOBREAK) failed, me, rc <%d>"
NVD8175I	"REXX Method <%s> executing from the <%s> directory"
NVD8176I	"Secondary method <%s> causes OCB rename of primary method <%s>"
NVD8177E	"%s: ERROR SETTING <%8.8s> IN <%s>"
8200-8299	Expressions
NVD8201E	"ERROR <%d> IN REXX EXPRESSION <%s>"
NVD8202E	"REXX EXPRESSION ERROR IN <%s.%s.%s.%s>"
NVD8203I	"Expression <%s> Len <%d>"
NVD8204I	"Result <%s> Len <%d>"
NVD8205I	"Verify expression started:"
NVD8206E	"ERROR IN EXPRESSION: <%s>"
NVD8207I	"Check expression values started:"
NVD8208I	"Expression setting resolution <%s> code"
NVD8209I	"Expression message <%s>"
NVD8210I	"Expression restype <%s>"
NVD8211I	"Expression changing message from <%s> to <%s>"
8300-8399	ZNFY
NVD8301E	"PARMS VALUE MISSING OR INCORRECT"
NVD8302I	"LU6.2 Notify Manager name is <%s>"
NVD8303E	"ACBNAME MISSING, NONE PROVIDED OR DEFAULTED, UNABLE TO CONTINUE"
NVD8304E	"ZNFY6MGR <%s> ENDING DUE TO ERRORS"

Message Number	Log Message
NVD8305I	"%s Diagnostic address: APICB = %08x, APOCB = %08x, DESTBLK = %08x"
NVD8306I	"%s Diagnostic address: Notify ACB = %08x"
NVD8307E	"%s: MESSAGE QUEUE WAS NOT INITIALIZED, UNABLE TO CONTINUE"
NVD8308E	"ZNFYTMGR <%s> ENDING DUE TO ERRORS"
NVD8309E	"%s: ERROR DURING OPEN, RC = <%d>"
NVD8310I	"ZNFYTMGR <%s> ending"
NVD8311E	"%s: Before call to msgrcv"
NVD8312I	"%s: After call to msgrcv"
NVD8313I	"%s: Shutdown request received"
NVD8314E	"%s: UNABLE TO RECEIVE MSG, BECAUSE OF INTERNAL ERROR"
NVD8315I	"ZNFY6MGR <%s> ending"
NVD8316I	"Signal received: <%d>. PID is <%d>"
NVD8317I	"%s has entered Notify request processing"
NVD8318E	"%s: ACB NOT INITIALIZED PROPERLY. REQUEST IS IGNORED"
NVD8319I	"%s: Request was removed or skipped by start method"
NVD8320I	"%s is now exiting Notify request processing"
NVD8321E	"%s: NO LUNAME SUPPLIED, UNABLE TO PROCESS NOTIFY REQUEST"
NVD8322W	"%s: No TPNAME supplied using default of <%s>"
NVD8323W	"%s: No LOGMODE supplied using default of <%s>"
NVD8324I	"%s: Calling parms: Luname=<%s> Logmode=<%s> Tpname=<%s> Acbname=<%s>"
NVD8325I	"%s: Calling parms: Luname=<%s> Logmode=<%s> Tpname=<%s>"
NVD8326E	"%s: ALLOCATION ERROR FOR <%s - %s>"
NVD8327E	"%s: ERROR SETTING TP NAME FOR <%s - %s>"
NVD8328E	"%s: ERROR SETTING PLU NAME FOR <%s> - <%s>"
NVD8329I	"%s: LU6.2 Notify now in progress for <%s>"
NVD8330E	"%s: ERROR SETTING MODE NAME FOR <%s - %s>"
NVD8331E	"%s: CONNECTION ERROR FOR <%s - %s>"

Message Number	Log Message
NVD8332E	"%s: CONNECTION OPEN ERROR FOR <%s - %s>"
NVD8333I	"%s: Entering ZNFY6STA processing"
NVD8334I	"%s: Entering ZNFY6END processing"
NVD8335I	"%s: On Entry-> RC = <%d>, ntfyrnum = <%d>, ntfyrmax = <%d>"
NVD8336I	"%s: Exiting ZNFY6END processing"
NVD8337I	"%s: Entering open notify ACB process"
NVD8338I	"%s: Exiting open notify ACB process"
NVD8339E	"%s: APISTARTLOGON ERROR - <%s>"
NVD8340I	"%s: Entering close notify ACB process"
NVD8341E	"%s: CLOSE ACB (DISCONNECT) ERROR FOR <%s - %s>"
NVD8342I	"%s: Exiting close notify ACB process"
NVD8343I	"%s: Entering notify comm cleanup process"
NVD8344I	"%s: Exiting notify comm cleanup process"
NVD8345I	"%s: Entering read notify response process"
NVD8346E	"LU6.2 NOTIFY MANAGER <%s>, RECEIVE ERROR OCCURRED FOR <%s>, RC = <%d>"
NVD8347I	"%s: Exiting read notify response process"
NVD8348I	"LU6.2 Notify Manager <%s>, has successfully notified <%s> Client Automation Agent"
NVD8349I	"TCP Notify Manager <%s> started "
NVD8350E	"MESSAGE QUEUE WAS NOT INITIALIZED, UNABLE TO CONTINUE"
NVD8351I	"Signal received: <%d>. PID is <%d>"
NVD8352I	"Port for REXEC not defined, default <%i> will be used"
NVD8353E	"ALLOCATION ERROR <%s> FOR <%s>"
NVD8354E	"CHECK IP ADDRESS AND PORT NUMBER -- <%s>: FOR <%s>"
NVD8355E	"EDMWAKE is issued to power up the destination machine, will retry notify in <%i> seconds"
NVD8356E	"SEND ERROR <%s> FOR <%s>"
NVD8357E	"RECEIVE ERROR <%s> FOR <%s>"

Message Number	Log Message
NVD8358I	"Successfully notified <%s>"
NVD8359E	"NOTIFY AUTHORIZATION FAILED ON DESTINATION, CHECK USER ID <%s>"
NVD8360E	"ERROR OBTAINING IPADDR FOR WAKE_ON_LAN CALL -- <%s> FOR <%s>"
NVD8361I	"Issuing WAKE_ON_LAN call for IP@ = <%s>, MAC_addr = <%s> on Subnet = <%s>"
NVD8362I	"DialUp Notify Manager Name is <%s>"
NVD8363I	"Request: address=<%s:%s> socket=<%s> uid=<%s> password=<%s> \n cmd=<%s>"
NVD8365E	"SENDRAW ERROR - <%s>"
NVD8366E	"RECVRAW ERROR - <%s>"
NVD8367I	"Simulating send"
NVD8368I	"Streaming mode send"
NVD8369I	"Non-streaming mode send"
NVD8370E	"INVALID PORT <%s> SPECIFIED"
8400-8449	ZMETHOD
NVD8401E	"CANNOT RUN MORE THAN ONE METHOD AT A TIME"
NVD8402E	"CURRENTLY EXECUTING METHOD WITH PID<%08X>"
NVD8403E	"CURRENTLY EXECUTING METHOD WITH PID<%ld>"
NVD8404I	"Params: <%s>"
NVD8405I	"<%s> with parameters <%s> started <%s> at <%2.2d:%2.2d:%2.2d:%3.3d>"
NVD8406I	"Executing <%s> directly as a DLL"
NVD8407I	"Executing method <%s>"
NVD8408E	"METHOD <%s> PID=<%ld>, TERMINATED ABNORMALLY"
NVD8409E	"TIMEOUT EXPIRED WAITING FOR TEMINATION OF METHOD <%s> PID=<%ld>,"
NVD8410E	"METHOD <%s> , TIMED OUT PID=<%08X>"
NVD8411E	"METHOD <%s> , TIMED OUT PID=<%ld>"
NVD8412E	"METHOD <%s> , EXECUTION ERROR PID =<%08X>, rc = <%d>"

Message Number	Log Message
NVD8413E	"METHOD <%s> , EXECUTION ERROR PID =<%ld> , rc = <%d>"
NVD8414I	"%s completed at <%2.2d:%2.2d:%2.2d:%3.3d> , duration <%1.3f> seconds"
NVD8415I	"%s with Task ID <%08X> completed with RC=<%d> at <%2.2d:%2.2d:%2.2d:%3.3d> , duration <%1.3f> seconds"
NVD8416I	"%s with Task ID <%ld> completed with RC=<%d> at <%2.2d:%2.2d:%2.2d:%3.3d> , duration <%1.3f> seconds"
NVD8417I	"Method <%s> ended, exit status <%d>"
NVD8418E	"ERROR DISCONNECTING FROM NAMED PIPE"
NVD8419I	"<%s> USERID <%s> validated"
NVD8420I	"<%s> validating USERID <%s> "
NVD8421I	"Unable to find requested method <%s> "
NVD8422E	"ERROR ATTACHING METHOD <%s>"
NVD8423E	"ERROR LOADING METHOD DLL <%s>"
NVD8424I	"ZERRORM component - <%s>"
NVD8425I	"ZERRORM description - <%s>"
NVD8426I	"ZERRORM - <%s>"
NVD8427I	"Request: r_name=<%s> r_parms=<%s>"
NVD8428I	"Request: r_name=<%s> executed rc=<%i>"
NVD8429I	"Old method <%s> is requested, new method <%s> will be called."
8450-8499	ZRTRYMGR
NVD8450W	"Unable to read instance <%s.%s.%s.%s> , RC = <%d>"
NVD8451W	"Will retry reading instance later"
NVD8452E	"NOTIFY TYPE NOT SPECIFIED FOR <%s>.<%s>.<%s>.<%s> WILL BE DELETED"
NVD8453E	"RETRY ERROR FOR <%s>.<%s>.<%s>.<%s> REMOVED FROM RETRY QUEUE"
NVD8454E	"WRONG NOTIFY TYPE <%c> SPECIFIED FOR <%s>.<%s>.<%s>.<%s> WILL BE DELETED"
NVD8455W	"Instance will be ignored"
NVD8456I	"<%d> notification requests were re-scheduled"

Message Number	Log Message
NVD8457I	"No notification requests were re-scheduled"
9000-9100	Multicast
NVD9000I	"Multicasting File <%s> Size <%d> Priority <%d> Hits <%d>"
NVD9001I	"Multicasting Instance <%s> : <%ld> Files in <%ld> packets"
NVD9002E	"ERROR COLLECTING FILES TO BE MULTICASTED"
NVD9003E	"ERROR GETTING MULTICAST INFORMATION"
NVD9004E	"ERROR INITIALIZING MULTICAST SEGMENT"
NVD9005I	"End of multicast for instance <%s> rc <%d>"
NVD9006I	"Error multicasting files for instance <%s>"
NVD9007I	"Multicaster Task has started"
NVD9008I	"Initializing Multicast IPCs"
NVD9009I	"Fetching the Multicast info for the current session"
NVD9010I	"Starting collection process for Multicast Instance <%s>"
NVD9011I	"End of collection process for Multicast Instance <%s>"
NVD9012I	"Number of files to be multicasted <%d>"
NVD9013E	"UNABLE TO GET INFORMATION ABOUT LIST FILE <%s>"
NVD9014E	"MULTICAST LIST FILE <%s> IS EMPTY"
NVD9015E	"INVALID FILE COUNT <%d> EXPECTED <%d>"
NVD9016E	"ERROR LOADING MULTICAST PARAMETERS FOR INSTANCE <%s>"
NVD9017E	"ERROR INITIALIZING MULTICAST SESSION RC <%d> MMSG <%s>"
NVD9018E	"ERROR SENDING MULTICAST SESSION INFO RC <%d> MMSG <%s>"
NVD9019E	"ERROR OPENING MULTICAST LIST FILE <%s>"
NVD9020I	"Start Multicasting Instance <%s> Files <%d> Address <%s> Port <%d>"
NVD9021E	"ERROR SENDING FILE <%s> RC <%d> MMSG <%s>"
NVD9022I	"End Multicasting for instance <%s> rc <%d> Address <%s> Port <%d>"
NVD9023E	"INVALID ADDRESS FOR MULTICAST <%s>"
NVD9024E	"MISSING MULTICAST/BROADCAST ADDRESS IN INSTANCE <%s>"

Message Number	Log Message
NVD9025E	"ERROR CREATING MULTICAST SHARED MEMORY"
NVD9026E	"ERROR ATTACHING MULTICAST SHARED MEMORY"
NVD9027E	"ERROR ALLOCATING STORAGE FOR MULTICAST INFO"
NVD9028E	"ERROR CREATING MUTEX SEMAPHORE FOR MULTICASTING"
NVD9029E	"ERROR OPENING MULTICAST SEMAPHORE"
NVD9030E	"ERROR LOCKING MUTEX SEMAPHORE"
NVD9031E	"ERROR CREATING EVENT SEMAPHORE FOR MULTICASTING"
NVD9032W	"No more slots available for multicasting"
NVD9033E	"ERROR OPENING MUTEX SEMAPHORE"
NVD9034E	"ERROR OPENING EVENT SEMAPHORE"
NVD9035E	"ERROR LOCKING MUTEX SEMAPHORE FOR MULTICASTING"
NVD9036E	"ERROR UNLOCKING MUTEX SEMAPHORE FOR MULTICASTING"
NVD9037E	"ERROR SETTING EVENT SEMAPHORE FOR MULTICASTING"
NVD9038E	"ERROR RESETTING EVENT SEMAPHORE FOR MULTICASTING"
9101-9149	General
NVD9101I	"Converting ISO date to YY/MM/DD format for <%8.8s>.<%8.8s> <%.*s>"
NVD9102E	"ERROR CONVERTING ISO DATE TO YY/MM/DD FORMAT FOR <%8.8s>.<%8.8s> <%.*s>"
NVD9103I	"Converting Old Date to ISO for <%8.8s>.<%8.8s> <%.*s>"
NVD9104E	"ERROR CONVERTING OLD DATE TO ISO FOR <%8.8s>.<%8.8s> <%.*s>"
NVD9105I	"Translating template from network to host format"
NVD9106I	"Translating heap from network to host format"
NVD9107I	"Translating template from host to network format"
NVD9108I	"Translation of template from host to network format complete"
NVD9109I	"Translating heap from host to network format"
NVD9110I	"Translating heap from host to network format using template"
NVD9111I	"Translation of heap from host to network format complete"
NVD9112I	"Use ZEDMAMS to update the database"

Message Number	Log Message
NVD9113I	"=====
NVD9114I	"=====
NVD9115I	"=====
NVD9116E	*****
NVD9117E	"* %s "
NVD9118E	"ERROR ADDING UP %8.8s"
NVD9119E	"FLAG ERROR ADDING UP %8.8s"
NVD9120I	"Profile Section(%s)"
NVD9121I	"%-26.25s ==> %s"
NVD9122E	"* DATABASE ERROR *
NVD9123E	"INVALID TRACE VAR: <%s>"
NVD9124E	"INVALID OPTION FOR ADD"
NVD9125E	"WRONG SECTION NAME: <%s>"
NVD9126I	"Section Name <%s> was not found in PROFILE"
NVD9127I	"%-27s==> Parameter not found in section <%s> and was not set"
NVD9128I	"%-27s==> Parameter not found in section <%s>. Set to default <%s>"
NVD9129I	"Initially set parameter <%s> in section <%s> to <%s>"
NVD9130I	"%-27s==> Parameter not found in section <%s>. Set to default <%s>"
NVD9131I	"%-27s==> Parameter in section <%s> was set to <%s>"
NVD9132I	"Settings in read_TRACE: set ALL trace flags to YES in section <%s>"
NVD9133I	"%-27s==> Parameter not found in section <%s>. Set to default <%ld>"
NVD9134I	"%-27s==> Parameter not found in section <%s>. Set to default <%d>"
NVD9135I	"Settings in read_MSG_CONTROL: line #%i read: ==>%s<=="
NVD9136I	"Settings in read_MSG_CONTROL: set ALL trace flags to YES in section <%s>"
NVD9137I	"Settings in read_TRANSLATION: line #%i read: ==>%s<=="
NVD9138I	"Section (%s): Following default or overwritten values have been assigned"
NVD9139I	"%s *%.*s"

Message Number	Log Message
9150-9299	ZSYSINFO
NVD9150I	"Adjust on exit semaphore maximum value (SEMAEM) = <max_value>."
NVD9151I	"Maximum memory allowed in buffer cache (bufhvm) = <max_value>."
NVD9152I	"Maximum number of processes (v.v_proc) = <max_value>."
NVD9153I	"Maximum global priority in sys class (MACLSYSPRI) = <max_value>."
NVD9154I	"Maximum processes per user id (v.v_maxup) = <max_value>."
NVD9155I	"Auto update time limit in seconds (NAUTOUP) = <limits>."
NVD9156I	"Page stealing low water mark (GPGSLO) = <page_value>."
NVD9157I	"FSflush run rate (FSFLUSHR) = <flash_rate_value>."
NVD9158I	"Minimum resident memory for avoiding deadlock (MINARMEM) = <%ld>."
NVD9159I	"Minimum swappable memory for avoiding deadlock (MINASMEM) = <%ld>."
NVD9160I	"System<%s> Version %.2s Release %.2s.%.2s"
NVD9161I	"Number of current processes is <%lu>"
NVD9162I	"Total usable main memory size is <%llu>"
NVD9163I	"Available memory size is <%llu>"
NVD9164I	"Total swap space size is <%llu>"
NVD9165I	"Swap space still available is <%llu>"
NVD9200I	"Client Automation Configuration Server Information Section"
NVD9201I	"Client Automation Configuration Server is Version <%1d.%.2d.%.3d%s> Build <%d %s> Service Pack <%s>"
NVD9202I	"Client Automation Configuration Server is Portable"
NVD9203I	"Client Automation Configuration Server Build is CPU Specific"
NVD9204I	"Client Automation Configuration Server built on <%s>"
NVD9205I	"System Information Section"
NVD9206I	"Error Getting Windows Foreground Priority Boost Information [open]; rc=<%ld>"
NVD9207I	"Windows Foreground Priority Boost is <%ld>"
NVD9208W	"Warning, Windows Foreground Priority Boost should be set to 0"
NVD9209I	"Error Getting Windows Foreground Priority Boost Information [query];

Message Number	Log Message
	rc=<%ld>“
NVD9210I	“Error Getting Windows Foreground Priority Boost Information [close]; rc=<%ld>“
NVD9211I	“Error Getting OS/2 System Information; rc=<%ld>“
NVD9212I	“OS/2 Time Slice Minimum=<%ld> Maximum=<%ld> milliseconds”
NVD9213I	“OS/2 Maximum Wait is <%ld> seconds”
NVD9214I	“OS/2 Priority is <%s>“
NVD9215I	“Error Getting Version Information”
NVD9216I	“Windows %s %s Version <%ld.%ld>; Build Number <%d>; CSD <%s>“
NVD9219I	“Windows 3.1 Build Number <%d>; CSD <%s>“
NVD9220I	“System <%s> Version %02X Release %02X.%02X”
NVD9221I	“Environment <%s> Version %02X Release %02X.%02X”
NVD9222I	“OS/2 Version <%s>“
NVD9223I	“Error Getting Version Information; rc=<%d-%s>“
NVD9224I	“System <%s> Version <%s> Release <%s>“
NVD9225I	“Error Getting POSIX.1 Version Information; rc=<%d-%s>“
NVD9226I	“Error Getting POSIX.2 Version Information; rc=<%d-%s>“
NVD9227I	“Error Getting X/Open Version Information; rc=<%d-%s>“
NVD9228I	“Error Getting AES/OSF Version Information; rc=<%d-%s>“
NVD9229I	“Versions: POSIX.1=<%ld> POSIX.2=<%ld> X/Open=<%ld> OSF/AES=<%ld>“
NVD9230I	“There are <%d> Processors of type <%s>“
NVD9231I	“There is 1 Processor of type <%s>“
NVD9232I	“Error Getting Hardware Provider; rc=<%d-%s>“
NVD9233I	“Error Getting Hardware Serial Number; rc=<%d-%s>“
NVD9234I	“Error Getting Architecture Information; rc=<%d-%s>“
NVD9235I	“Error Getting Platform Information; rc=<%d-%s>“
NVD9236I	“Error Getting Number of Processors Configured; rc=<%d-%s>“

Message Number	Log Message
NVD9237I	"Error Getting Number of Processors Online; rc=<%d-%s>"
NVD9238I	"Hardware Manufacturer <%s>; Serial Number <%s>"
NVD9239I	"There are <%ld> Configured Processors of which <%ld> are Online"
NVD9240I	"Processor Architecture is <%s> and Platform is <%s>"
NVD9241I	"Error Getting sysinfo; rc=<%d-%s>"
NVD9242I	"Error Getting CPU Version; rc=<%d-%s>"
NVD9243I	"Processor Architecture is <%s>"
NVD9244I	"Physical Memory: Total=<%llu> Available=<%llu>"
NVD9245I	"System Virtual Memory: Total=<%llu> Available=<%llu> Page Size=<%llu>"
NVD9246I	"Process Virtual Memory: Total=<%llu> Available=<%llu>"
NVD9247W	"Warning, Windows Virtual Memory should minimally be 100MB"
NVD9248I	"Error Getting Memory Information"
NVD9249I	"Physical Memory: Total=<%llu> Resident=<%llu>"
NVD9250I	"Virtual Memory: Total=<%llu> Page Size=<%llu>"
NVD9251I	"Shared Memory: Total=<%llu>"
NVD9252I	"Process Memory: Total=<%llu>"
NVD9253I	"Error Getting System Memory Page Size; rc=<%d-%s>"
NVD9254I	"Error Getting Total Physical Pages; rc=<%d-%s>"
NVD9255I	"Error Getting Unused Physical Pages; rc=<%d-%s>"
NVD9256I	"Physical Memory: Total=<%llu> KB Available=<%llu> KB Page Size=<%llu> bytes"
NVD9257I	"Error Getting System Memory Page Size; rc=<%d-%s>"
NVD9258I	"Memory: Page Size=<%llu>"
NVD9259I	"Running on Domain <%s> Computer <%s> as User <%s>"
NVD9260I	"Error Getting Machine Information; rc=<%d-%s>"
NVD9261I	"Running on Machine <%s> Hostname <%s> as User <%s>"
NVD9262I	"Error <%ld> getting System Directory"
NVD9263I	"System Directory is <%s>"

Message Number	Log Message
NVD9264I	"Error <%ld> getting Windows Directory"
NVD9265I	"Windows Directory is <%s>"
NVD9266I	"Error <%d> getting volume information for drive <%s>"
NVD9267I	"Client Automation Configuration Server Database is on <%s> <%s> <%s> Drive <%c>"
NVD9268I	"Drive <%c> supports <%d> character file names"
NVD9269I	"Drive <%c> supports <%s>"
NVD9270I	"Disk Size Information for Database Directory <%s> not available"
NVD9271I	"Database resides in <%s>"
NVD9272I	"Database Disk Space Total=<%f> Available=<%f>"
NVD9273I	"Error getting Disk Size Information for Database Directory <%s>"
NVD9274I	"Disk Size Information for Log Directory <%s> not available"
NVD9275I	"Log will be created in <%s>"
NVD9276I	"Log Disk Space Total=<%f> Available=<%f>"
NVD9277I	"Error getting Disk Size Information for Log Directory <%s>"
NVD9278I	"Logging Client Automation Configuration Server module versions"
NVD9279I	"Logging Client Automation Configuration Server DLL versions"
NVD9280I	"Logging Method versions"
NVD9281I	"Logging Utility module versions"
NVD9282W	"Warning, The volume containing the Client Automation Configuration Server Database has only %.0lf free bytes"
NVD9283W	"Warning, The volume containing the Client Automation Configuration Server log has only %.0lf free bytes"
NVD9284I	"System swap file <%s>, total pages <%lu>, available <%lu>"
NVD9285I	"System swap file <%s>, total pages <%lu>, available <%lu>"
NVD9286W	"OS seems to be capable of supporting 3GB processing space, but is not configured to exploit additional address space."
NVD9287I	"Error Getting IPC Information: rc=<%d-%s>"
NVD9288I	"Maximum shared memory segment size (SHMMAX) = <%ld>."

Message Number	Log Message
NVD9289I	"Minimum shared memory segment size (SHMMIN) = <%ld>."
NVD9290I	"Total shared memory identifiers (SHMMNI) = <%ld>."
NVD9291I	"Maximum attached shared memory segments per process (SHMSEG) = <%ld>."
NVD9292I	"Entries in semaphore map (SEMMAP) = <%ld>."
NVD9293I	"Number of semaphore-identifiers in system (SEMMNI) = <%ld>."
NVD9294I	"Total number of semaphores in system (SEMMNS) = <%ld>."
NVD9295I	"Undo structures in system (SEMMNU) = <%ld>."
NVD9296I	"Maximum semaphores per ID (SEMMSL) = <%ld>."
NVD9297I	"Maximum # of operations per semop () call(SEMOPM) = <%ld>."
NVD9298I	"Maximum undo entries per process (SEMUME) = <%ld>."
NVD9299I	"Semaphore maximum value (SEMVMX) = <%ld>."
9300-9399	TASKLIM
NVD9300I	"Task context and intent limit statistics - %s"
NVD9301I	"%-20.20s %5.5s %5.5s %5.5s %5.5s %10.10s %10.10s"
NVD9302I	"%-20.20s %5.1d %5.1d %5.1d %5.1d %10.1lu %10.1lu"
NVD9304I	"
NVD9305I	"Task context and intent limit maximum for <%s> <%s> reset from <%0.1d> to <%0.1d>"
NVD9306I	"Task context and intent limit priority for <%s> <%s> reset from <%0.1d> to <%0.1d>"
NVD9307I	"
NVD9308I	"
NVD9309I	"
NVD9310I	"Task context and intent limits for all contexts: limit <%d>, current <%d>, peak <%d>, allowed <%lu>, rejected <%lu>"
NVD9311I	"Task context and intent limits for <%s> <%s>: limit <%d>, current <%d>, peak <%d>, allowed <%lu>, rejected <%lu>"
NVD9312I	"Task context and intent limits OK for context <%s>, intent <%s>"
NVD9313W	"Task context and intent limits exceeded for context <%s>, intent <%s>"

Message Number	Log Message
9400-9499	I18N / UTF-8
NVD9401I	"IDENTITY: Object contains CODEPAGE <%s>, LOCALE <%s>, CENCODE <%s>, SENCODE <%s>"
NVD9402I	"IDENTITY: Inbound and outbound Client Automation Agent objects are encoded in UTF8"
NVD9403I	"IDENTITY: Client Automation Configuration Server is running in legacy mode, UTF8 capable Client Automation Agent must exchange objects in legacy compatibility mode"
NVD9404I	"IDENTITY: Client Automation Agent is running in legacy mode"
NVD9405I	"IDENTITY: Inbound and outbound Client Automation Agent objects will be encoded for the locale <%s>"
NVD9406I	"IDENTITY: Inbound Client Automation Agent objects will be translated from locale <%s> into UTF8"
NVD9407I	"IDENTITY: Outbound Client Automation Agent objects will be translated from UTF8 into locale <%s>"
NVD9410W	"IDENTITY: Client Automation Agent LOCALE is missing or blank, expected Client Automation Agent locale <%s>"
NVD9411W	"IDENTITY: Client Automation Agent LOCALE <%s> does not match the Client Automation Configuration Server system locale <%s>"
NVD9412W	"IDENTITY: Client Automation Agent LOCALE <%s> does not match the expected Client Automation Agent locale <%s>"
NVD9413W	"IDENTITY: Client Automation Agent CODEPAGE is missing or blank, expected Client Automation Agent codepage <%lu>"
NVD9414W	"IDENTITY: Client Automation Agent CODEPAGE <%lu> does not match the Client Automation Configuration Server system codepage <%lu>"
NVD9415W	"IDENTITY: Client Automation Agent CODEPAGE <%lu> does not match the expected Client Automation Agent codepage <%lu>"
NVD9420I	"IDENTITY: Object contains IPADDR <%s>, HOSTNAME <%s>"
NVD9421I	"IDENTITY: Changing the task name from <%s> to match the actual Client Automation Agent IP address <%s>"
NVD9422I	"IDENTITY: Changing the remote IP name from <%s> to match the actual Client Automation Agent hostname <%s>"
NVD9423I	"IDENTITY: Object contains UUID <%s>"
NVD9424I	"IDENTITY: Object contains DOMNAME <%s>, ZUSERID <%s>"

Message Number	Log Message
9500-9599	I18N / UTF-8
NVD9500I	"The Client Automation Configuration Server Database is encoded in <%s>"
NVD9501I	"The locale LC_CTYPE is <%s>"
NVD9502E	"ERROR, THE CLIENT AUTOMATION CONFIGURATION SERVER REQUIRES A LOCALE THAT SUPPORTS UTF8"
NVD9503E	"ERROR, PLEASE CORRECT THE LOCALE SETTING AND RESTART THE CLIENT AUTOMATION CONFIGURATION SERVER"
NVD9504I	"Please review the 'locale' command and Client Automation Configuration Server documentation..."
9600-9699	Currently Unassigned
9700-9799	Currently Unassigned
9800-9899	Currently Unassigned
9900-9999	Currently Unassigned

Configuration Server Response Codes

Many Configuration Server operations (such as the REXX Methods, the Database Utilities, and the ZEDMAMS verbs) send a response code to the Configuration Server log that indicates the condition or outcome of the operation. Depending on the operation and its status, the response code will be either a return code or a reason code, as described in the sections that follow.

Return Codes

The following list of return codes are those that will be generated after a Configuration Server operation has completed. These codes are based on the outcome of the operation and are applicable to all Configuration Server processes.

Code	Type	Description
0	OK	The operation completed successfully and as expected.
4	Warning	The operation completed successfully, but a condition exists that might require the attention of an administrator.

Code	Type	Description
8	Error	The operation failed.
16	Severe Error	The operation failed and a condition exists that requires the attention of an administrator.

Reason Codes

Reason codes differ from return codes in that they are seen solely as the result of an abnormal condition during an internal operation on a Configuration Server. Therefore, they are associated with return codes 8 and 16.

The Configuration Server Reason Codes table lists and describes the reason codes that might be encountered in the Configuration Server log.

Note: In order to group similar messages in future, there are intentional gaps in the sequencing in the following table.

Configuration Server Reason Codes

Code	Description	Code	Description
100	Invalid arguments	101	No instances updated
102	No classes updated	103	No instances copied
104	No domains deleted	105	ALL instances not written
106	No instances written	107	STRING not found
108	KEYWORD not found	109	KVPARSE not found
110	No parent instance found	111	Attribute entry in error
200	Domain was not found	201	Class was not found
202	Instance was not found	203	No commands present in file
300	File name required	301	Domain name required
302	Class name required	303	Instance name required
304	File name invalid	305	Domain name invalid
306	Class name invalid	307	Instance name invalid
308	TODB value invalid	309	TOINST required
318	TOCLASS required	319	FROMDOMA required
320	FROMCLAS required	321	TODOMAIN required
322	FROMINST required	323	INSTANCE prefix required

Code	Description	Code	Description
324	Domain exists	325	Error deleting resource
326	FLDNAME required	327	LENGTH required
328	TYPE required	329	FIELD required
330	TODATA required	331	FROMDATA required
332	FROMNAME required	333	TONAME required
334	LISTALL required	335	FROMFLD required
336	TOFLD required	337	OLDSUFF required
338	NEWSUFF required	339	VALUE required
340	COUNT required	341	INFILE required
342	PREVIEW required	343	OLD_PREFIX required
344	NEW_PREFIX required	345	STRING required
346	LENGTH invalid	347	INDEX invalid
348	TYPE invalid	349	ATTRIBUTE NAME INVALID
350	ATTRIBUTE DATA INVALID	351	Instance name too long
352	NOT in DMA format	353	Invalid resource size
354	Resolution size difference	355	Same FROM and TO names
356	Same FROM and TO fields	357	Template structure error
500	MALLOC failed	501	Database path error
502	Database open error	504	Database close error
505	File not found	506	Domain not found
507	Class not found	508	Instance not found
509	Resource not found	510	Error creating class
511	Class lock error	512	Error reading class
513	Class exists already	514	Error writing class
515	Class unlock error	516	Error deleting resource
517	Instance lock error	518	Instance exists already
519	Error reading instance	520	Instance delete error
521	Error writing instance	522	Instance unlock error

Code	Description	Code	Description
523	Error creating domain	524	Error deleting domain
525	Error locking resource	526	Error reading resource
527	Error writing resource	528	Error deleting resource
529	ZEDMAMS.LOG open error	531	FILE open error
532	FILE read error	533	Database locked
534	Corrupt class	535	Corrupt instance
536	Corrupt resource	537	ZFILE open error
538	INPUT file error	539	OUTPUT file error

CSDB Domains and Classes

Introduction

The Configuration Server Database

The Configuration Server Database (CSDB) is an object-oriented database that stores all the information needed to manage assets on a device, including the software, patches, OS images and/or data that Sentient distributes, the policies that determine which users are entitled to which resources, and security and access rules for administrators.

The master CSDB on a Core server is the authoritative repository source for replicated CDSBs on Satellite servers. Satellite CDSBs are always replicated from an upstream Core or Satellite server.

Note: Do not make changes to the Sentient CSDB that is located on a Satellite server because it is a replication of its upstream server's (on either a Core or another Satellite) and any database changes at this level will be lost on the next synchronization with that upstream server.

The CSDB has a hierarchical structure containing four levels: files, domains, classes, and instances. Use the Sentient Enterprise Administrator Configuration Server Database Editor (CSDB Editor) to view and manipulate the CSDB components.

CSDB structure

Term	Description
File	Highest level in the hierarchy of the CSDB. Groups similar domains together. Example: The PRIMARY File is used to define and maintain the desired state. This is one of the pre-configured files distributed with Sentient.
Domain	Logically partitions a Sentient file. Groups similar classes together. Example: The POLICY Domain contains the classes needed to create users and

Term	Description
	groups.
Class	A category of the desired state. The class is a template for the attributes needed to create an instance of the class. Example: The USER Class of the POLICY Domain defines subscribers of Sentient-managed applications. It defines all of the attributes necessary to identify the agent computer to be managed by Sentient.
Class Instance or Instance	An object containing a specific occurrence of a class. This is analogous to a row in a relational data table, or a record in a traditional flat file. The attributes of a class instance object contain data describing one specific entity of that class. Example: A USER instance is an object created from the USER Class, containing the information needed to identify a subscriber's agent computer.
Attribute Attribute Value	An attribute is a data element of a class. The class contains the definition (e.g., the name, data type, description, and length) for each attribute comprising the class. Each class instance created from the class contains a value for each of the attributes defined in the class. Example: The NAME attribute of a USER Class contains the name of the subscriber, and the USERID attribute contains the User ID, as specified by the Sentient administrator.

How to Read the Class Reference Tables

The *Sentient Enterprise Console Online Help* describes the attributes of the classes in the default domains in the PRIMARY File. Also included is an explanation of the different components of the class reference tables.

Class Attributes

Within each of these domains, you will find classes that divide the files into functional categories. Class attributes can be one of four types as described in the following table:

Class Attribute Types and their Functions

Attribute Type Icon	Usage
Expression 	Contains a single line REXX command that is executed during resolution. In an attribute named ZSTOP, causes resolution of the current instance to terminate, if the expression evaluates to "true".
Variable 	A piece of named storage containing a variable value. The variable's value forms a part of the client's resolved desired state, and can influence the resolution process through messaging or symbolic substitution.
Connection (available) 	Class connections determine the path of resolution of an agent's desired state during the agent connection process. Available connections are attributes into which a connection to another class may be set. Set connections are attributes containing a specification for a connection to another class. A class connection is, in effect, a branch in the resolution

Connection (set)  Note: The letter in the icon denotes the connection type: A – Always C – Class I – Includes R – Requires	process. Refer to the <i>Sentient Enterprise Console Online Help</i> for more information. • An <i>Alwaysconnection</i> is always resolved, no matter what the name of its attribute. • A <i>Class connection</i> is resolved if its attribute's name is <code>_ALWAYS_</code>, or if the name of its attribute matches the current value of the system message. • An <i>Includesconnection</i> is always resolved, no matter what the name of its attribute. • A <i>Requiresconnection</i> blocks resolution of instances in the required instance and its children, whose instance names are identical to instance names in the requiring instance and its children. <i>Requires</i> and <i>Includes</i> connection types are useful, among other uses, for expressing inter-package relationships, connecting instances of the PACKAGE Class. When one package <i>Requires</i> another, components of the required package are superceded by the same component in the requiring package. When one package <i>Includes</i> another, both are deployed as a single entity.
Method 	Methods are programs executed as part of the resolution process. The method attribute identifies the program to be executed.

The first page of each class provides a brief overview. It identifies the file and domain that contain the class, the class name and icon, a description of the class including an explanation of what instances of the class are used for, and a list of permitted class connections for the class. The attributes in each class are listed alphabetically in table format.

Class attribute reference table columns

Column	Description
Attribute Name	Lists the class attributes. Attributes are not alphabetized; they appear in the order that you will find them in the CSDB Editor.
Explanation	Defines the attribute. Here you will find a more detailed description of the attribute. Where necessary, the column includes information on how and why this attribute is used.
Value of Base Instance	The default value of the base instance.
Length	The total number of characters allowed for the attribute.
Type	A code indicating the attribute type, as follows: C: Class connection M: Method V: Variable E: Expression

Column	Description
	I: Included
	R Required
	A Always

The ADMIN Domain

Use the ADMIN Domain to define administrative rights and rules for connecting classes. For more information, see the Persistent Systems support web site and *Sentient Enterprise Console Online Help*. To read the class reference tables, see the section "[How to Read the Class Reference Tables](#)" on page 861.

Administrator IDs (ADMINID) Class

File:	PRIMARY
Domain:	ADMIN
Class:	ADMINID
Icon:	
Description:	The ADMINID Class contains one instance for each authorized administrator. Each instance connects to one instance of the ZADMIN Class.
Can connect to:	ZADMIN

ADMINID Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ALWAYS	ZADMIN Connect to Used to connect the ADMINID instance to the ZADMIN instance. &ZADMFUNC must be passed as the message to the ZADMIN Class.		50 C
ADMDPT1 to ADMDPT5	Department Prefix Used to restrict access to policy-related class instances by the departments or workgroups assigned to an administrator. Used to control the DEPT, LAN, and PKGR types of administrators.	_NULL_	20 V
ADMNAME	Administrator Name		25 V
ADMPFX1to	Application Prefix	_NULL_	20 V

Attribute Name	Explanation	Value of Base Instance	Length Type
ADMPFX10	Used to restrict access to application-related class instances by the application prefixes assigned to an administrator. Used to control the DEPT, LAN, and PKGR types of administrators.		

Administrator Types (ZADMIN) Class

File:	PRIMARY
Domain:	ADMIN
Class:	ZADMIN
Icon:	
Description:	The ZADMIN Class describes the different administrator types. Instances of the ZADMIN Class define what files, domains, classes, and instances can be accessed and changed by each type of Sentient administrator. Also, class instance connection rules, as articulated by instances of the ZLIST Class, can be specified for each instance of the ZADMIN Class.
Can connect to:	ZCONNECT, ZLIST, ZLIST32, ZVERC, ZVERD, ZVERF, ZVERFDCI, ZVERI

ZADMIN Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
CONNECT	Class Connection Rules Connect this field to the appropriate ZLIST or ZCONNECT Class instance. It defines the class connection rules you want an Sentient administrator to have access to.		50 C
DELOBJ	Delete Object		50 C
GETINFO	Get Control Information for Classes		50 C
LISTCLAS	List Domain Classes Connect this field to the ZLIST Class instance. It defines the access an Sentient administrator will have to classes.		50 C
LISTDOMN	List File Domains		50 C

Attribute Name	Explanation	Value of Base Instance	Length Type
	Connect this field to the ZLIST Class instance. It defines the level of access you want an Sentient administrator to have.		
LISTFILE	List Files Connect this field to the ZLIST32 Class instance. It defines the access you want a Sentient administrator to have.		50 C
LISTINST	List Class Instances Connect this field to the appropriate ZLIST Class instance. It defines filters for Sentient files.		50 C
MASKCLAS	List Class Masks Connect this field to the ZLIST Class instance. It defines the access a Sentient administrator will have to classes.		50 C
MASKDOMN	List Domain Masks		50
MASKFILE	List File Masks Connect this field to the ZLIST32 Class instance. It defines filters for Sentient files		50 C
MASKINST	List Instance Masks		50 C
VEROBJ	List Discrete Instances		50 C

Attribute Verify (ZVERA) Class

File:	PRIMARY
Domain:	ADMIN
Class:	ZVERA
Icon:	
Description:	Instances of the ZVERA Class can be used to group instances of the ZLIST Class that define attribute access in your administrator schema.

ZVERA Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ADD	Add Filter		50 C

Attribute Name	Explanation	Value of Base Instance	Length Type
DELETE	Delete Filter		50 C
DRAGDROP	Drag Drop Filter		50 C
EXECUTE	Execute Filter		50 C
LIST	List Filter		50 C
MASK	Mask Filter		50 C
UPDATE	Update Filter		50 C

Class Verify (ZVERC) Class

File:	PRIMARY
Domain:	ADMIN
Class:	ZVERC
Icon:	
Description:	Instances of the ZVERC Class can be used to group instances of the ZLIST Class that define class access in your administrator schema.
Can connect to:	ZLIST

ZVERC Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ADD	Add Filter		50 C
DELETE	Delete Filter		50 C
DRAGDROP	Drag Drop Filter		50 C
EXECUTE	Execute Filter		50 C
LIST	List Filter		50 C
MASK	Mask Filter		50 C
QUERY	Query Filter		50 C
UPDATE	Update Filter		50 C

Connection Rules (ZCONNECT) Class

File	PRIMARY
Domain:	ADMIN
Class:	ZCONNECT
Icon:	
Description:	The ZCONNECT Class lists the rules for what can be connected to other classes.
Can connect to:	ZLIST

ZCONNECT Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ADMINID	ADMINID Connect List This connection field allows you to connect a ADMINID Class instance to instances of the ZLIST Class.		50 C
BEHAVIOR	BEHAVIOR Connect List This connection field allows you to connect a BEHAVIOR Class instance to instances of the ZLIST Class.		50 C
CONUSEID	CONUSEID Connect List This connection field allows you to connect a CONUSEID Class instance to instances of the ZLIST Class.		50 C
DEPT	DEPT Connect List An alphanumeric name that connects the DEPT defined in this class. This connection field allows you to connect a DEPT Class instance to instances of the ZLIST Class.		50 C
DESKTOP	DESKTOP Connect List This connection field allows you to connect a DESKTOP Class instance to instances of the ZLIST Class.		50 C
DIALOG	DIALOG Connect List This connection field allows you to connect a DIALOG Class instance to instances of the ZLIST		50 C

Attribute Name	Explanation	Value of Base Instance	Length Type
	Class		
DRIVEMAP	DRIVEMAP Connect List		50 C
FILE	FILE Connect List This connection field allows you to connect a FILE Class instance to instances of the ZLIST Class.		50 C
LDS	LDS Connect List		50 C
LME	LME Connect List		50 C
LOCATION	Location Connect List This connection field allows you to connect a LOCATION Class instance to instances of the ZLIST Class.		50 C
MANUFACT	MANUFACT Connect List		50 C
METACLAS	METACLAS Connect List This connection field allows you to connect a METACLAS Class instance to instances of the ZLIST Class.		50 C
MODEL	Model Connect List		50 C
PACKAGE	PACKAGE Connect List This connection field allows you to connect a PACKAGE Class instance to instances of the ZLIST Class.		50 C
PANEL	PANEL Connect List This connection field allows you to connect a PANEL Class instance to instances of the ZLIST Class.		50 C
PRDMAINT	PRDMAINT Connect List This connection field allows you to connect a PRDMAINT Class instance to instances of the ZLIST Class.		
REGISTRY	REGISTRY Connect List This connection field allows you to connect a REGISTRY Class instance to instances of the ZLIST Class.		50 C
ROLE	ROLE Connect List		50 C
SCANNER	SCANNER Connect List This connection field allows you to connect a SCANNER Class instance to instances of the ZLIST Class.		50 C

Attribute Name	Explanation	Value of Base Instance	Length Type
SUBNET	SUBNET Connect List		50 C
UNIXFILE	UNIX File Connect List This connection field allows you to connect a UNIX FILE Class instance to instances of the ZLIST Class.		50 C
USER	USER Connect List This connection field allows you to connect a USER Class instance to instances of the ZLIST Class. An alphanumeric name defining a new USER instance.		50 C
WMI	WMI Connect List This connection field allows you to connect a WMI Class instance to instances of the ZLIST Class		50 C
WORKGRP	WORKGRP Connect List This connection field allows you to connect a WORKGRP Class instance to instances of the ZLIST Class. An alphanumeric name defining a new WORKGRP instance.		50 C
ZADMIN	ZADMIN Connect List This connection field allows you to connect an Administrator Class instance to instances of the ZLIST Class. An alphanumeric name defining a new ZADMIN instance.		50 C
ZCONNECT	ZCONNECT Connect List This connection field allows you to connect a ZCONNECT Class instance to instances of the ZLIST Class. An alphanumeric name defining a new ZCONNECT instance.		50 C
ZSERVICE	ZSERVICE Connect List This connection field allows you to connect a ZSERVICE Class instance to instances of the ZLIST Class. An alphanumeric name defining a new ZSERVICE instance.		50 C
ZVERC	ZVERC Connect List This connection field allows you to connect a ZVERC Class instance to instances of the ZLIST Class. An alphanumeric name defining a new ZVERC instance.		50 C
ZVERD	ZVERD Connect List This connection field allows you to connect a ZVERD Class instance to instances of the ZLIST		50 C

Attribute Name	Explanation	Value of Base Instance	Length Type
	Class. An alphanumeric name defining a new ZVERD instance.		
ZVERF	ZVERF Connect List This connection field allows you to connect a ZVERF Class instance to instances of the ZLIST Class. An alphanumeric name defining a new ZVERF instance.		50 C
ZVERFDCI	ZVERFDCI Connect List This connection field allows you to connect a ZVERFDCI Class instance to instances of the ZLIST Class. An alphanumeric name defining a new ZVERFDCI instance.		50 C
ZVERI	ZVERI Connect List This connection field allows you to connect a ZVERFDCI Class instance to instances of the ZLIST Class. An alphanumeric name defining a new ZVERI.		50 C

Console Privileges (ZCONUSER) Class

File:	PRIMARY
Domain:	ADMIN
Class:	ZCONUSER
Icon:	
Description:	Reserved for future use.

ZCONUSER Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ALWAYS	Connect to Class connections determine the path of resolution of an agent's desired state. An _ALWAYS_ class connection is always resolved.		50 C

Console User IDs (CONUSEID) Class

File:	PRIMARY
Domain:	ADMIN
Class:	CONUSEID
Icon:	
Description:	Reserved for future use.
Can connect to:	ZCONUSER

CONUSEID Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
<u>_ALWAYS_</u>	ZCONUSER Connect to Class connections determine the path of resolution of an agent's desired state. An <u>_ALWAYS_</u> class connection is always resolved.		50 C
CONNAME	Console User Name		25 V

Domain Verify (ZVERD) Class

File:	PRIMARY
Domain:	ADMIN
Class:	ZVERD
Icon:	
Description:	Instances of the ZVERD Class can be used to group instances of the ZLIST Class that define domain access in your administrator schema.
Can connect to:	ZLIST

ZVERD Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ADD	Add Filter		50 C

Attribute Name	Explanation	Value of Base Instance	Length Type
DELETE	Delete Filter		50 C
DRAGDROP	Drag Drop Filter		50 C
EXECUTE	Execute Filter		50 C
LIST	List Filter		50 C
LIST	Filtering Method		50 M
MASK	Mask Filter		50 C
UPDATE	Update Filter		50 C

File Verify (ZVERF) Class

File:	PRIMARY
Domain:	ADMIN
Class:	ZVERF
Icon:	
Description:	Instances of the ZVERF Class can be used to group instances of the ZLIST Class that define file access in your administrator schema.
Can connect to:	ZLIST32

ZVERF Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ADD	Add Filter		50 C
DELETE	Delete Filter		50 C
DRAGDROP	Drag Drop Filter		50 C
EXECUTE	Execute Filter		50 C
LIST	List Filter		50 C
MASK	Mask Filter		50 C
UPDATE	Update Filter		50 C

Function Verify (ZVERFDCI) Class

File:	PRIMARY
Domain:	ADMIN
Class:	ZVERFDCI
Icon:	
Description:	Instances of the ZVERFDCI Class can be used to group instances of the ZLIST Class that define function access in your administrator schema.
Can connect to:	ZLIST, ZLIST32, ZVERC, ZVERD, ZVERF, ZVERI

ZVERFDCI Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
A	<A>		50 C
A	<A>		50 M
C	Class Verify		50 C
C	Class Ver Method		50 M
D	Domain Verify		50 M
D	Domain Ver Verify		50 M
F	File Verify	ZVERFDCI.LIMITED	50 C
F	File Ver Method		50 M
I	Instance Verify		50 C
I	Instance Ver Method		50 M

Instance Verify (ZVERI) Class

File:	PRIMARY
Domain:	ADMIN
Class:	ZVERI
Icon:	
Description:	Instances of the ZVERI Class can be used to group instances of the ZLIST Class that define instance access in your administrator schema.

File:	PRIMARY
Can connect to:	ZLIST32

ZVERI Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ADD	Add Filter		50 C
DELETE	Delete Filter		50 C
DRAGDROP	Drag Drop Filter		50 C
EXECUTE	Execute Filter		50 C
LIST	List Filter		50 C
LIST	Filtering Method		50 M
MASK	Mask Filter		50 C
QUERY	Query Filter		50 C
UPDATE	Update Filter		50 C

Name Lists (8) (ZLIST) Class

File:	PRIMARY
Domain:	ADMIN
Class:	ZLIST
Icon:	
Description:	The ZLIST Class allows you to specify access to eight-character files and class instances. The instances of the ZLIST Class specify access to VM domains and classes, and articulate class connection rules. An instance of this class relates the files, domains, classes, and instances to the rules that govern their manipulation by an Sentient administrator.

ZLIST Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ZNAME01 to ZNAME20	Name Allows you to enter access controls for Sentient domains and classes. You can use a wildcard character to specify access to all domains, or limit		8 V

Attribute Name	Explanation	Value of Base Instance	Length Type
	access by entering one or more valid eight-character domain names.		

Name Lists (32) (ZLIST32) Class

File:	PRIMARY		
Domain:	ADMIN		
Class:	ZLIST32		
Icon:			
Description:	The ZLIST32 Class allows you to specify access to 32-character files and class instances. You can also create instances of the ZLIST32 Class to predefine filters for class instances.		

ZLIST32 Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ZNAME001 to ZNAME020	Instance Name Allows you to enter access controls for Sentient files and class instances. You can use a wildcard character or limit access by entering one or more valid 32-character file or class instance names.		32 V

The AUDIT Domain

Use the AUDIT Domain to configure tasks that will inventory or audit agent computers' assets. To read the class reference tables, see the section "[How to Read the Class Reference Tables](#)" on [page 861](#)

Application (ZSERVICE) Class

File:	PRIMARY		
Domain:	AUDIT		
Class:	ZSERVICE		

File:	PRIMARY
Icon:	
Description:	These are sample services distributed with the Sentient Enterprise Inventory Manager (Inventory Manager). The AUDIT.ZSERVICE instance is connected to a policy instance. A policy instance can be an instance of the Users, Departments, or Workgroups class. It can also be a customer-defined class within the POLICY Domain. Each of the sample ZSERVICE Classes is connected to the PACKAGE instances.
Can connect to:	PACKAGE, FILE, BEHAVIOR, DIALOG, DEPT, WORKGRP, TIMER, VGROUP, STAGER, HTTP, EXECUTE, ZSERVICE, PDACONFIG

ZSERVICE Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ALWAYS	Contains Class connections determine the path of resolution of an agent's desired state. An _ALWAYS_ class connection is always resolved.		50 C
ALWAYS	Contains An _ALWAYS_ class connection is always resolved. An Includes connection is used to express inter-package relationships.	AUDIT.RIMOPTS . DIFF_INSTALL_UPDATE	50 I
ALWAYS	Contains An _ALWAYS_ class connection is always resolved. An Includes connection is used to express inter-package relationships.	AUDIT.PACKAGE .AUDIT _EXECUTE _BEHAVIOR	50 I
ALWAYS	Utility Resolution Method Any method you specify for this attribute is unconditionally executed by Sentient when this instance is resolved. A valid method name, for example, ZSYSTEM.ZMETHOD. PUT_ZERROR_HIST.		51 M
ADAPTIVE	Auto Adaptability [Y/N] Indicates whether the installed package is dependent on client settings that must be monitored periodically. If the settings change, the client		1 V

Attribute Name	Explanation	Value of Base Instance	Length Type
	must reconnect to the Configuration Server to get new or different components. Useful for "plug and play" services.		
AUTHOR	Author Name Name of the author of the service that appears in the properties for the service in the Service List.		50 V
BDELETE	Service Pre-Delete Method Method to run before deleting the service.		100 M
BUPDATE	Service Pre-Update Method Method to run before updating the service.		100 M
BREPAIR	Service Pre-Repair Method Method to run before repairing the service.		100 M
CACHE	App Element Caching [Y/N]		1 V
CACHELIM	Percent Disk Limit for Cache Cache limit, which is defined as the percentage of used drive space. Type a number between 000 and 100. Note: Used for Windows Installer applications only. If the percentage of used space is greater than the cache limit, then all of the cached files for the product are removed and the cache folder is deleted. This is checked after every file is cached on the disk.	000	3V
CACHELOC	CACHE Location on Client For Windows Installer applications only. Location of the folder on the agent computer that is used to cache the compressed application files needed for the product.		254 V

Attribute Name	Explanation	Value of Base Instance	Length Type
	CM support for Windows Installer tags the PRODGUID value to this value to create the folder. For example, if CACHELOC=C:\progra~1\PSL and PRODGUID = 12345_XXXX the cache folder would be: c:\progra~1\PSL\12345_XXXX\cache Note: The folder name \cache is automatically appended to PRODGUID. If you are not deploying a Windows Installer-enabled application, the files will be cached in IDMDATA. _UNDEF_		
CATGROUP	Catalog Group Name Type a name for the catalog that will be displayed to subscribers when they click Properties in the Application Self-service Manager user interface.		30 V
COMPSIZE	Application Size – Compressed The size of the compressed application displayed to the subscribers when they click Properties in the Application Self-Service Manager user interface. Cumulative value of the COMPSIZE defined in the PACKAGE Class.		12 V
DELDATE	Delete Date Indicates when the application was removed (in local time) from the agent computer. The Agent manages and maintains this attribute.		32 V

Attribute Name	Explanation	Value of Base Instance	Length Type
DESCRIPT	Application Description Description of the service that appears in the properties for the service in the Service List. Value is set initially in the Long Description field in the New Application Wizard (used to create a service).		255 V
ERTYPE	Event Reporting Method [O/E/X] Set on the Configuration Server, this sends an APPEVENT object to the Configuration Server. Currently supports object format only	o	1 V
EVENTS	Events to Report Indicates which events to report on.	AI=B AD=B AU=B AR=B AV=F VA=B VD=B	50 V
INSTDATE	Installed Date Indicates when the application was installed (in local time) on the agent computer. The Agent manages and maintains this attribute. The date is in text form, as in: Jan 28, 1998 16:10:00.		32 V
ITEMS	Number of Component Instances		7 V
LREPAIR	Local Repair [Y/N] Enables local repair of broken applications. If an application is broken because of missing files, the files (stored locally) can be used to repair the application.		1 V

Attribute Name	Explanation	Value of Base Instance	Length Type
MCELIBL	Service Multicast Eligible [Y/N] Indicates if the application is eligible for multi-casting.	Y	1 V
NAME	Friendly Name Friendly name for the service. This appears in the properties for the service in the Service List.		80 V
OWNER	Application Contact Package owner, as documented by the Admin Publisher.		80 V
PRICE	Price Type the price of an application that will be displayed to subscribers when they click Properties in the Application Self-service Manager user interface.		10 V
PUBDATE	Published Date of Service Date service was published.		32 V
REBOOT	Install/Update/Delete/Version Change Used to restart the agent computer after the service is installed. Blank = Restart only if a file being deployed is already in use by an application on the agent computer. INSTALL,DELETE = Restart if an application is installed or deleted, or if a file being deployed is already in use. VERSION CHANGE = Restart if an application is changing versions, or if a file being deployed is already in use. UPDATE = Restart if an application is updated or repaired during the agent connect process, or if a file being deployed is already in use.		64 V

Attribute Name	Explanation	Value of Base Instance	Length Type
RECONFIG	Reconfiguration Enabled [Y/N] Indicates whether an application can be relocated after it has been installed.		1 V
REMOVAL	Un-Managed Behavior [A/D/U] Controls removal of the service. If ZVCMO is set to M , set REMOVAL to: A = Abandon (deletes the objects, but not the components) D = Delete (deletes the objects and components) If ZVCMO is set to O , set REMOVAL to: A = Abandon (deletes the objects, but not the components) D = Delete (deletes the objects and components) U = Unmanage (does not delete the objects or components)	D	1 V
RSTRSIZE	Download Restart Threshold (in bytes)	0	11 V
RUNDLG	Dialog Processing [Y/N] Specifies whether to enable processing of dialog boxes during the installation of the service.	N	1 V
SCHEDOK	Update Schedule Locally [Y/N] Specifies whether the subscribers are allowed to change the update schedule for the service locally. Y = Subscriber is allowed to change the schedule. N = Configuration Server controls the update schedule.		1 V
SIZE	Application Size –		12

Attribute Name	Explanation	Value of Base Instance	Length Type
	Uncompressed The size of the uncompressed application displayed to the subscribers when they click Properties in the Application Self-service Manager user interface.		V
UIOPTION	Progress Indicator [NONE/FULL/INFO] Controls whether the status window appears. NONE = No interface appears. FULL = Interface appears and Cancel button is available. INFO = Interface appears with no option to cancel.		4 V
UIOPTMSI	MSI Progress Indicator [NONE/FULL/INFO] Controls whether the MSI status window appears. NONE = No interface appears. FULL = Displays what is happening on the agent computer and provides the subscriber with controls to make changes. INFO = Displays what is happening on the agent computer, but disables all the controls so that the subscriber cannot make any changes. See the Setup log files and Event Viewer for warnings, errors, successes, failures, and other details.		
UPDDATE	Upgrade Date (Programmatic) Stores the date when the file was updated on the Configuration Server.		12 V
UPGDATE	When Application was Upgraded		32 V

Attribute Name	Explanation	Value of Base Instance	Length Type
	Stores the date when the application was last upgraded (in local time) on the agent computer. This value is set by the Sentient user after the dialogs have been installed. This date can be used for reporting purposes. The date is in text form, as in: Jan 28, 1998 16:10:00.		
URL	WEB URL Name Address of a Web page where the subscriber can find additional information about the service.		255 V
VENDOR	Vendor Name Name of the vendor of the service.		24 V
VERDATE	Verified Date of Service Indicates when the application was last verified (in local time) on the agent computer. The Agent manages and maintains this attribute.		12 V
VERSION	Version Description		255 V
ZAVIS	Available, Verified, Installed, Sync F The Agent manages and maintains this attribute to show the different states of the application in the Service List. Y = Yes N = No X = Unknown The four states are: Available indicates whether a service is available from the Configuration Server. Verified indicates whether a	YXNX	4 V

Attribute Name	Explanation	Value of Base Instance	Length Type
	service has been verified. Installed indicates whether the service has been installed. Synchronized indicates whether the installed service has all of the latest changes from the Configuration Server.		
ZBITARCH	Target Bit Architecture [/32/64] Set to 32 to deploy to 32-bit architectures. Set to 64 to deploy to 64-bit architectures. Set to blank to deploy to all architectures.		8 V
ZCONTEXT	Component Context [M/U]		1 V
ZCREATE	Service Installation Method Sentient-supplied method or fully qualified, user-defined method.		100 M
ZDELETE	Service Delete Method Sentient-supplied method or fully qualified, user-defined method.		100 M
ZDISCONN	Disconnect on Install [Y/N] Allows the client to disconnect from the Configuration Server if there is an open session with it. Y = Disconnects the client from the Configuration Server. N = Does not disconnect from the client from the Configuration Server.	Y	1 V
ZINIT	Service Initialization Method A fully qualified, user-defined method.		100 M
ZREPAIR	Service Repair Method		100

Attribute Name	Explanation	Value of Base Instance	Length Type
			M
ZSTOP000 to ZSTOP002	Stop Resolution Expression Expressions evaluating to "true" in ZSTOP attributes cause resolution of the instance to be skipped. If left blank, the instance is not skipped, and resolution continues. This is useful for assigning a version to a specific set of users. Use the CSDB Editor to set this attribute.		255 E
ZSTOP999	Stop Unless RCA Connect	"\UPPER(EDMGETV (ZMASTER, ZDOMNAME))='AUDIT')"	50 E
ZSVCCAT	Service Visible in Catalog? [Y/N]		1 V
ZSVCCSTA	Service Status on Client (999) Service status code. 000 = All actions for service completed. 008 = At least one resource had a writer error. 014 = No disk space for at least one resource. 015 = ZDLIMIT exceeded for at least one resource. 024 = A method timed out. 036 = A create method set ZBRC>7. 900 = Not all files were verified. No work done for service.	999	3 V
ZSVCMO	Mandatory/Optional Service [M/O/MO/OM] Indicates whether the service is mandatory or optional on the Agent. If you are using Application Manager and Application Self-service Manager, you could also	M	1 V

Attribute Name	Explanation	Value of Base Instance	Length Type
	specify mandatory and then optional, MO , or optional then mandatory, OM . The first character indicates how the application should be handled before installation. The second character indicates how the application should be handled after installation.		
ZSVCMODE	Application Context [M/U/MU/EMU] Specifies whether the service is to be installed in the Machine or User context. M = Machine U = User MU = Both machine and user components. Verify that the machine components have been installed before installing the user components. EMU = Enhanced machine and user components.		2 V
ZSVCNAME	Service Name/Description For example, VB Calculator.	Unknown Service	24 V
ZSVCPRI	Service Create Ordering [01-99] Enables you to specify a priority level for this service. If disk space on the client desktop is not available, services will be downloaded based on the priority level set here. 01 = Highest priority 10 = Default value 99 = Lowest priority		2 V
ZSVCCRASH	Verify Attributes [Y/F/D/N] The Agent will check permissions on files and/or directories when ZSVCCRASH is set. Valid values		1 V

Attribute Name	Explanation	Value of Base Instance	Length Type
	are: Y – verify files and directories F – verify files only D – verify directories only N – do not verify permissions		
ZSVCTTYP	Application Target Type [A/S] Indicates which Client Automation agent will receive this application. A = Application Manager S = Application Self-Service Manager	Y	2 V
ZSYSACCT	Install Under System Account [Y/N/M] Specifies whether to install the service under the system account or the user's account. Y = The application is installed using the system rights. N = The application is installed using the rights of the user who is logged on. M = The application is installed using the system rights if the current connect is in the machine context (context =m).	Y	1 V
ZUPDATE	Service Update Method Sentient-supplied method or fully qualified, user-defined method.		100 M
ZVERIFY	Service Verify Method		100 M

Audit Packages (PACKAGE) Class

File:	PRIMARY
Domain:	AUDIT
Class:	PACKAGE
Icon:	
Description:	Defines what information to collect, and then what actions to take. These packages would contain various audit components. A good example is an audit of running services on a desktop. The AUDIT.ZSERVICE instance must contain a connection to an AUDIT.PACKAGE instance.
Can connect to:	DESKTOP, FILE, PATH, MSIFEATS, MSI, PACKAGE, PDACONFIG, REGISTRY

PACKAGE Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ALWAYS	Connect to Class connections determine the path of resolution of an agent's desired state. An _ALWAYS_ class connection is always resolved.		50 C
ALWAYS	Package Includes Class connections determine the path of resolution of an agent's desired state. An _ALWAYS_ class connection is always resolved.		50 C
BEHAVIOR	Behavior Includes connection to BEHAVIOR Class instances.		255 I
COMPSize	Package Compressed Resource Size Compressed size of package (in bytes).		12 V
DESCRIPT	Application Description Description of the service that appears in the properties for the service in the Service List. Value is set initially in the Long Description field in the New Application Wizard (used to create a service).		255 V
DESKTOP	Desktop Includes connection to DESKTOP Class instances.		255 I

Attribute Name	Explanation	Value of Base Instance	Length Type
FILE	File Includes connection to FILE Class instances.		255 I
INCLUDES	Package Includes Generic Includes connection.		50 I
ITEMS	Number of Component Instances Number of component instances in a package at time of publishing.		7 V
NAME	Friendly Name		80 V
OWNER	Application Contact Package owner, as documented by the Admin Publisher.		80 V
PATH	Path Includes connection to PATH Class instances.		255 I
PKGNAME	Package Name		80 V
REGISTRY	Registry Includes connection to REGISTRY Class instances.		255 I
RELEASE	Package Release Release level of the package, as documented by the Admin Publisher.		80 V
REQUIRES	Required Package Level Generic Requires connection.		80 C
SIZE	Package Resource Size Size of package (in bytes).		12 V
UNIXFILE	UNIX File Includes connection to UNIXFILE Class instances.		255 I
ZBITARCH	Target Bit Architecture [/32/64] - Set to 32 to deploy to 32-bit architectures. - Set to 64 to deploy to 64-bit architectures. - Set to blank to deploy to all architecture.		8 V
ZSTOP000	Stop Resolution Expression		255

Attribute Name	Explanation	Value of Base Instance	Length Type
	Expressions evaluating to "true" in ZSTOP attributes cause resolution of the instance to be skipped. If left blank, the instance is not skipped, and resolution continues. This is useful for assigning a version to a specific set of users. Use the CSDB Editor to set this attribute.		V
ZSTOP001to ZSTOP003	Stop Resolution Expression Expressions evaluating to "true" in ZSTOP attributes cause resolution of the instance to be skipped. If left blank, the instance is not skipped, and resolution continues. This is useful for assigning a version to a specific set of users. Use the CSDB Editor to set this attribute.		255 E

Behavior Services (BEHAVIOR) Class

File:	PRIMARY
Domain:	AUDIT
Class:	BEHAVIOR
Icon:	
Description:	Defines instances that enable the execution of auditing on the client. Normally, there is no need to add or modify instances in this class.
Can connect to:	PACKAGE

BEHAVIOR Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ALWAYS	Connect to Class connections determine the path of resolution of an agent's desired state. An _ALWAYS_ class connection is always resolved.		50 C
CMPMD5	Compressed File MD5 Signature	_UNDEF_	32 V
FILEMD5	File MD5 Signature	_UNDEF_	32 V

Attribute Name	Explanation	Value of Base Instance	Length Type
LOCATION	Location		255 V
NAME	Friendly Name Name that appears in the CSDB Editor for this class.		80 V
OBJGUID	Instance GUID Object Global Unique Identification – for future use.	_UNDEF_	32 V
ZCMPKEY	Compression Key		80 V
ZCMPSIZE	Compressed File Size From promotion process if the compressed file option was selected. The size of the compressed file.		11 V
ZCREATE	Method to Install Behavior Sentient-supplied method or fully qualified, user-defined method.		50 M
ZDELETE	Method to De-install Behavior Sentient-supplied method or fully qualified, user-defined method.	EDMREMF	50 M
ZFILEUPD	Client File Update/Add Method If a file has been updated by an administrator, this method will be executed. Sentient-supplied method or fully qualified, user-defined method.		50 M
ZINIT	Resource Initialization Method A fully qualified, user-defined method.		50 M
ZOPENERR	Client Method on File Open Error Sentient-supplied method or a fully qualified, user-defined method.	EDMLKM	50 M
ZRSCCFIL	Behavior File Name Specific file to be distributed. A generic pattern (a wildcard character, such as *) or an explicit filename.		255 V
ZRSCCKPT	Resource Checkpoint Not used. Replaced by ZRSCRSTR.		1 V
ZRSCCONF	Confirm File Download [Y/N]	N	1

Attribute Name	Explanation	Value of Base Instance	Length Type
	Y enables you to confirm that the resource was downloaded.		V
ZRSCCRC	Resource CRC From the promotion process if the CRC option was selected. Eight-character calculated hexadecimal value.		8 V
ZRSCCSTA	Client File Status Resource status code. 000 = Successfully downloaded 001 = Data can be found in EDMDATA directory 004 = Resource with no data 008 = Could not create the file 009 = No disk space for file (ZRSCMO=O) 010 = ZDLIMIT was exceeded (ZRSCMO=O) 011 = No data in RCA for resource 014 = No disk space (ZRSCMO=M) 015 = ZDLIMIT was exceeded (ZRSCMO=M) 024 = Time-out waiting for method. 036 = A method was set ZBRC>7 999 = No work done for resource	999	3 V
ZRSCDATE	Behavior Date Stamp Date stamp on resource file from promote process.		8 V
ZRSCMDAT	Mgr. Stores Data [Y/N]	Y	1 V
ZRSCMMEM	PDS Member Name PDS where resource actually resides. PDS name where resource resides in Configuration Server (MVS only).		8 V
ZRSCMO	Mandatory/Optional [M/O]	M	1 V
ZRSCNAME	Behavior Name	&ZRSCCFIL	30

Attribute Name	Explanation	Value of Base Instance	Length Type
	Name or description of resource. Valid name of resource to distribute to the Agent.		V
ZRSCOBJN	Persistent Systems Object Name The name of the in-storage object. Valid only when the ZLOCMGR location is STORAGE.		8 V
ZRSCPRI	Create Sequence Priority Enables you to specify a priority level for this resource. If disk space on the client desktop is not available, resources will be downloaded based on the priority set here. 01 = Highest priority 10 = Default value 99 = Lowest priority	10	2 V
ZRSCRASH	DOS File Attribute [R/A/S/H] Enables you to specify file attributes for a resource. Read-only Archive System Hidden.		4 V
ZRSCRSTR	Restart [Y/N] You can either restart a download process at the point it was interrupted (Yes), or resend the entire object (No).	N	3 V
ZRSCSIZE	Behavior Size Byte count of resource file from promote process.		11 V
ZRSCSRC	Resource Source that is Publisher		20 V
ZRSCSRC	Resource Source		1 V
ZRSCSTYP	Server File Type [BINARY/TEXT] File format for file download: Binary, Text, or Object.	BINARY	8 V
ZRSCSVRB	Version Skip Rebuild Flag [Y/N] Enables you to skip the reactivation of version resources if the version is already active. Y = Skip the rebuilding of version resources N = Rebuild version resources	Y	1 V
ZRSCTIME	Behavior Time Stamp Time stamp on		8

Attribute Name	Explanation	Value of Base Instance	Length Type
	resource file from promote process.		V
ZRSCVERS	Resource Version - From Promote Version number of resource from promotion process.		12 V
ZRSCVRFY	Verify Resource File on Connect Verifies the file statistics of a resource file on the desktop during each agent connect process. If you are deploying versioned resources, this attribute is ignored. The rules for the E(xist) flag apply to both Sentient- and non-Sentient distributed files. (For example, a Sentient-managed application file or CONFIG.SYS file.) Thus, you must specify Y or N for an existing file to be distributed. Y = The file will be distributed if one or more of the following conditions is met: - The file has not been distributed before. - The date, time, or size of the file has changed. - The file was re-promoted. N = The file will be distributed if: - The file has not been distributed before. - Or, The file was re-promoted. E = The file will be distributed only if it does not currently exist on the desktop.	Y	1 V
	M = Reserved for Sentient maintenance files. During the last step of the connect, the file is moved from the EDMDATA subdirectory to its proper location. Example: can be used to update EDMGETRS.EXE. S = This file is distributed if the file size has changed. T = The file is distributed if the file time has changed. D = File is distributed if the file date has changed. L = Forces the lock method to be used. The file will be automatically queued up to be deployed on restart, and the reboot flag will be set. If the L flag is not specified, and the file to be deployed is currently in use (locked) the lock method is automatically used anyway. What the L flag does is force the lock method to be always used		

Attribute Name	Explanation	Value of Base Instance	Length Type
	to deploy this file, whether or not it is currently in use (locked). The L flag will also skip decompression and dispatch the open error method (RADLKM). This is used for system dlls that must be replaced. Note that for the file to be verified during subsequent connects, you must also set this variable to Y. U = File is distributed only if the date and time on the CONFIGURATION SERVER is newer than the date and time on the desktop. R = Resource Stamp. For internal use only. V = (Windows only) Used in when normal date/time stamp check cannot be used. Usually when a .dll or .exe is involved and the date and time stamp are changing frequently. Radconct.exe checks the internal version number of the target file and compares it with the one that Sentient has. If the incoming file has a higher internal version number than it will be placed on the desktop over the existing file.		
	P = (Windows only) Set during packaging, enforces Windows System File Protection (SFP). If the verification is successful, the resource is in the desired state and no further verification is necessary. If the verification fails, Sentient checks to see if System File Protection protects the file. If the file is protected, it will not be deployed. If the ZRSCVRFY attribute is set to P and deployment of this resource and the service are marked as failed due to SFP processing. If you do not enforce SFP, the file is marked as having passed verification.		
ZSTOP	Stop Resolution Expression Expressions evaluating to "true" in ZSTOP attributes cause resolution of the instance to be skipped. If left blank, the instance is not skipped, and resolution continues. This is useful for assigning a version to a specific set of users. Use the CSDB Editor to set this attribute.		255 E
ZUPDATE	Client Instance Update Method If an object (resource instance) has been updated by an administrator, this method will be executed.		50 M

Attribute Name	Explanation	Value of Base Instance	Length Type
ZRUSERID	Admin ID		30 V

Client Methods (CMETHOD) Class

File:	PRIMARY
Domain:	AUDIT
Class:	CMETHOD
Icon:	
Description:	This class is used to configure method points for Tcl inventory scans. The base instance of the SCANNER Class is connected to the CMETHOD.INV_FULL instance. This instance can be used for all inventory scans defined in the SCANNER Class.
Can connect to:	RESEARCH

CMETHOD Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ALWAYS	Connect to Class connections determine the path of resolution of an agent's desired state. An _ALWAYS_ class connection is always resolved.		50 C
NAME	Friendly Name	Defaults	80 V
ZCREATE	Method to Install Resource Sentient-supplied method or fully qualified, user-defined method.	Hide nvdkit method	50 M
ZDELETE	Method to De-install Resource Sentient-supplied method or fully qualified, user-defined method.		50 M
ZINIT	Resource Initialization Method A fully qualified, user-defined method.		50 M
ZREPAIR	Client Instance Repair Method	Hide nvdkit	50 M

Attribute Name	Explanation	Value of Base Instance	Length Type
		method	
ZSTOP	Stop Resolution Expression Expressions evaluating to "true" in ZSTOP attributes cause resolution of the instance to be skipped. If left blank, the instance is not skipped, and resolution continues. This is useful for assigning a version to a specific set of users. Use the CSDB Editor to set this attribute.		255 E
ZUPDATE	Client Instance Update Method Hide nvdkit method	Hide nvdkit method	50 M
ZVERIFY	Client Instance Verify Method Hide nvdkit method	Hide nvdkit method	50 M

Desktop (DESKTOP) Class

File:	PRIMARY
Domain:	AUDIT
Class:	DESKTOP
Icon:	
Description:	This class is reserved for future use.
Can connect to:	CATALOG, ZSERVICE

DESKTOP Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ALWAYS	Connect to Class connections determine the path of resolution of an agent's desired state. An _ALWAYS_ class connection is always resolved.		50 C
JITARGS	Application Manager Arguments Stores the arguments to be passed to the just-in-time executable.		255 V

Attribute Name	Explanation	Value of Base Instance	Length Type
JITLNK	Application Manager Installation [Y/N]		1 V
JITPATH	Application Manager Path Stores the fully qualified name of the just-in-time executable. Should be a fully qualified path name with the drive and directory. If the directory name contains embedded spaces then the whole value should be in double quotes, e.g., "c:\program files\PSL\test.exe".		255 V
MACHUSER	Machine User		32 V
NAME	Friendly Name		80 V
VERIFY	Verify [Y/E/N]		1 V
ZCONTEXT	Component Context		8 V
ZCREATE	Method to Create Link Sentient-supplied method or fully qualified, user-defined method.	RADCLNK	50 M
ZDELETE	Method to Delete Link Sentient-supplied method or fully qualified, user-defined method. RADDLNK		50 M
ZINSTALL	Method CLNK Processing [Y/N] Stores if the links/icons have to be installed.		1 V
ZOBJPRI	Priority [00-99]	60	2 V
ZPUBVER	Publisher Version	_UNDEF_	8 V
ZRSCLFIL	Link Path Stores the fully qualified name for the link. The links are usually created in the Windows folder, which might appear in different drives for different users. The Windows system directory is usually stored in the ZMASTER object in ZWINDRV and ZWINDIR variables. A special variable, \$NTUSER\$, can be used to distinguish between different user directories. This variable is interpreted by the EDMCLNK. Create method to store the value of the current user.		255 V
ZRSCMO	Mandatory/Optional	O	1 V
ZSTATUS	Client Status	999	3 V
ZSTOP	Stop Resolution Expression Expressions		255 E

Attribute Name	Explanation	Value of Base Instance	Length Type
	evaluating to "true" in ZSTOP attributes cause resolution of the instance to be skipped. If left blank, the instance is not skipped, and resolution continues. This is useful for assigning a version to a specific set of users. Use the CSDB Editor to set this attribute.		
ZW95ARGS	Command Line Parameters Stores the arguments to be passed to the executable specified in the ZW95PATH field.		255 V
ZW95DESC	Icon Description A brief description for the icon/link.		60 V
ZW95HOTK	Hot Key Assigned by MS Windows		12 V
ZW95ICON	Icon File		128 V
ZW95INDX	Index of Icon		60 V
ZW95PATH	Command Line Stores the name of the executable associated with this icon/link.		128 V
ZW95SHOC	Visibility Flag [00000000/00000001]00000001		8 V
ZW95WDIR	Working Directory Stores the starting directory for the program associated with the icon or link. Should be a fully qualified path name with the drive and directory. If the directory name contains embedded spaces then the whole value should be in double quotes, e.g., "c:\program files\PSL".		120 V

File (FILE) Class

File:	PRIMARY
Domain:	AUDIT
Class:	FILE
Icon:	
Description:	Defines file scans, such as auditing system DLLs.

FILE Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ALWAYS	Connect to Class connections determine the path of resolution of an agent's desired state. An _ALWAYS_ class connection is always resolved.		50 C
ACTION	Report Flags [I,N,C,D,S,D,C] RIMDIFF method performs actions on the files discovered on the user's computer during the agent connect. Y configures RIMDIFF to perform the action. N configures RIMDIFF to not perform the action. The first four flags determine when to report that the files were found: Report on: Initial, New, Changed, Deleted Initial means that the file was found during the first scan of the agent computer. New means that the file was found during the current scan. The file was not present during the previous scan. Changed means that the file was present during the previous scan and is different from the file found during the current scan. Deleted means that the file was found during the previous scan. The file is not present for the current scan.	YYYYNNN	7 V
	The last three flags control the actions to take on the files detected during the current scan. Action to take on discovery: Send, Delete, Custom Send means to send the files to the Configuration Server and store them in the location indicated by the ZRSCVLOC attribute (see ZRSCVLOC in this table). Delete means to delete the files from the user's computer. Custom means to execute the method indicated in the CUSTOM attribute. YYYYNYN – Report whenever encountered and delete the files. NNYYNNN – Report when changed or deleted and take no action. NYYNYYN – Report when the files are new or changed. Then send and delete the files.		

Attribute Name	Explanation	Value of Base Instance	Length Type
GROUP	Group Name for Audited Component Optional way to identify a set of scan results. This may be useful for querying and reporting on the audited files from the database where audit results can be stored.		50 V
INTERNAL	Internal Name		32 V
LANGUAGE	Language		16 V
LOCATION	Location		255 V
NAME	Friendly Name	Defaults	80 V
ORNAME	Original File Name		32 V
OUTPUT	Output Object Name	FILEAUDT	8 V
PRODUCT	Product Name For more details, see "ZVERINFO " on the next page.		26 V
PRODVERS	Version For more details, see "ZVERINFO " on the next page.		16 V
SCANFOR	File Name or Pattern to Search for Indicate a fully qualified path and file name to search for. Wildcard characters are permitted.	C:**.*	80 V
TYPE	Scan Type [FILE] Scan different file locations. Available scans are Behavior Services, Desktop, File, Path, Registry, and WBEM.	FILE	8 V
VENDOR	Vendor		32 V
VERSION	Product Version		16 V
ZCRCINFO	Collect File CRC (Y/N)		1 V
ZINIT	Resource Initialization Method A fully qualified, user-defined method.		50 M
ZRSCCRC	Resource CRC From the promotion process if the CRC option was selected. Eight-character calculated hexadecimal value.		8 V
ZRSCMFIL	Manager Directory Location		80 V
ZRSCMEM	PDS Member Name		8 V

Attribute Name	Explanation	Value of Base Instance	Length Type
	This field is optional.		
ZRSCOBJN	Persistent Systems Object Name		8 V
ZRSCPADM	Admin ID		30 V
ZRSCSRC	Resource Source, i.e. Publisher		20 V
ZRSCSTYP	Service File Type Server file type. This can be either Binary or Text. The administrator does not set this.	BINARY	8 V
ZRSCVLOC	Virtual Manager Location The location on the Configuration Server where the files are stored because of the Send Action (see ACTION in this table). This variable needs to be configured when sending a file back to the Configuration Server. The variable should contain the name of the MGRVLOC instance that will be used to resolve the location to store the uploaded file. systemdrive:\Data\&(ZOBJPID)\&(name)	_BASE_ INSTANCE_	32 V
ZSTOP	Stop Resolution Expression Expressions evaluating to "true" in ZSTOP attributes cause resolution of the instance to be skipped. If left blank, the instance is not skipped, and resolution continues. This is useful for assigning a version to a specific set of users. Use the CSDB Editor to set this attribute.		255 E
ZVERINFO	Collect Extended Info [1/0] Collect extended information. Set value to 1 to collect additional information for a file. Set value to 0 to not collect additional information. In order for this data to be collected, the associated attribute must exist in the AUDIT.FILE Class template. You can limit the scan to only those files that have some particular values in their extended information. You do so by supplying a value (either 1 or 0) for any of the associated attributes in an AUDIT.FILE instance. This causes the scan to be filtered. Only those files whose extended information data element contains the value you specify in its associated	1	1 V

Attribute Name	Explanation	Value of Base Instance	Length Type
	attribute will be scanned. Extended file information consists of one or more of the following data elements. The associated attribute name for the data element is in parentheses: (VENDOR) – Seller of the file/product. (PRODUCT) – Name of item for which the file is a part. (PRODVERS) – Version of product of which file is a part. (ORGNAME) – Name of the organization. (INTERNAL) – Internal data element encoded in the file. (VERSION) – Version of the file. (LANGUAGE) – Language of the file.		
ZMD5INFO	Collect MD5 [Y/N]		1 V

File Scanner (FILESCAN) Class

File:	PRIMARY
Domain:	AUDIT
Class:	FILESCAN
Icon:	
Description:	Persistent Systems component class used to configure an inventory scan. Adding File Scanner components to an audit package creates instances of the FILESCAN Class.

FILESCAN Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
<u>_ALWAYS_</u>	Connect to Class connections determine the path of resolution of an agent's desired state. An <u>_ALWAYS_</u> class connection is always resolved.	CMETHOD .FILESCAN	50 C
DIFF	Difference [Y/N] Specifies if differencing is to be done or not. If DIFF = Y, then the information from the scanned files will be compared with the information from the	Y	1 V

Attribute Name	Explanation	Value of Base Instance	Length Type
	previous file scan.		
INCLUDES	Includes Connection		50 I
NAME	Friendly Name	Default	50 V
OUTPUT	Output Object Prefix Specifies the prefix to be used for the object names created. If OUTPUT=FILE, then FILEAUDIT, FILEPREV objects will be created on the agent computer.	FILE	4 V

File Scanner Filters (FILTER) Class

File:	PRIMARY
Domain	AUDIT
Class:	FILTER
Icon:	
Description:	Persistent component class used to configure an inventory scan. Adding FILE Scanner Filters components to an audit package creates instances of the FILTER Class.

FILTER Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ACTION	Report Flags [I,N,C,D,S,D,C] I – Initial (Used for file auditing only [not currently supported]) N – New C – Changed D – Deleted S – Send (upload to Configuration Server) D – Delete (not currently supported) C – Custom (not currently supported)	YYYYNNN	7 V
COMPRESS	Compress [Y/N]	N	1 V
DEPTH	# Subdirectory Levels to Scan Number of subdirectory levels to scan Values:	-1	2 V

Attribute Name	Explanation	Value of Base Instance	Length Type
	-1root directory and all of its subdirectories 0root directory only 1root directory and its files >1root directory and its files down to the specified depth		
DIR	Directory to Scan		255 V
EXCLUDE	Exclude Globe Pattern		255 V
INCLUDE	Include Globe Pattern	*	255 V
NAME	Friendly Name	Filter for File Scanner	50 V
ZRSCVLOC	RCS Directory Location Name of an instance in the PRIMARY.AUDIT.MGRVLOC Class that defines the location to place the uploaded scanned files. Default is RCA_UPLOAD.	RCA_UPLOAD	255 V

Inventory Options (RIMOPTS) Class

File:	PRIMARY
Domain:	AUDIT
Class:	RIMOPTS
Icon:	
Description:	Contains the attributes that offer options that control an inventory management task.

RIMOPTS Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
COLLECT	Audit Collection Type [Diff/Full] Select Diff to report difference between the previous information collected for the service and the information collected during the current client audit. This is the default setting. Select Full to report the information collected for the service during the current agent connect process without differencing	Diff	7 V

Attribute Name	Explanation	Value of Base Instance	Length Type
	against the previous collection.		
NAME	Friendly Name Name displayed for the instance in the tree view of the CSDB Editor.	Default	80 V
RUNEXEC	Run Execute on [I,V,U] Indicated what actions the Inventory Manager will take upon connection. I – invokes collection of information when the service is installed (default). U – invokes collection of information when the service is updated (default). V – invokes collection of information when the service is verified.	IU	10 V
ZSVCTYPE	Service Type Contains code that is used internally by the Inventory Manager agent. In all cases, this values should remain set to I.	I	1 V

Inventory Scanners (SCANNER) Class

File:	PRIMARY
Domain:	AUDIT
Class:	SCANNER
Icon:	
Description:	This persistent component class is used to configure an inventory scan. Create instances of the SCANNER Class by adding Inventory Scanners components to an audit package.
Can connect to:	CMETHOD

SCANNER Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ALWAYS	Connect To Class connections determine the path of resolution of an agent's desired state. An _ALWAYS_ class connection is always resolved.	AUDIT.CMETHOD .INV_FULL	50 C

ACTION	Report Flags [I, N, C, D, S, D, C] Report on: Initial, New, Changed, Deleted, Scan, Delete, Custom Initial means that the file was found during the first scan of the agent computer. New means that the file was found during the current scan. The file was not present during the previous scan. Changed means that the file was present during the previous scan and is different from the file found during the current scan. Deleted means that the file was found during the previous scan. The file is not present for the current scan. Scan means that the file was found during the current scan. Delete means that the file was found during the previous scan. The file is not present for the current scan. Custom means that the file was found during a custom scan.	YYYXXN	7 V
CLASS	Class		50 V
DIFF	Difference [Y/N] Specifies if differencing is to be done or not. If DIFF = Y, then the information from the scanned files will be compared with the information from the previous file scan.	Y	1 V
KEYS	Class Keys		50 V
NAME	Friendly Name		80 V
NAMESPACE	Name Space	RCA	80 V
OUTPUT	Output Object Prefix	WBEM	4 V
PARMS	Parameters		50 V
SCHEMA	Schema[Table is <schema>_<class>]	SITE	16 V
TYPE	Scan Type	WBEM	4

			V
URL	Post Audit Data to URL		80 V7

Path (PATH) Class

File:	PRIMARY
Domain:	AUDIT
Class:	PATH
Icon:	
Description:	This class stores the drive and directory required to install a resource. Packages can be relocated by updating instances of this class.
Can connect to:	DESKTOP, FILE, PATH, MSIFEATS, MSI, PACKAGE, PDACONFIG, REGISTRY

ZSERVICE Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
DIR	Directory		255 V
DRIVE	Drive		20 V
NAME	Friendly Name		255 V
ZRSCMO	Mandatory/Optional [M/O]	M	1 V
ZRSCPRI	Priority	5	2 V

Registry (REGISTRY) Class

File:	PRIMARY
Domain:	AUDIT
Class:	REGISTRY
Icon:	
Description:	The REGISTRY Class uses WMI to obtain a Registry scan of a Windows machine.

REGISTRY Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ACTION	Report Flags (I, N, C, D, S, D, C) Y, X, or N for each flag.	YYYYXXN 7.	7 V
CLASS	WBEM Class Do not change	StdRegProv	50 V
CNDITION	Registry subkey Any Windows registry subkey.	SOFTWARE \Microsoft \Internet Explorer	255 V
DEPTH	Starting at the registry subkey named in the CNDITION attribute, depth specifies the number of descendent key levels to include in the scan: 0 = Only scan current subkey - 1 = Scan all subkey levels 1 to n = Scan current subkey and the specified subkeys levels deep	0	2 V
FORMAT	Output format Do not change.	REGISTRY	8 V
NAME	Friendly Name Friendly name for this instance displayed in CSDB Editor.	Default	80 V
NAMESPACE	Name Space Do not change.	root\default	80 V
OUTPUT	Output Object Name	WBEMAUDT	8 V
PROPERTY	Registry hive Any Windows registry hive: HKEY_CLASSES_ROOT HKEY_CURRENT_USER HKEY_LOCAL_MACHINE HKEY_USERS HKEY_CURRENT_CONFIG HKEY_DYN_DATA	HKEY_ LOCAL_ MACHINE	255 V
RPTCLASS	Report Class Name A valid table name. If blank "StdRegProv" will be used.	Registry	32 V
TYPE	Scan Type (WBEM) Do not change.	WBEM	4 V

Scheduling (TIMER) Class

File:	PRIMARY
Domain:	AUDIT
Class:	TIMER
Icon:	
Description:	This class contains the instances that enable the Sentient administrator to set a timer on agent computers. One or multiple auditing services can be processed whenever the timer expires.

TIMER Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ALWAYS	Connect to Class connections determine the path of resolution of an agent's desired state. An _ALWAYS_ class connection is always resolved.		50 C
APPSVC	Application		32 V
DOMAIN	Server Domain Name	&[ZMASTER.ZDOMNAME]	32 V
IPADDR	Server IP Address/Name	&[ZMASTER.ZIPADDR]	32 V
MGRNAME	Server Name	&[ZMASTER.ZMGRNAME]	32 V
NAME	Friendly Name		80 V
NETAVAIL	Check Network Availability [Y/N/W] Checks for network availability before executing the command.		1 V
OBJGUID	Instance GUID: Object Global Unique Identification – for future use.	_UNDEF_	32 V
PINGCNT	Number of Ping Attempts This attribute sets the number of ping attempts to be made by the Configuration Server.		32 V
PINGDLAY	Time between Pings (in msec) This attribute sets the amount of	2000	6 V

Attribute Name	Explanation	Value of Base Instance	Length Type
	time between pings in msec.		
REQUEST	Application Request	"Apply Updates"	32 V
RETRYFLG	Enable Retry Support [Y/N]		1 V
RETRYINT	Time between Retries (in minutes)		3 V
RETRYLMT	Number of Retry Attempts		3 V
RETRYRC	Valid Retry RC's (Blank = 200)		32 V
RUNSYNC	Synchronous Timer Execution [Y/N] This attribute sets the value of Yes or No for the synchronous timer execution. The default value is Yes.	Y	1 V
SOCKET	Server Socket Number	&[ZMASTER.ZDSTSOCK]	32 V
ZCHNNAME	Channel [DOMAIN] Name &(ZSERVICE.ZOBJDOMN)		32 V
ZCREATE	Client Create Method Sentient-supplied method or fully qualified, user-defined method. This value is inherited from the base instance and should not be changed.	radtimeq	80 M
ZDELETE	Client Delete Method Sentient-supplied method or fully qualified, user-defined method. This value is inherited from the base instance and should not be changed.	radtimeq	80 M
ZNOPING	Stop RCS Detection [Y/N/W/S/P] Controls the automatic sensing of a network connection between the agent computer and the Configuration Server. An expired time will continually evaluate whether communications with the Configuration Server can be established. If set to S or P , additionally, it will ping SAP entries according to	Y	1 V

Attribute Name	Explanation	Value of Base Instance	Length Type
	their priorities, along with the Configuration Server. When communications are established, the command line associated with the time is executed. After executing the command line, the Scheduler service resumes normal evaluation of whether the timer has expired again. This attribute is especially useful for mobile users. Note: In order to use this attribute, you must add it to the TIMER Class template.		
ZOBJPRI	Priority [00-99] Sets the priority for deployment of the ZTIMEQ object. The ZTIMEQ object is deployed relative to the other elements being deployed during the agent connect. The elements with a priority number less than the value of ZOBJPRI are deployed before the ZTIMEQ object. A value of 90 is inherited from the base instance and should not be changed.	90	2 V
ZPRVNAME	Provider (Manager) Name	&(ZSERVICE.ZMGRNAME)	32 V
ZRSCCMDL	Command Line to Execute This attribute indicates the command line that is executed on the subscriber's computer when the timer expires.	radskman req=	255 V
ZSCHDEF	Time Parameter Indicates when the timer expires. The syntax varies depending on the frequency of expiration that can be DAILY, HOURLY, INTERVAL, NUMDAY, WEEKDAY, WEEKLY.		50 V
ZSCHFREQ	Frequency	PERIODIC	10 V

Attribute Name	Explanation	Value of Base Instance	Length Type
	[PERIODIC/ONCE/RANDOM] This attribute indicates how often the timer should expire according to the frequency specified in the ZSCHDEF attribute. Once for a one-time expiration. Periodic for a repeated expiration. Random for random intervals.		
ZSCHMODE	Time Owner [MANAGER/DEFAULT] This attribute specifies the timer owner. It is recommended that you leave the timer owner configuration value as DEFAULT.	DEFAULT	32 V
ZSCHTYPE	Type [IMMEDIATE/DEFERRED] Used only when ZSCHFREQ = PERIODIC. Set ZSCHTYPE to DEFERRED to indicate that the first time you attempt to launch an event, it will be deferred until the next scheduled time, no matter when the timer instance is evaluated. This will handle a daily 4 AM (non-peak) scheduled event that is sent to the agent computer during the day. If it was not deferred, it would launch during the day instead of "waiting" until the next morning. Example 1: If you create and deploy a timer with the ZSCHDEF = DAILY (&ZSYSDATE,4:00:00) If ZSCHTYPE = IMMEDIATE and it is: Before 4:00:00, the command in the instance will be executed the same day at 4:00:00 After 4:00:00, the command in the instance will be executed immediately If ZSCHTYPE = DEFERRED and it is: Before 4:00:00, the command in	DEFERRED	10 V

Attribute Name	Explanation	Value of Base Instance	Length Type
	the instance will be executed the next day at 4:00:00 After 4:00:00, the command in the instance will be executed the next day at 4:00:00 Example 2: Suppose you create and deploy a timer with the ZSCHDEF = WEEKDAY(FRIDAY,4:00:00) If ZSCHTYPE = IMMEDIATE and it is: Not Friday or Friday and before 4:00:00, the command in the instance will be executed on Friday at 4:00:00 Friday and after 4:00:00, the command in the instance will be executed immediately If ZSCHTYPE = DEFERRED and it is: Not Friday or Friday and before 4:00:00, the command in the instance will be executed a week later on Friday at 4:00:00 Friday and after 4:00:00, the command in the instance will be executed a week later on Friday at 4:00:00		
ZSTOP	Stop Resolution Expression Use to assign timer conditions. Indicate true to cause resolution of the instance to be skipped. The timer is not deployed for end users. Leave blank for the instance to be accepted, and resolution will continue.		255 E
ZSVCOID	Service ID This attribute specifies the object ID of the Application instance that this Scheduling instance is connected to. This value is inherited from the base instance and should not be modified.	&(ZSERVICE.ZOBJID)	80 V
ZUPDATE	Client Update Method The Scheduler UPDATE method	radtimeq	80 M

Attribute Name	Explanation	Value of Base Instance	Length Type
	that runs on the agent computer. This value is inherited from the base instance and should not be changed.		
ZVERIFY	Client Verify Method	radtimeq	80 M

UNIX Permissions (UNIXPERM) Class

File:	PRIMARY
Domain:	AUDIT
Class:	UNIXPERM
Icon:	
Description:	UNIX permissions.

UNIXPERM Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ZSTOP000	<ZSTOP000>	WORDPOS(EDMGETV (ZMASTER,ZOS), 'UNIXHPUX UNIXSOL UNIXAIX UNIXLNUX')=0	255 E
ZRSCRASH	<ZRSCRASH>	0666	4 V
NAME	<NAME>	Base Instance	80 V

Virtual Mgr Location (MGRVLOC) Class

File:	PRIMARY
Domain:	AUDIT
Class:	MGRVLOC
Icon:	
Description:	This class is used to specify the initial path for files being transferred to the Configuration Server during a FILE audit.

MGRVLOC Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ZRSCMFIL	Audit File Path		255 V

WBEM (WBEM) Class

File:	PRIMARY
Domain:	AUDIT
Class:	WBEM
Icon:	
Description:	This class contains instances that define Inventory Manager scans of WMI Classes. These can include any class in the WMI database such as Win32_Services.

WBEM Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ACTION	Report Flags [I, N, C, D, S, D, C] The RIMDIFF method performs actions on the WBEM namespaces (s) instances discovered on the user's computer during the agent connect. • Y configures RIMDIFF to perform the reporting action. • N configures RIMDIFF to not perform the reporting action. • The first four flags determine when to report that the WBEM namespace instance was found: • Report on: Initial, New, Changed, Deleted, Scan, Delete, Custom • Initial means that the file was found during the first scan of the agent computer. • New means that the file was found during the current scan. The file was not present during the previous scan. • Changed means that the file was present during the previous scan and is different from the file found during the current scan.	YYYYXXN	7 V

Attribute Name	Explanation	Value of Base Instance	Length Type
	- Deleted means that the file was found during the previous scan. It is not present for the current scan. - Scan means that the file was found during the current scan. - Delete means that the file was found during the previous scan. It is not present for the current scan. - Custom means that the file was found during a custom scan. The last three flags are not applicable to WBEM audits.		
CLASS	WBEM Class The name of the WBEM Class to query or HARDWARE.		50 V
CNDITION	Query Condition Statement An optional condition to narrow results of an audit.		255 V
NAME	Friendly Name This name will appear in the CSDB Editor's tree view to identify this instance.	Default	80 V
NAMESPACE	Name Space The name of the WBEM namespace to query or HARDWARE.	root\cimv2	80 V
OUTPUT	Output Object Name This is the name of the object to send to the Configuration Server.	WEBMAUDY	8 V
PROPERTY	Property Values to Collect Specify one or more property names to be queried and reported. Use commas to separate more than one property name. If this attribute is blank, all properties in the class will be queried and reported.	*	255 V
RPTCLASS	WBEM Reporting Class		50 V
TIMEOUT	Query Timeout (Seconds)		4 V
TYPE	Scan Type [WBEM] Indicates that WBEM scan is to be employed for this audit package.	WEBM	4 V

The CLIENT Domain

Use the CLIENT Domain to configure Client Operations Profiles, including setting access points for application data. For more information, see the *Sentient Enterprise Console Online Help*. To read the class reference tables, see the section "[How to Read the Class Reference Tables](#)" on [page 861](#).

Alert Management (RADALERT) Class

File:	PRIMARY
Domain:	CLIENT
Class:	RADALERT
Icon:	
Description	This class is reserved for future use.

RADALERT Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ALRTDISP	Display WMI Events Locally [Y/N] Specify Y to locally display WMI events.	N	1 V
DISPCTG	CSV Event Categories to Display		200 V
DISPSEV	Minimum Alert Severity to Display	0	3 V
REPCTG	CSV Event Categories to Report		200 V
REPSEV	Minimum Alert Severity to Report	0	3 V
TIMEOUT	Timeout for Local Event to Display	5	3 V
WMIALRT	Monitor WMI BIOS Events [Y/N] Specify Y to monitor WMI BIOS events.	N	1 V
SMRTMON	Specify Y to monitor SMART events.		

Attribute Name	Explanation	Value of Base Instance	Length Type
SMRTMON	Specify Y to locally display SMART events.		
SMRTREP	Specify Y to report SMART events.		

Connect Defer Prefer (CDFCFG) Class

An administrator can use the Connect deferral function to configure options for downloading and installing mandatory service-related actions. It gives you control to defer the service-related actions. Use the CDFCFG class in the Sentient Administrator CSDB Editor to configure the user-facing Connect Deferral dialog box. For more information on Connect Deferral, see the *Sentient Enterprise Console Online Help*.

File:	PRIMARY
Domain:	CLIENT
Class:	CDFCFG
Icon:	
Description:	Use this class to configure the user-facing Connect Defer dialog box.

CDFCFG Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ABORT	Allow / Disallow abort [Y/N] Specify Y (the default) to enable the Cancel button. If N is specified, the Cancel button will be disabled and the text that describes the cancel feature ("Click Cancel to cancel this process without rescheduling.") will be hidden.	Y	1 V
DESCTEXT	Descriptive text for CDF dialog DESCTEXT will replace the first two sentences of text.		255 V
DESCTXT2	More descriptive text for dialog DESCTXT2 will replace the text "To continue with these actions...".		255 V
DOMAINS	Allowable domains for CDF The Domains attribute uses a comma-delimited list of domains for Connect Deferral. The default value is SOFTWARE,PATCHMGR,OS. For	SOFTWARE, PATCHMGR, OS	255 V

Attribute Name	Explanation	Value of Base Instance	Length Type
	example, if <code>SOFTWARE</code>, OS is specified, the <code>PATCHMGR</code> connects are not deferred. You can also add custom domains to the Domains attribute. This enables you to defer services in the custom domains and associate them to a single display name. For example, if you have custom domain names such as, <code>CUS_DOMAIN1</code>, <code>CUS_DOMAIN2</code>, <code>PATCH1</code>, and so on, you can categorize and associate these to any domain and define the Domains attribute value as: <code>SOFTWARE (CUS_DOMAIN1, CUS_DOMAIN2), PATCHMGR (PATCH1), OS</code>. The display name for any service deferred from the <code>CUS_DOMAIN1</code> will be <code>SOFTWARE</code>. You can also add filters to the domain names. For example, if the specified Domains attribute value is <code>SOFTWARE (CUS*) , PATCHMGR (PATCH*) , OS</code>, the display name for all domains starting with <code>CUS</code> and <code>PATCH</code> will be <code>SOFTWARE</code> and <code>PATCHMGR</code> respectively. Formatting rules for the Domains attribute are: • Every opening parenthesis must have a matching closing parenthesis and also in the reverse order. For example, <code>SOFTWARE (CUS_DOMAIN1, CUS_DOMAIN2), PATCHMGR (PATCH1</code> is invalid. • Use filters for domain names inside the parenthesis. You cannot use filters for domain display names. For example, <code>SOFTWARE (CUS_*) , PATCHMGR (PATCH1) , OS</code> is valid, <code>SOFT* (CUS_DOMAIN1, CUS_DOMAIN2), PATCHMGR (PATCH1) , OS (OS1, OS2)</code> is invalid. • Domain names must be comma-delimited and all mapping of domains must be done within parenthesis. • The maximum character length of the Domains attribute is 255. • The Domains attribute is case insensitive. Valid usage example for the Domains attribute is, <code>SOFTWARE (CUS_DOMAIN1, CUS_DOMAIN2), PATCHMGR (PATCH1), OS</code>.		

Attribute Name	Explanation	Value of Base Instance	Length Type
	Note: Make sure you follow the formatting rules for the Domains attribute. If an incorrect value is used, Connect Deferral may not function in the desired way.		
ENABLE	Enable / Disable CDF [Y/N] Specify Y (the default) to enable the connect-deferral function. Note: RADSKMAN has been updated to include a new command line option, <code>cdf=y/n</code>, which enables/disables this function. The RADSKMAN setting will supersede this CDFCFG setting.	Y	1 V
NAME	Friendly Name		32 V
OS, PATCHMGR and SOFTWARE	Max days since last os conn Max days since last patch conn Max days since last software conn Specify the maximum number of days that a user can defer a connect for each of the supported domains. For example, if SOFTWARE is set to 5, the user can defer a SOFTWARE connect for a maximum of 5 days; after that, the pending action will be forced on the user's machine. PATCHMGR and OS are for specifying the maximum days that a connect for those domains can be deferred. A local value is saved in the CDFDEFER object in <code>IDMROOT</code> to determine when the user started deferring. (If the user clicks Allow, this date is reset to 0.) On the next connect, this value is queried in order to determine how many "deferral" days remain. For example, if the action is deferred on Monday and SOFTWARE is set to 5, on Tuesday the message will indicate 4 "deferral" days remaining, and so on, until the number of days reaches zero. When there are no remaining "deferral" days, Cancel and Defer will be disabled and the user will have to allow the connect. Notes: These values will affect the "Defer for" list; intervals that are greater than this setting will not be available in the drop-down list.		3 V

Attribute Name	Explanation	Value of Base Instance	Length Type
	On the right side of the window, a message will indicate the remaining number of days that the user can defer the actions. Connect Deferral looks at dname= to figure out which value in CDFCFG to use. Custom domains in an Sentient Configuration Server Database can be added to this list so that users can defer services in these domains also. The default value is SOFTWARE if dname is not specified.		
TIMEOUT	Dialog timeout in minutes Specify the length of time (in minutes) for the Connect Deferral dialog to wait before automatically triggering the pending action.	5	3 V
DEFAULT	Default max days since last conn Specify the maximum number of days that a user can defer a connect for domains not specified in the CDFCFG class. For example, when a connect is run for any domain that is not specified in the CDFCFG class, Connect Deferral reads the DEFAULT attribute value as the number of days for deferral.		3 V
VRFYONLY	Specify Y to display the connect deferral dialog on verify only connect. The default value of this variable is N .		
ABTNTXT	Specifies an alternative text for the Allow button. By default, this button is labeled 'Allow'. This option should only be used for the special case where prior customized dialogs used different verbiage for connect deferral actions.		
CBTNTXT	Specifies an alternative text for the Cancel button. By default, this button is labeled 'Cancel'. This option should only be used for the special case where prior customized dialogs used different verbiage for connect deferral actions.		
DBTNTXT	Specifies an alternative text for the Defer button. By default, this button is labeled 'Defer'. This option should only be used for the special case where prior customized dialogs used different verbiage for connect deferral actions.		

Core and Satellite Servers (SERVER) Class

File:	PRIMARY
Domain:	CLIENT
Class:	SERVER
Icon:	
Description:	Each instance is a Core or a Satellite, automatically created and managed by the Core and the Satellite server. There is a bi-directional link to and from the Portal for modeling. Each instance contains a direct connection to its managed SAPs. Any manual changes to the SERVER class will be lost during the next automatic Satellite management update process. Changes should be made to the Satellite server directly, which will then update its respective SERVER and SAP instances. <i>It is not recommended to make manual changes directly to the CSDB. All changes should be performed through the Sentient Core Console. If manual changes are required, they should be performed under the direction of Persistent Systems Support. Manual changes could impact UI operation.</i>

SERVER Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
NAME	Display name The display name of the Core or the Satellite server.		80 V
DESCRIPT	Description The description of the Core or the Satellite server.		255 V
TYPE	Server type This value is automatically filled in for the Core or the Satellite server, based on the server type.		32 V
ENABLED	Enable server [Y/N] Enable or disable this server. When disabled, the client does not use this server or its SAPs for resources.	Y	1 V
DEVDN	Portal device DN This value defines how the server is represented in the Portal.		100 V

Attribute Name	Explanation	Value of Base Instance	Length Type
HOSTNAME	Fully qualified hostname This value is automatically set to the fully qualified hostname of the Core or the Satellite server.		100 V
ALWAYS	Server parameter override method The method <code>SYSTEM_ZMETHOD_SERVER_PARAM_OVERRIDE</code> is called to check whether the server should be part of the resolution or not.	SYSTEM_ZMETHOD_SERVER_PARAM_OVERRIDE	100 M
ALWAYS	Connect to SAP This instance always connects to the SAP instance.		100 A

Settings (SETTINGS) Class

Use an instance in the SETTINGS Class to define defaults for Server Access Profiles defined as part of satellite management. These defaults could be overwritten based on configuration defined as part of Satellite management in the Core console. SETTINGS can also define scripts you want to use in pre-configuration processing, and to set other global parameters.

File:	PRIMARY
Domain:	CLIENT
Class:	SETTINGS
Icon:	
Description:	Use an instance in the SETTINGS Class to define how to use your Server Access Profile, define scripts you want to use in pre-configuration processing, and other global parameters.

SETTINGS Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ACTMAINT	Maintenance Activation [I/D] Set to I to immediately run the Sentient maintenance module, Upgrdmaint. It is launched to activate the staged maintenance by moving the maintenance modules into their live	I	

Attribute Name	Explanation	Value of Base Instance	Length Type
	locations. Upgrdmaint can be launched by radskman immediately after the maintenance is staged or on an independent schedule. This will be controlled by a variable in COP called ACTMAINT (set to Immediate or Deferred). If radskman launches Upgrdmaint, it should only do so after it first checks if there is any new maintenance in the staging directory		
ADINFO	Query Active Directory Info [Y/N] Specify Y if you want to collect the agent computer's active directory information. The information is stored in the ADINFO object in the RADSETUP directory. The default location for the RADSETUP directory is <code><InstallDir>\Lib\RADSETUP</code> . This information will be sent to the Configuration Server for all resolution processes.	Y	1 V
ALWAYS	Always Download CFG Objects [Y/N] Set to Y to always download pre-configuration objects. This guarantees that your SAP or persistent objects are downloaded even if nothing has changed. If your SAP client object is corrupted for any reason, then it will be re-downloaded even if the desired state did not change. In addition, if one of the variables is a substitution then you will download the object with the new values since a variable change by substitution does not change the desired state.	Y	1 V
ALWAYS	Always Upload CFG Objects [Y/N] Always upload all objects in the RADSETUP directory. The default location for the RADSETUP directory is <code><InstallDir>\Lib\RADSETUP</code>	Y	1 V
BANDWIDTH	Bandwidth Percentage (1-99) Specify the percentage of bandwidth to use between 1 and 99. If the value is blank or the variable does not exist, then all of the bandwidth will be used.		2 V
CMETHOD	Post Catalog Script Specify a script that can run after catalog resolution, but before service processing		
DEFROLE	Default SAP ROLE (A,O,S,M,R)		

Attribute Name	Explanation	Value of Base Instance	Length Type
	Specify roles for the Configuration Server specified on the command line. If not specified, the ROLE is set to A (All), and the Configuration Server will be able to perform any ROLE. Note: USEDEFS must be set to Y to use DEFROLE.		
DETPROXY	Internet Proxy detection [Y/N] Set to N to skip running Internet proxy detection at the beginning of the agent connect.		
DISKFREE	Minimum Free Disk Space Threshold Specify a minimum of free disk space for Sentient to maintain. If a service is over the limit, it will not be installed.		
DTIMEOUT	Radiamsi Driver Timeout (1 - 4080) Seconds Configure the Radiamsi driver timeout (1 - 4080 seconds) if Radiamsi driver fails to download large files.	0	4 V
EQUISORT	Secondary SAP Priority [R/S] If several SAP instances have the same priority, set this to R to randomly select which one to use. Set to S to use the SAP with faster network speed. SAPPING must be set to Y to use EQUISORT= S. Set to R (the default) to randomly select which SAP instance to use. This is recommended for workload balancing.	R	1 V
EXASETUP	Post Config Resolution Script Specify a script to run after pre-configuration processing. This script must be in the agent computer's IDMSYS directory. The default location is <InstallDir>\Agent.	INITMETH .REX	80 V
EXBEXIT	Post Connection Script Specify a script to execute before RADSKMAN ends. If you are doing a customized reboot process, this is where you would specify it. This script must be in the agent computer's IDMSYS directory. The default location is <InstallDir>\Agent. Note: Client Operations Profiles must be enabled on the Sentient agent for the EXBEXIT		80 V

Attribute Name	Explanation	Value of Base Instance	Length Type
	to be used. This script executes only for Sentient Application Manager (RAM) connect.		
EXBOUTBX	Pre Outbox Script Specify a script that can run after service processing, but before the objects in the outbox are flushed to the Configuration Server		
EXBSETUP	Pre Config Resolution Script Specify a script to run before pre-configuration processing. This script must be in the agent computer's IDMSYS directory. The default location is <i><InstallDir>\Agent</i> .	PRESETUP .REX	80 V
FLUSHU	User Reporting Option [Y/N/A] Supports flushing objects for all users during a connect or saving the objects locally for transfer at a later time. A All Users Y Always flush the outbox (default) N Never flush the outbox Sample usage: Specify FLUSHU=N on user connects to build up reporting objects in each user's outbox folder. During a machine connect specify FLUSHU=A to transfer all objects. The default (Y), will always send the current connecting client's reporting objects from the outbox folder.	Y	1 V
LSCRIPT	Disable Connect on UI Reboot [Y/N] If you have set a service to perform an immediate reboot and you run RADSKMAN from a login script, set this to Y to run RADSKMAN from the login script. If you have set a service to perform an immediate reboot, and you want radskman to be restarted in the User context when a user logs, set this to N. In other words, if your users are configured to connect to Configuration Server when they log in, set this to N. If you want to do an immediate reboot with context=u, and you want the user to re-establish connection with the Configuration Server, then set LSCRIPT=Y.	Y	1 V

Attribute Name	Explanation	Value of Base Instance	Length Type
NAME	Friendly Name		50 V
NETSPEED	Speed Check Method C = Detect using ICMP and Link Speed as failsafe (Default - original behavior) C will first try ICMP and fall back to max. M =Use the max value of both detect methods. Max will bypass ICMP checking and enumerate the available network connections. Reports the highest speed regardless of loading/latency. N =Do not run netspeed checking. (Will not display error). H = Perform a high-resolution speed calculation that differentiates between SAPs (for COP) on a network. Uses ICMP. W = Detect using ICMP, with improved algorithm for calculating the network bandwidth.		1 V
NETTTL	Network Time to Live (0-999) Controls the maximum number of network segment hops allowed when calculating the speed of network connections. If NETTTL is not available or blank, the default value of 3 is used.		3 V
PUSHBACK	Push Back (0-999 retries) Set to 0 to skip a Configuration Server if the Configuration Server pushes back on the agent connect. Set to 1 to 999 for number of retries if the Configuration Server pushes back.	0	1 V
RAD2XUI	Enable RADUI 2.x Specify Y to view the vintage UI dialogs. Use this if you are not using System Tray or if you want a message to pop up on the screen in addition to the System Tray.		
RADTRAY	Radtray Command Line Arguments Set command line arguments you want to use for the System Tray. Specify Y for the first argument to enable the System Tray, and N to disable it. If set to Y, you can then specify other parameters separated from the Y with a comma. Possible parameters are: /C = Show the System Tray in console mode when it starts /NOCANCEL = Hide the Cancel button.		

Attribute Name	Explanation	Value of Base Instance	Length Type
	/NOPAUSE = Hide the Pause button. /D = Add debug message to the log for troubleshooting. Example: Y, /C /NOPAUSE enables the System Tray in console mode and does not display the PAUSE button.		
RCSDATA	Download DATA from RCS [Y/N] After using all of the <code>TYPE = DATA SAPS</code>, if all the needed data has not been downloaded then specify Y to go to SAPs with <code>TYPE = RCS</code>. If you do not want the agent computers to use Configuration Servers, set RCSDATA to N. For Multicast to work, the RCSDATA value should be set to Y.	Y	1 V
REMUNINS	Allow Remote Notify Uninstall [Y/N/A] Specify Y to stop notifies from remote machines from un-installing a service. Specify A to stop notifies from remote and local machines from un-installing a service. This does not stop applications from being un-installed as part of a policy change if a standard Sentient agent connect is started from a remote notify. The remove notify string must contain the text <code>req="Un-install"</code>. Specify N to stop remote machines from un-installing a service.		
RSTROPT	Bandwidth Checkpoint Restart Use this attribute to determine when a file is eligible for checkpoint restart based on calculated network bandwidth. This will apply to all files to be downloaded in this agent connect. Specify eligibility in the format (Below Threshold limit, Network Threshold Value, Above Threshold). Suppose you set RSTROPT to 100 KB, 86 KB, 10 MB. First the client calculates the network bandwidth. One of two scenarios will apply: If the network bandwidth is under 86 KB, the file size is compared to 100 KB. If the file size is over 100 KB, checkpoint restart is enabled for that file. If the network bandwidth is over 86KB, the file size is compared to 10MB. If the file size is over 10MB, checkpoint restart is enabled for that file.		

Attribute Name	Explanation	Value of Base Instance	Length Type
RTBWMGMT	Real-time Bandwidth Management Use this attribute to enable the Agent machine to adapt to real time available bandwidth. The bandwidth is calculated based on the Frequency and Percentage parameters. The bandwidth can be considered real-time, if you specify a short duration in frequency. By default, this attribute is disabled. This attribute has the following parameters: • Enable: Describes whether real-time bandwidth management is enabled or not. Default value is N. Specify Y to enable the real-time bandwidth management. • Frequency: Number of seconds for one cycle to calculate the allocated bandwidth from the available bandwidth and percentage value. Default value is 15. • Percentage: Specify the percentage of bandwidth to use from the calculated available bandwidth. Default value is 40. For example, you can specify the values for this parameter as Enable=Y, Percentage=40, Frequency=15.		200 V
SAPPING	Ping all SAP [Y/N] Set to Y if the client should ping all of the SAPs. If EQUISORT is set to S, then you must set SAPPING to Y. A result reflecting the speed of the connection will be returned and stored in the SPEED attribute in the SAPSTATS object	N	1 V
SENDERPT	Send Reporting Object [I/D] Set to D to defer sending all reporting objects to Configuration Server at the end of agent connect. Usually, the reporting objects for each service, such as APPEVENT, CLISTATS, and ZSVCSTAT, are sent to the Configuration Server immediately (I) after they are created. This requires multiple disconnects and reconnects to the Configuration Server.	I	
SLFHEAL	Agent Self-Healing [N/Y/X]. Specify Y to enable self-healing of the corrupted agent object. Specify X to enable self-healing and to display report of the corrupted agent object. Default	N	1 V

Attribute Name	Explanation	Value of Base Instance	Length Type
	value is N .		
THROTYPE	Throttle [RESERVED/ADAPTIVE/NONE/] Type of Bandwidth throttling to use. Set to ADAPTIVE to yield to other services that are using the network. Set to RESERVED to allow for a specific reservation of the bandwidth. It is the maximum percentage of network bandwidth to use. Set to NONE for no bandwidth throttling, and use the maximum available bandwidth. NONE is the default.		8 V
TIMEOUT	Communications Timeout (0-3200)s Specify the timeout in seconds for the Server Access Profile (SAP). This will override the agent timeout (ZMASTER.ZTIMEO) if it contains a valid numeric value. If the value is blank, then the agent will use the existing timeout value on the Agent.		4 V
USEDEFS	Use Default SAP [Y/N] If a SAP cannot be found for the needed ROLE, specify Y to default to the Configuration Server set on the command line.		
USELSAP	Use Last SAP [N/Y] Set this Y to specify that the last SAP used in this agent connect should be the SAP used for all remaining services to be resolved. Use of SAPs with type of DATA is at the service level. If set to N, then the client will go through the SAPs in priority for each service. During an agent connect, if a service has to go to a lower priority SAP to complete the data download, decide if you want the remaining services to continue from this SAP (USELSAP=Y) or go back to the highest priority DATA SAP to search for files for the next service (USELSAP=N) and to continue through the SAP priorities.	Y	1 V
ZGRPINFO	Query NT User Group Info [Y/N] Specify Y if you want to collect the agent computer's Windows NT user group information. This information will be reflected in the NTGROUPS object in the RADSETUP directory. The default location for the	N	1 V

Attribute Name	Explanation	Value of Base Instance	Length Type
	RADSETUP directory is <i><InstallDir>\Agent\Lib\RADSETUP</i> . This information will be sent to the Configuration Server for all resolution processes.		
NATVHTTP	Use Windows HTTP [Y/N] Specify this attribute to enable the Sentient agent to use Windows WinHTTP library for HTTP communication. Specify Y to use Windows WinHTTP library. Specify N to use RCA HTTP library.	N	1 V
SALVAGE	Re-stamp same files [Y/N] Allows the agent to restamp and reuse resources that have the same signature but different date and time stamp. The checksig=Y parameter added to the RADSKMAN command line instructs the agent to verify the resources' signatures. The packages do not need to be published in a special way to use this flag. This flag is for the agent only, the salvage is always on for staging clients. This will provide savings for packages that have resources with the same signature but different date/time stamps. The salvage flag can only be set in the System Explorer.	N	1 V
RALERTEN	Use Alert Management [Y/N] Specify this attribute to enable or disable alert management (RADALERT). To enable the alert management, set the value to Y.	N	1 V

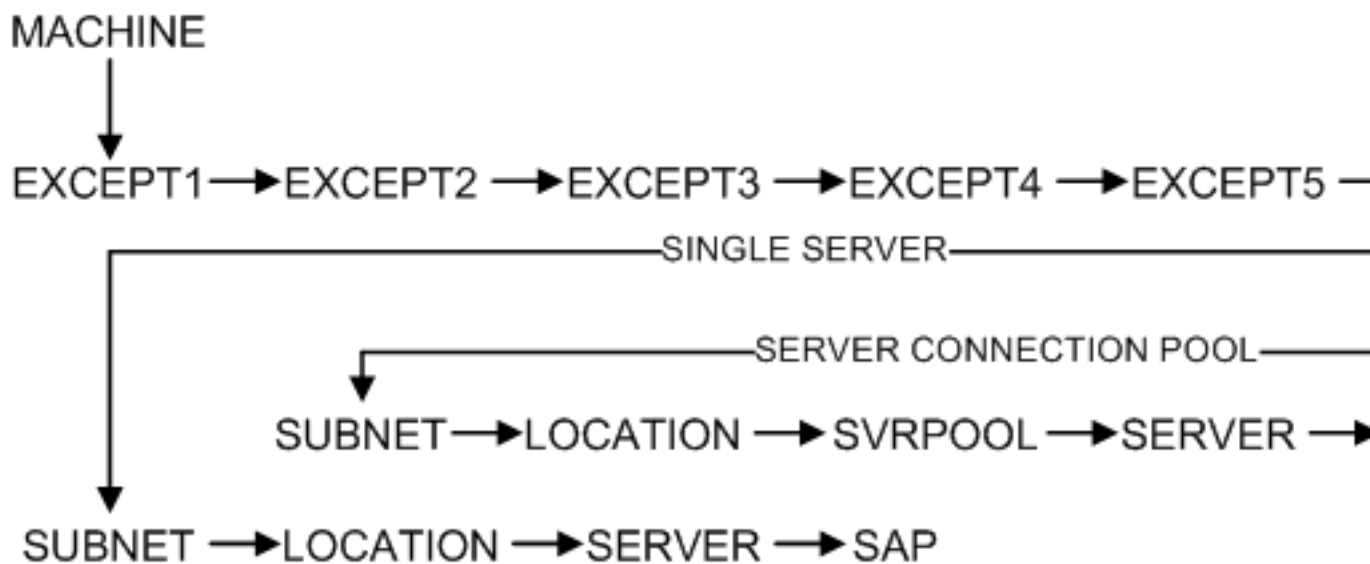
Custom Exception (EXCEPT1-EXCEPT6) Class

File:	PRIMARY
Domain:	CLIENT
Class:	EXCEPT1 EXCEPT2 EXCEPT3 EXCEPT4 EXCEPT5

File:	PRIMARY
	EXCEPT6
Icon:	
Description:	A MACHINE instance connects to an EXCEPT1 instance, which chains up to EXCEPT6 and then connects to a SUBNET instance. The _ALWAYS_ connection is EXCEPT2 through SUBNET depending on which EXCEPT# is being used. The SUBNET instance connects to a LOCATION instance. <i>Custom resolution logic should be added here, before the exception pathway reaches the SUBNET instance. It is not recommended to manually change the SUBNET class or any class after this (LOCATION, SERVER, SVRPOOL, and SAP) in the exception pathway as it may introduce errors.</i>

The following figure shows the connection flow for single tier Satellite server:

Satellite Server Connection Flow



EXCEPT1-6 Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
NAME	Display name The display name of the custom exception.		80 V
DESCRIPT	Description The description of the custom exception.		255 V
EXCEPTC	Exception class name	&(ZOBJCLAS)	32

Attribute Name	Explanation	Value of Base Instance	Length Type
	Do not change this value. This is used for reporting purposes only. The exception CLASS value that was used as a part of the connect process is reported. These values are automatically generated at client resolution time.		V
EXCEPTI	Exception instance name Do not change this value. This is used for reporting purposes only. The exception INSTANCE value that was used as a part of the connect process is reported. These values are automatically generated at client resolution time.	&(ZOBJNAME)	32 V
ALWAYS	Connect To Specify the custom exception instance to connect to depending on the exception number.	CLIENT.EXCEPT2. &(PREFACE.ZEXCEPT2) CLIENT.EXCEPT3. &(PREFACE.ZEXCEPT3) CLIENT.EXCEPT4. &(PREFACE.ZEXCEPT4) CLIENT.EXCEPT5. &(PREFACE.ZEXCEPT5) CLIENT.EXCEPT6. &(PREFACE.ZEXCEPT6) CLIENT.SUBNET. &(ZCONFIG.ZNETLOC)_ &(ZCONFIG.ZSUBNETU)	100 A

Diagnostics (DIAGS) Class

Use this class to override default trace settings on the Sentient agent computer. You can also set parameters for running the RADSTATE program. RADSTATE is a diagnostic module that is designed to give an overview of the current state of the Sentient agent. The information in the RADSTATE output is based on data that has been retrieved from numerous Sentient agent objects.

Note: Instances of this class allow you to easily set tracing levels as well as RADSTATE parameters for a user, a machine, or a group of users. These attributes were intentionally put into their own transient class for this purpose. To do this, set the `_ALWAYS_` Diagnostics Class connection in `LOCATION._BASE_INSTANCE_` to `DIAGS.&(ZCONFIG.ZHWDWCOMP)`. Then, create an instance in the DIAGS Class with the computer name of the Sentient agent computer for which you want to set the tracing. If the machine name does not exist in the DIAGS Class, the `DEFAULT_DIAGS` instance settings will be used.

File:	PRIMARY
Domain:	CLIENT
Class:	DIAGS
Icon:	
Description:	Use this class to override default trace settings on the Agent computer.

DIAGS Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
NAME	Friendly Name Friendly name of the instance.		50 V
RADSTATE	Command String for RADSTATE This will run <code>RADSTATE . EXE</code> with the parameters specified in this attribute. If no parameters are specified, RADSTATE will not run. The base instance of the DIAGS Class is set to VO, which will run RADSTATE in verbose mode, building the ZRSTATE and ZRSTATES objects. If no parameters are specified, RADSTATE will not run. RADSTATE must exist in the IDMSYS directory. You only need to specify the parameters for RADSTATE, not the RADSTATE executable. Refer to the document on RADSTATE for additional information.	MODE=VO	80 V
ZTRACE	Communication Tracing [Y/S/N] Specify N to turns off communication buffer tracing. Tracing is off by default. Specify S to provide summary communication buffer information to the client log. This includes number of records read and written and the type of records processed. Specify Y to provide full communication buffer information to the client log. All data transmitted and received will be echoed to the client log file. Caution: Setting <code>ZTRACE = Y</code> may generate very large logs and severely impact client performance. Do not set this unless instructed to do so by technical support.		1 V

Attribute Name	Explanation	Value of Base Instance	Length Type
ZTRACEL	Trace level (000/040/999) Specify tracing level (as 000, 040, or 999) that will be recorded to the Sentient agent log file. If blank, use existing value. Caution: Setting ZTRACEL to a higher number may generate very large logs and severely impact client performance. Do not set this unless instructed to do so by technical support.	040	3 V

Hardware Scan Config (RADHWCFG) Class

Use instances in the RADHWCFG Class to specify the type of hardware scans you want performed on the agent device. Hardware scan information is reported in the ZCONFIG object. To implement the hardware scan options, connect an instance of the RADHWCFG Class to an instance in the LOCATION Class.

Note: Client Operations Profiles must be enabled in order to use the RADHWCFG Class. For testing, consider creating a RADHWCFG object on the agent device with all the attributes in the RADHWCFG Class, and then change the attributes to Y or N to see the result in the ZCONFIG object.

It provides four sample instances in RADHWCFG.

- **Base Instance**
Create copies of the `_BASE_INSTANCE_` to create your own hardware scans.
- **Default Hardware Scan**
This instance scans for the most commonly requested information.
- **Hardware Configuration (Network Only)**
This instance scans for network information only.
- **Sample Dynamic Scan**
This instance provides samples using the Dynamic Scan variables.

The table below details each of the possible hardware scans. Examples of the ZCONFIG attributes that might be returned are provided.

Note: The attributes that are returned will depend on the hardware configuration. For example, if the agent device has only one printer connected, only one ZHDWPA0n attribute will be reported in ZCONFIG.

File:	PRIMARY
Domain:	CLIENT
Class:	RADHWCFG
Icon:	

RADHWCFG Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
NAME	Friendly Name The friendly name of the instance.		50 V
CPU	CPU [Y/N] Specify Y to scan for CPU information. ZCONFIG attributes: ZHDWBIOS, ZHDWCOMP, ZHDWCPU, ZHDWCPUN, ZHDWCPUS, ZHDWFPU, ZHDWXPAG, ZHWCPU01, ZHDFPU01	Y	1 V
OS	OS [Y/N] Specify Y to scan for operating system information. ZCONFIG attributes: REBOOTD, REBOOTT, WTSSRVR, ZHDWLANG, ZHDWOS, ZHDWOSDB, ZHDWOSOG, ZHDWOSOW, ZHDWSVCP	Y	1 V
MEMORY	Memory [Y/N] Specify Y to scan for memory information. ZCONFIG attributes: ZHDWMEM, ZHDWMEMF	Y	1 V
HDLOCAL	Local Drives [Y/N] Specify Y to scan for internal hard drives. ZCONFIG attributes: ZHDWCDDR, ZHDWD00, ZHDW00C, ZHDWD00F, ZHDWD00S, ZHDW00T, ZHDWD01, ZHDW01C, ZHDWDF_A, ZHDWDLST, ZHDWDNUM	N	1 V
HDREMOTE	Remote Drives [Y/N] Specify Y to scan for external hard drives. ZCONFIG attributes: ZHDW00, ZHDWD00C, ZHDWD00F, ZHDW00S, ZHDW00T, ZHDWDLST, ZHDWDNUM	N	1 V
NETWORK	Network [Y/N] Specify Y to scan for network information.	Y	1 V

Attribute Name	Explanation	Value of Base Instance	Length Type
	ZCONFIG attributes: GATEWY01, IPADDR01, LADAPT01, NETLOC01, SUBNET01, ZGATEWAY, ZHDWIPAD, ZHDWLANA, ZHDWNET1, ZHDWNNET, ZNETLOC, ZSUBNET		
PERIPHER	Peripherals [Y/N] Specify Y to scan for peripherals such as keyboard and mouse. ZCONFIG attributes: ZHDWKYBD, ZHDWMOUS, ZHDWPPAR, ZHDWPSE, ZHDWVIDO, ZHDWVRES	N	1 V
PRINTER	Printers [Y/N] Specify Y to scan for printers. ZCONFIG attributes: ZHDWPA00, ZHDWPA01, ZHDWPPRN	N	1 V
HAL_VER	HAL Statistics [Y/N] Specify Y to scan for the HAL (Hardware Abstraction Layer) version. ZCONFIG attributes: HALCOMP, HALDATE, HALFNAME, HALFVER, HALINAME, HALLANG, HALPNAME, HALPVER, HALSIZE	N	1 V
APP_VER	Application Version [Y/N] Specify Y to scan for versions of MSI (ZHDWVMSI) and IE (ZHDWVIE).	Y	1 V
WMISCAN	Use WMI to collect data [Y/N] Specify Y to perform the scan using WMI (Windows Management Instrumentation).	Y	1 V
DSCAN00n	Dynamic Scan 00n Specify Y to use the dynamic scan variable.		1 V
ZCFGOBJ	Specify the name of an object created to receive the results of any dynamic scans that are defined in the RADHWCFG class. The default is the ZCONFIG object. The new object will be created in the RADSETUP directory and will be sent to the Configuration Server as part of the Sentient agent connect.		
EVNDEFLT	Specify Y to create environment variables in ZCONFIG.EDM file if they do not already exist in your system. The default value is N.	N	1 V

Dynamic Scanning

In addition to the built-in scans, create your own scans using the Dynamic Scan (DSCAN00n) Instances. There are three types of dynamic scan instances **WMI**, **Registry**, and **File**. The format for a dynamic scan is:

VariableName = *Type*(*Parm1*, *Parm2*, ...)

Where

- *VariableName* is the attribute in ZCONFIG where you want the information to be reported.
- *Type* is WMI, Registry or File.
- *Parmn* is the query for the information.

Listed below are some examples.

Note: The Dynamic Scan for ZCONFIG is restricted to the `root\cimv2` namespace only.

Example 1: WMI

A WMI scan would use the following format: *VariableName* = *WMI*(*WQL Statement*, *Property*, *Default*). To collect the Model of an agent device using WMI, create a DSCAN000 variable similar to:

```
HWMODEL=WMI("Select * from Win32_ComputerSystem"; Model; NONE)
```

This scan would create the variable ZCONFIG.HWMODEL, and populate it with the agent device's model.

Example 2: Registry

To scan a registry key to determine where Adobe 5.0 is installed, create a DSCAN001 variable similar to:

```
ADOBEPATH=REG(HKLM\SOFTWARE\Adobe\Acrobat_Reader\5.0\InstallPath)
```

The result will be reported in ZCONFIG.ADOBEPATH.

Note: When scanning for a default registry value, the path to the registry key must end with a backslash. For example, to read the default value of the Installer key type `ADOBEPATH=REG("HKLM\SOFTWARE\Adobe\Acrobat Reader\6.0\Installer\")`
To read the Path value of the Installer key, type `ADOBEPATH=REG("HKLM\SOFTWARE\Adobe\Acrobat Reader\6.0\Installer\Path")`

Example 3: File

Dynamic file scanning can return size (SIZE), date stamp (DATE), file version (FVER), product version (PVER), and time (TIME) stamp of a specified file. You can request any combination of these properties. To scan for the file `C:\temp\test.exe`, create a DSCAN002 similar to:

```
TEST####=FILE(c:\Temp\Test.exe;SIZE,DATE,FVER,PVER,TIME)
```

The ##### will be replaced by the corresponding file property name. One attribute will be created in the ZCONFIG object for each file property for which you scanned. In this example, five variables will be created based on the information collected on the C:\temp\test.exe file, ZCONFIG.TESTSIZE, ZCONFIG.TESTDATE, ZCONFIG.TESTFVER, ZCONFIG.TESTPVER, and ZCONFIG.TESTTIME.

Load Balancers (BALANCER) Class

The Load Balancers (BALANCER) class is reserved for future use.

File:	PRIMARY
Domain:	CLIENT
Class:	BALANCER
Icon:	
Description:	A balancer connects to a SVRPOOL instance only. It has a hostname configured that overrides all SERVER hostname in a SVRPOOL. This class should be used with hardware load balancers. BALANCER configuration is not currently supported in the UI.

BALANCER Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
NAME	Display name The display name of the load balancer.		80 V
DESCRIPT	Description The description of the load balancer.		255 V
ENABLED	Enable balancer [Y/N] Specify Y if the environment uses hardware load balancers.	Y	1 V
URL	Management URL		100 V
HOSTNAME	Fully qualified host name Fully qualified host name of the load balancer, or the IP address on which the load balancer is listening for requests.		100 V
ALWAYS	Connect to SVRPOOL Specify the SVRPOOL instance to which the load balancer connects.		100 A

Machine Exception (MACHINE) Class

File:	PRIMARY
Domain:	CLIENT
Class:	MACHINE
Icon:	
Description:	This is the entry point exception class to allow per machine overrides. By default, it connects back to the SUNBNET instance through the exception pathway for normal production operation. However, this can be customized if needed. RADSETUP has been changed to point to this instance using CLIENT.MACHINE.&(ZCONFIG.ZHDWCOMP). <i>Custom resolution logic should be added here, before the exception pathway reaches the SUBNET instance. It is not recommended to manually change the SUBNET class or any class after this (LOCATION, SERVER, SVRPOOL, and SAP) in the exception pathway as it may introduce errors.</i>

MACHINE Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
NAME	Display name The display name of the machine.		80 V
DESCRIPT	Description The description of the machine.		255 V
EXCEPTC	Exception class name Do not change this value. This is used for reporting purposes only. The exception CLASS value that was used as a part of the connect process is reported. These values are automatically generated at client resolution time.	&(ZOBJCLAS)	32 V
EXCEPTI	Exception instance name Do not change this value. This is used for reporting purposes only. The exception INSTANCE value that was used as a part of the connect process is reported. These values are automatically generated at client resolution time.	&(ZOBJNAME)	32 V
ALWAYS	Connect to Specify the exception instance to connect to.	CLIENT. EXCEPT1.	100 A

Attribute Name	Explanation	Value of Base Instance	Length Type
		&(PREFACE. ZEXCEPT1	

Network Locations (LOCATION) Class

File:	PRIMARY
Domain:	CLIENT
Class:	LOCATION
Icon:	
Description:	Use instances in the LOCATION class to define the agent connection order to use when obtaining resources from the Sentient infrastructure. <i>It is not recommended to make manual changes directly to the CSDB. All changes should be performed through the Sentient Core Console. If manual changes are required, they should be performed under the direction of Persistent Systems Support. Manual changes could impact UI operation.</i>
Can connect to:	DIAGS, SETTINGS, NTFYSEC, RADALERT, SERVER, SVRPOOL, BALANCER

LOCATION Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
NAME	Display name of this location		80 V
DESCRIPT	Description of this location		255 V
ALOCINST	Actual location instance name	&(ZOBJNAME)	32 V
RLOCINST	Resolved location instance name	&(ZOBJNAME)	32 V
REGION	Region		50 V
COUNTRY	Country		50 V

Attribute Name	Explanation	Value of Base Instance	Length Type
STATE	State		50 V
CITY	City		50 V
ADDRESS	Address		50 V
BUILDING	Building		50 V
FLOOR	Floor		50 V
ALWAYS	Core SETTINGS Class Connection Specify an instance in the SETTINGS Class.	SETTINGS .DEFAULT _SETTINGS	100 C
ALWAYS	Diagnostics Class Connection Specify an instance in the DIAGS Class.	DIAGS .DEFAULT _DIAGS	100 C
ALWAYS	UI Class Connection Specify an instance in the RADUICFG Class.		100 C
ALWAYS	Hardware Class Connection Specify an instance in the RADHWCFG Class.		100 C
ALWAYS	Notify Security Class Connection Specify an instance in the NTFYSEC Class.	NTFYSEC.DEFAULT_ NTFYSEC	100 C
ALWAYS	RADALERT Class Connection Specify an instance in the RADALERT Class RADALERT.DEFAULT_ RADALERT		100 C
ALWAYS	Connect Defer Class Connection Specify an instance in the CDFCFG Class.		100 C
ALWAYS	Reboot Defer Class Connection Specify an instance in the RBOOTCFG Class.		100 C
ALWAYS	Connect To Class		100 C

Attribute Name	Explanation	Value of Base Instance	Length Type
	Specify an instance in any class to connect to this Location instance		
ALWAYS	Connect To Class Specify an instance in any class to connect to this Location instance.		100 C
SAPPRI	SAP Priority Specify the priority of the SAP instance referenced in the _ALWAYS_ class connection below this attribute.	10	3 V
ALWAYS	Connect To SVRPOOL SERVER BALANCER Specify SVRPOOL, SERVER or BALANCER instance for the priority entered into SAPPRI above this attribute.		100 A
SAPPRI	SAP Priority Specify the priority of the SAP instance referenced in the _ALWAYS_ class connection below this attribute.	20	3 V
ALWAYS	Connect To SVRPOOL SERVER BALANCER Specify SVRPOOL, SERVER or BALANCER instance for the priority entered into SAPPRI above this attribute.		100 A
SAPPRI	SAP Priority Specify the priority of the SAP instance referenced in the _ALWAYS_ class connection below this attribute.	30	3 V
ALWAYS	Connect To SVRPOOL SERVER BALANCER Specify SVRPOOL, SERVER or BALANCER instance for the priority entered into SAPPRI above this attribute.		100 A
SAPPRI	SAP Priority Specify the priority of the SAP instance referenced in the _ALWAYS_ class connection below this attribute.	40	3 V
ALWAYS	Connect To SVRPOOL SERVER BALANCER Specify SVRPOOL,		100

Attribute Name	Explanation	Value of Base Instance	Length Type
	SERVER or BALANCER instance for the priority entered into SAPPRI above this attribute.		A
SAPPRI	SAP Priority Specify the priority of the SAP instance referenced in the _ALWAYS_ class connection below this attribute.	50	3 V
ALWAYS	Connect To SVRPOOL SERVER BALANCER Specify SVRPOOL, SERVER or BALANCER instance for the priority entered into SAPPRI above this attribute.		100 A
SAPPRI	SAP Priority Specify the priority of the SAP instance referenced in the _ALWAYS_ class connection below this attribute.	60	3 V
ALWAYS	Connect To SVRPOOL SERVER BALANCER Specify SVRPOOL, SERVER or BALANCER instance for the priority entered into SAPPRI above this attribute.		100 A
SAPPRI	SAP Priority Specify the priority of the SAP instance referenced in the _ALWAYS_ class connection below this attribute.	70	3 V
ALWAYS	Connect To SVRPOOL SERVER BALANCE Specify SVRPOOL, SERVER or BALANCER instance for the priority entered into SAPPRI above this attribute.		100 A
SAPPRI	SAP Priority Specify the priority of the SAP instance referenced in the _ALWAYS_ class connection below this attribute.	80	3 V
ALWAYS	Connect To SVRPOOL SERVER BALANCER Specify SVRPOOL, SERVER or BALANCER instance for the priority entered into SAPPRI above this		100 A

Attribute Name	Explanation	Value of Base Instance	Length Type
	attribute.		
SAPPRI	SAP Priority Specify the priority of the SAP instance referenced in the <code>_ALWAYS_</code> class connection below this attribute.	90	3 V
<code>_ALWAYS_</code>	Connect To SVRPOOL SERVER BALANCER Specify SVRPOOL, SERVER or BALANCER instance for the priority entered into SAPPRI above this attribute.		100 A
SAPPRI	SAP Priority Specify the priority of the SAP instance referenced in the <code>_ALWAYS_</code> class connection below this attribute.	100	3 V
<code>_ALWAYS_</code>	Connect To SVRPOOL SERVER BALANCER Specify SVRPOOL, SERVER or BALANCER instance for the priority entered into SAPPRI above this attribute.		100 A

Notify Security (NTFYSEC) Class

Notify is a protocol used to remotely run an Sentient agent connect. The administrator can use Sentient Console to initiate a Notify job from Sentient Configuration Server. The agent computer's Notify Daemon, RADEXECD, receives the notify request and initiates the agent connect. If the user ID and password is defined for the agent computer, the verification of these credentials is done before any action is performed.

The administrator can configure additional security options to control the actions performed using Notify. These options are defined in the Notify Security (NTFYSEC) configuration class as part of Client Operations Profile (COP).

File:	PRIMARY
Domain:	CLIENT
Class:	NTFYSEC
Icon:	
Description:	Use this class to use COPs to set RADEXECD to use internal authentication with the Sentient Portal

NTFYSEC Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
RMTALLOW	Custom Exe list for Remote notify List of the executables with extensions and batch files located at IDMSYS directory which has to be executed during the remote notify when ZNTFYSEC value is x. Executable should be comma separated.		255 V
ZEXTSEC	Extra security [Y/N/L] Enables notify security. Valid values are Y (Yes), N (No), and L (Lock). The default is N. Set Y to restrict execution of the notify request to programs residing in the agent's IDMSYS directory. Set L to further limit execution to the agent's main execution modules RADSKMAN, RADPINIT, RADCONCT, and UPGRDMAINT. Set N, to issue the notify request from wherever it resides.	N	1 V
ZIGNDURI	DATAURI stripping [Y/N/A] This anti-spoofing attribute lets you to enable DATAURI stripping (the notify daemon strips out the value before the command is run). Valid values are Y (Yes), N (No), and A (All). The default is N. The option Y enables you to strip the DATAURI for notifies from remote and local machines. The option A enables you to strip the DATAURI for notifies from remote and local machines, direct radskman launch, and direct radpinit launch.	N	1 V
ZIGNRURI	RCSURI stripping [Y/N/A] This anti-spoofing attribute lets you to enable RCSURI stripping (the notify daemon strips out the value before the command is run). Valid values are Y (Yes), N (No), and A (All). The default is N. The option Y enables you to strip the RCSURI for notifies from remote and local machines. The option A enables you to strip the RCSURI for notifies from remote and local machines, direct radskman launch, and direct radpinit launch.	N	1 V
ZNTFYSEC	Notify Security [Y/N/L/X] Enables notify security. Valid values are Y (Yes), N (No), and L (Lock) and X. The default is Y. Set Y to restrict execution of the notify request to programs residing in the agent's IDMSYS directory. Set L to further limit execution to the agent's main execution modules RADSKMAN, RADPINIT, RADCONCT, and UPGRDMAINT. Set X to allow the custom		1 V

Attribute Name	Explanation	Value of Base Instance	Length Type
	executable listed in RMTALLOW and the agent's main execution modules RADSKMAN, RADPINIT, RADCONCT, and UPGRDMAINT. Set N, to issue the notify request from wherever it resides.		
ZVRFYPWD	Verify Internal Password [Y/N/I] Enables password verification. Valid values are Y (Yes) and N (No). The default is N. Set Y to verify that the password sent by the Notify command matches the ZPWD field found in an agent's ZMASTER object. Note: The ZPWD attribute is encrypted and is not visible in clear text.	N	1 V
ZVRFYUID	Verify Internal User ID [Y/N/I] Enables user ID verification. Valid values are Y (Yes) and N (No). The default is N. Set Y to verify that the user ID sent by the Notify command matches the ZUSERID field found in an agent's ZMASTER object. Set I to verify that the user ID sent by the Notify command matches the Sentient agent's CRC'd device domain name.	N	1 V
REJFETCH	Nvdkit Fetch Security [Y/N] Enables Nvdkit notify security. Valid values are Y (Yes) and N (No). The default is N. Set Y to restrict execution of the notify request to Nvdkit Fetch to download the payload. Set N to issue the notify request to download the payload.	N	1 V

Sentient Extended Security

Extended security runs pre-defined command tokens to additionally limit Notify. The command tokens are defined to represent a full command line that is run on the agent computer. To enable extended security set ZEXTSEC=Y. Once enabled, the Notify daemon requires an additional control object called ZNFYXSEC in the IDMROOT directory before any command is run. If the ZNFYXSEC object is not found in the directory no command is run on the agent computer. The ZNFYXSEC object contains a set of authorized command tokens and corresponding commands for an agent. The object contains one instance for each of the command tokens.

When the Notify daemon receives a Notify job request with extended security enabled, the first parameter in the command line is matched with the value of one of the ZNFYXSEC.NFYNAME variables in each of the ZNFYXSEC instances. If a matching ZNFYNAME is found, the daemon compares the received password with the value of ZNFYPWD attribute of that instance. If the password matches, the daemon initiates the command that is specified by the value of

ZNFYCMDL of that instance. If ZNFOPTOK=Y, then any parameters received from the server are appended to the value of ZNFYCMDL before the command is run.

The following table describes the variables of the ZNFYXSEC object on Sentient agent computer.

Variables of ZNFYXSEC object

Variable	Description
ZNFYNAME	Specifies the key to locate or identify the heap. This will correspond to the user ID parameter on the sender's command line.
ZNFYPWD	This value is validated against the incoming password field.
ZNFYGRP	For future use in potential group-notification design.
ZNFYCMDL	Specifies the command line that is run on the agent computer.
ZNFOPTOK	Specify Y to indicate that a parameter can be appended to the command line found in ZNFYCMDL.

Restricting Notify to Self Notify Only

When Sentient agent connects in a System context, it uses a self notify process. The Notify Daemon accepts an optional command-line argument, /L. This argument restricts the daemon to self notify only and rejects any remote notify requests. It also restricts any requests to the main agent execution modules such as RADSKMAN, RADPINIT, RADCONCT, and UPGRDMAINT.

When the Notify daemon receives a new request and /L parameter is used, the daemon checks the IP address of the source and the command requested. If the source IP address is different from the localhost address, 127.0.0.1, the connection is rejected. Also, the request is rejected if the command is not one of the main agent execution modules. The date and time of the rejected request is logged in the daemon's log files.

Note: For Linux, the optional command line argument is -L. Also, L is case sensitive.

Usage Note

Changes to the notify-security settings will not be immediately effective; Sentient agent connect must be performed in order to enable the new settings. This differs from other COPs settings, which happen as part of a single connect.

Reboot Defer Prefer (RBOOTCFG) Class

An administrator can use the Reboot deferral function to configure options for a reboot operation. It gives you control to defer a reboot. Use the RBOOTCFG class in the Sentient Administrator CSDB Editor to configure the user-facing Reboot Deferral (RDF) dialog box. For more information on Reboot Deferral, see the *Sentient Enterprise Console Online Help*.

File:	PRIMARY
Domain:	CLIENT
Class:	RBOOTCFG
Icon:	
Description:	Use this class to configure the user-facing Reboot Defer (RDF) dialog box.

RBOOTCFG Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
NAME	Friendly name of the instance.		32 V
ENABLE	Enable / Disable RDF [Y/N]	N	1 V
DESCTEXT	Descriptive text for RDF dialog. DESCTEXT replaces the first two sentences of text.		32 V
DESCTEXT2	More descriptive text for RDF dialog. DESCTEXT2 replaces the text "To continue with these actions...".		32 V
TIMEOUT	Dialog timeout in minutes Specify time for the RDF dialog to wait before automatically triggering the reboot.	10	3 V
MAXDAYS	Specify the maximum number of days for which the reboot operation can be deferred. Reboot deferral is not allowed when the MAXDAYS value is zero or empty. Example, if MAXDAYS is set to 3, the user can defer a reboot for a maximum of 3 days; after that, the Defer button becomes inactive and reboot will be forced on the user's machine. Note: - These values will affect the "Defer for" list; intervals that are greater than this setting will not be available in the drop-down list. - On the right side of the window, a message will indicate the remaining number of days that you can defer the reboot.	3	3 V
MAXCANCL	Specify the maximum number of times the reboot operation can be cancelled. Reboot cancel is not	-1	3 V

Attribute Name	Explanation	Value of Base Instance	Length Type
	allowed when the MAXCANCL value is zero. MAXCANCL value -1 signifies that you can cancel the reboot any number of times. - You cannot cancel the reboot if the maximum number of days (MAXDAYS) for reboot defer are exhausted. - On the right side of the window, a message will indicate the remaining number of days that you can cancel the reboot.		

RepObjs Differencer (REPTDIFF) Class

File:	PRIMARY
Domain:	CLIENT
Class:	REPTDIFF
Icon:	
Description:	Use this class to configure the Messaging server to report difference data objects.

REPTDIFF Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
FLUSHDB	Recreate Reporting Reference Info Removes all the reference information of the reporting objects in the SQLite database and rebuilds the database. Set to Y to recreate the SQLite database.	N	1 V
FULLREFR	Perform a Full Refresh Refreshes the SQLite database and sends the data to the Messaging Server. Set to Y to refresh the complete data to CSDB. Note: The value given to the FULLREFR attribute can be overridden by passing the command line parameter 'fullrefr=y' during the	Y	1 V

Attribute Name	Explanation	Value of Base Instance	Length Type
	agent connect.		
REPTDIFF	Enable Differencer Engine (Y/N) Enables the Reporting Differencer. Sentient Agent invokes the RepObjDiff.exe before flushing the objects out of the OUTBOX folder. Set to Y to invoke the RepObjDiff.exe. Note: To enable the Reporting Differencer, set REPTDIFF to Y and FULLREFR to N.	N	1 V
REPTOBS	List of Reporting Objects Contains the data objects for differencing, such as DESTATUS.edm, BUSTATUS.edm, and PASTATUS.edm. Sentient agent invokes the RepObjDiff.exe only if any of these objects are available in the OUTBOX folder and the Reporting Differencer is enabled. There is no default for this attribute.		255 V

RSM UI Preferences (RADUICFG) Class

RADSKMAN Client Operations Profiles Parameters

Use the RADUICFG Class to specify settings for Sentient Application Self-Service Manager interface.

Note: You must be licensed for Sentient Application Self-Service Manager to use this class.

File:	PRIMARY
Domain:	CLIENT
Class:	RADUICFG
Icon:	

RADUICFG Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ASKOFFL	Prompt for offline mode [Y/N/U] Specify U to allow the user to control prompting for offline use of Application Self-service Manager. Specify Y or N to turn the prompt on or off, and not allow the user to control the prompt.	Y	1 V
BNBNDWTH	Display Bandwidth [Y/N] Specify Y to display the Bandwidth button on the Side Bar.	Y	1 V
BNHOME	Display Home Button [Y/N] Specify Y to display the Home button on the Side Bar.	Y	1 V
BNHISTORY	Display History [Y/N] Specify Y to display the History button on the Side Bar.	Y	1 V
BNMYSOFT	Display My Software [Y/N] Specify Y to display the My Software button on the Side Bar.	Y	1 V
BNPREFER	Display Preferences [Y/N] Specify Y to display the Preferences button on the Side Bar.	Y	1 V
BNSTATUS	Display Status [Y/N] Specify Y to display the Status button on the Side Bar.	Y	1 V
BTNCANCL	Enable Cancel Button [Y/N] Specify Y to enable the Cancel button.	Y	1 V
BTNDEL	Enable Delete Button [Y/N] Specify Y to enable the Delete button.	Y	1 V
BTNDWLD	Enable Download Button [Y/N] Specify Y to enable the Download button.	Y	1 V
BTNINST	Enable Install Button [Y/N] Specify Y to enable the Install button.	Y	1 V
BTNPAUSE	Enable Pause Button [Y/N] Specify Y to enable the Pause button.	Y	1 V
BTNRECFG	Enable Reconfigure Button [Y/N]	Y	1

Attribute Name	Explanation	Value of Base Instance	Length Type
	Specify Y to enable the Reconfigure button.		V
BTNREPR	Enable Repair Button [Y/N] Specify Y to enable the Repair button.	Y	1 V
BTNUNDO	Enable Undo Button [Y/N] Specify Y to enable the Undo button.	Y	1 V
BTNUPDT	Enable Update Button [Y/N] Specify Y to enable the Update button.	Y	1 V
BTNVRFY	Enable Verify Button [Y/N] Specify Y to enable the Verify button.	Y	1 V
BWSTRTUP	Show BW Control on Startup [A/Y/N] Set to A to automatically display the bandwidth control when processing a service that has bandwidth settings. Set to Y to always display whether or not the service has bandwidth settings and N never to display.	A	1 V
COLNAMES	Name of Columns (CSV format) Specify the columns you want displayed. Separate the columns with a comma.	Size, Version, Vendor	50 V
COLORBAK	Specify RGB or actual Color Back Specify color for the interface's background. Check the Microsoft Web site for possible colors.		20 V
COLORBTN	Specify RGB or actual Color Button Specify the button colors. Check the Microsoft Web site for possible colors.		20 V
COLORSEL	Specify RGB or actual Color Select Specify color for selection areas. Check the Microsoft Web site for possible colors.		20 V
COLORSET	Colors [SYSTEM/DEFAULT/CUSTOM/USER] Select SYSTEM to use the operating system colors, DEFAULT to use the Sentient default color scheme, CUSTOM to use COLORSEL, COLORBAK, COLORBTN and COLORWK. The user will not be able to change the colors if SYSTEM, DEFAULT, or CUSTOM is selected. Select USER to allow the user to control the colors.	DEFAULT	10 V

Attribute Name	Explanation	Value of Base Instance	Length Type
	- COLORSEL: Specify a color for the selection areas. - COLORBAK: Specify a color for the interface background. - COLORBTN: Specify the color for the buttons. - COLORWK: Specify the color for the work area. You can change the color scheme of COLORSEL, COLORBAK, COLORBTN, and COLORWK by specifying a combination of RGB values or the name of the color. For example, to change the color of the selection area to color red, set the attribute value of COLORSEL to RED or set the RGB values as R=255, G=0, and B=0. Note: Check the Microsoft web site for available colors.		
COLORWK	Specify RGB or actual Color WorkAr Specify the color for the work area. Check the Microsoft Web site for possible color.		20 V
CUSTIMG	Custom Image File or RSM Banner Specify a custom image file or banner. Acceptable file types are JPG/JPEG, GIF, TIF, and BMP. The size limitations in pixels are approximately height of 60 and a width of 250. If no location is specified for the file, the default is IDMLIB (<InstallDir>\ConfigurationServer\Lib).		20 V
CUSTURL	Custom URL on RSM Banner Specify a URL that the agent computer's default Internet browser will open to if the subscriber clicks on the CUSTIMG.		40 V
CUSTTEXT	Custom Hover Text on RSM Banner Specify the text to display when the agent computer's mouse hovers over the CUSTOMIMAGE.	Visit Tech Support.	20 V
CUSTTITLE	Custom Title on RSM Banner		20

Attribute Name	Explanation	Value of Base Instance	Length Type
	Specify the text to display in the Application Self-service Manager title bar.		V
COLTYPE	Columns [Forced/Required] Set to Forced if you want only the columns specified in COLNAMES to appear. Set to Required if at least the columns specified in COLNAMES should appear. Name and Status are always displayed.	REQUIRED	10 V
EXPCITEM	Expand Active Catalog Item [Y/N/U] Specify U to allow the user to control the expansion of the active catalog item. Specify Y or N to allow or not allow the user to expand the active catalog item.	N	1 V
EXPSITEM	Expand Active Service item [Y/N/U] Specify U to allow the user to control the expansion of the active Service List item. Specify Y or N to allow or not allow the user to expand the active item in the Service List.	Y	1 V
NAME	Friendly Name Friendly name of the instance.		20 V
PNLOUTBR	Display Outbar [Y/N] Specify Y to display the Side Bar. This is located on the left side of the panel and provides navigation throughout the interface.	Y	1 V
PROXYADD	Proxy Server Address Specify the Internet proxy server's address.		20 V
PROXYDSC	Discover proxy address [Y/N/U] Specify U to allow the user to control Internet proxy discovery. Specify Y or N to turn proxy discovery on or off, and not allow the user to control its appearance.	U	1 V
PROXYPRT	Proxy Server Port Specify the Internet proxy server's port.		4 V
PROXYUSE	Use Proxy Server [Y/N/U] Specify U to allow the user to control the use of an Internet proxy. Specify Y or N to turn the use of an Internet proxy on or off, and not allow the user to control the use of an Internet proxy.	U	1 V

Attribute Name	Explanation	Value of Base Instance	Length Type
SHWADVOP	Show Advanced Options [Y/N/U] Specify U to allow the user to control the display of Advanced Options. Specify Y or N to turn off or on the display of Advanced Options such as Download Only, Reconfigure, and Undo buttons.	Y	1 V
SHWCATLG	Show Catalog [Y/N/U] Specify U to allow the user to control appearance of the list of catalogs. Specify Y or N to turn the catalog list on or off, and not allow the user to control its appearance.	U	1 V
SHWCOLEX	Show Expand/Collapse Button [Y/N] Specify Y to show the Install button.	Y	1 V
SHWGRID	Show Grid Lines [Y/N/U] Specify U to allow the user to control the display of grid lines. Specify Y or N to turn off or on the display of grid lines.	N	1 V
SHWINFO	Show/Hide Extended Info [Y/N] Specify Y to show the Extended Info button when a service item is expanded.	Y	1 V
SHWMENUS	Show Menus [Y/N/U] Specify U to allow the user to control appearance of the menu bar. Specify Y or N to turn the Menu bar on or off, and not allow the user to control its appearance.	U	1 V
SHWSCHEV	Show/Hide Scheduled Event [Y/N] Specify Y to show the Scheduled Event button when a service item is expanded. This button looks like a clock.	Y	1 V
STATSTR	Status window on startup [Y/N] Set to Y to show the status window on start up.	N	1 V
STRTCHNG	Startup param modifications [Y/N] Specify Y to allow the user to modify the startup parameters in General Options in Preferences. Specify N to not allow the user to modify the startup parameters.	N	1 V
STRTFILE	Startup parameter filename Specify the file name for the startup parameters found in the General Options in Preferences.		50 V

Attribute Name	Explanation	Value of Base Instance	Length Type
STRUPMSG	Warn if Startup file changes [Y/N] Specify Y to warn the user if the startup parameter file has changed.	Y	1 V
TMNUCMDn	RadTray Menu Command <i>n</i> (<i>n</i> is any value ranging from 0 to 9) Create a custom menu item for the System Tray. Specify a command to run when you click TMNUTXT<i>n</i> in the System Tray. The command must be available from the IDMSYS directory.		80 V
TMNUTXTn	RadTray Menu Text <i>n</i> (<i>n</i> is any value ranging from 0 to 9) Create a custom menu item for the System Tray. This menu will be available when you right click on the System Tray icon. To create a separator bar, type SEPARATOR as the menu text.		20 V

Server Access Profile (SAP) Class

File:	PRIMARY
Domain:	CLIENT
Class:	SAP
Icon:	
Description:	A SAP instance is used by the SERVER class to identify resources provided by Sentient Core and Satellite servers. <i>It is not recommended to make manual changes directly to the CSDB. All changes should be performed through the Sentient Core Console. If manual changes are required, they should be performed under the direction of Persistent Systems Support. Manual changes could impact UI operation.</i> <i>Any manual changes to the SAP class will be lost during the next automatic Satellite management update process. Changes should be made to the Satellite server directly, which will then update its respective SERVER and SAP instances.</i>

SAP Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ZSTOP000	Expression Resolution Method Use a ZSTOP expression to stop the process from completing if certain requirements are met, e.g., you may want to prevent a laptop computer from using this SAP.		100 E
ZSTOP001	Expression Resolution Method Use a ZSTOP expression to stop the process from completing if certain requirements are met, e.g., you may want to prevent a laptop computer from using this SAP.	(UPPER(EDMGETV (SERVER, ENABLED)))<>'N')	255 E
NAME	Friendly Name Friendly name of the SAP instance.		80 V
TYPE	Type [RCS /DATA/ROM/PMG] This setting is automatically managed by the respective Core and Satellite Servers.		4 V
BALANCER	Balancer Instance Name This value is used for reporting purposes and should not be changed.	& (BALANCER.ZOBJNAME)	32 V
SVRPOOL	Server Pool Instance Name This value is used for reporting purposes and should not be changed.	&(SVRPOOL.ZOBJNAME)	32 V
SERVER	Server Instance Name This value is used for reporting purposes and should not be changed.	&(SERVER.ZOBJNAME)	32 V
ALOCINST	Actual Location Instance Name This value is used for reporting purposes and should not be changed.	&(LOCATION.ALOCINST)	32 V
RLOCINST	Resolved Location Instance Name This value is used for reporting purposes and should not be changed.	&(LOCATION.RLOCINST)	32 V

Attribute Name	Explanation	Value of Base Instance	Length Type
PROTOCOL	Protocol [http/https/tcp/tcps] Specify a transport protocol.		5 V
HOSTNAME	Fully Qualified Host Name This value is automatically set from the parent SERVER instance and should not be changed.	&(SERVER.HOSTNAME)	100 V
PORT	Port This setting is automatically managed by the respective Core and Satellite Servers.		5 V
URI	Universal Resource Identifier This value is dynamically built based on the SERVER and SAP information available and should not be changed.	&(PROTOCOL)://& (HOSTNAME):&(PORT)	255 V
ROLE	RCS Role A,B,O,S,M,R,D,Z Specifies the role of the SAP. Specify as many values as are needed separated by a comma. A blank or null value defaults to ALL. Possible values are A = all, B = Multicast Enablement, O = Client Operations Profiles, S = Service Resolution, M = Agent Self Maintenance, R = Reporting, D= Data Download. Note: Only a Configuration Server can be designated as anything other than type D. For servers where TYPE=DATA, if you specify anything other than D, that SAP instance will be skipped.	A	12 V
ENABLED	Enable/Disable [Y/N] This setting is automatically managed by the respective Core and Satellite Servers	Y	1 V
TIMEOUT	Communications Time-out (0-3200)s Specify the timeout in seconds. This will override client time-out		4 V

Attribute Name	Explanation	Value of Base Instance	Length Type
	(ZMASTER.ZTIMEO) if it contains a valid numeric value. If blank, then use existing variable value on client.		
PUSHBACK	Push Back (0-999 retries) Set to 0 to skip a Configuration Server if the Configuration Server pushes back on the agent connect. Set to 1 to 999 for number of retries if the Configuration Server pushes back.	0	3 V
RTBWMGMT	Real-time Bandwidth Management Use this attribute to enable the Agent machine to adapt to real time available bandwidth. The bandwidth is calculated based on the Frequency and Percentage parameters. The bandwidth can be considered real-time, if you specify a short duration in frequency. By default, this attribute is disabled. This attribute has the following parameters: • Enable: Describes whether real-time bandwidth management is enabled or not. Default value is N. Specify Y to enable the real-time bandwidth management. • Frequency: Number of seconds for one cycle to calculate the allocated bandwidth from the available bandwidth and percentage value. Default value is 15. • Percentage: Specify the percentage of bandwidth to use from the calculated available bandwidth. Default value is 40. For example, you can specify the values for this parameter as Enable=Y, Percentage=40, Frequency=15.		

Attribute Name	Explanation	Value of Base Instance	Length Type
THROTYPE	[NONE/ADAPTIVE/RESERVED] Type of bandwidth throttling to use. Set to ADAPTIVE to yield to other services that are using the network. Set to RESERVED to allow for a specific reservation of the bandwidth. It is the maximum percentage of network bandwidth to use. Set to NONE for no bandwidth throttling, and use the maximum available bandwidth. This will override client bandwidth throttling if it contains a valid value. If blank, then use existing variable value on the agent computer.		8 V
BANDWIDTH	Bandwidth Percentage (1-99) Specify the percentage of bandwidth to use between 1 and 99. If blank value or non-existent variable, then use all of the bandwidth. This will override agent bandwidth setting if it contains a valid value. If blank, then use existing variable value on agent computer.		2 V
STREAM	Enable Streaming [Y/N] Specify Y to use streaming. This will override the client setting in ZMASTER.ZNORSPNS. Caution: Streaming is not suitable for all network environments. Consult your network administrator before setting this to Y.	N	1 V
PROXY	Internet Proxy URI Do not modify. The Internet proxy URI through which the agent will connect to the SAP. Maintained by agent		255 V
PRIORITY	Selection Priority Do not modify. The SAP obtains its priority by looking at the priority		20 V

Attribute Name	Explanation	Value of Base Instance	Length Type
	specified in the LOCATION Class. &(LOCATION .SAPPRI)		
PRODUCT	Product Filter Specify which types of agents can use this SAP instance. Specify multiple clients separated by a comma. Below are suggested identifiers for each agent: Application Manager: AM Inventory Manager: IM Application Self-service Mgr: ASM OS Manager: OSM Patch Manager: PATCH On your radskman command line, specify which products to filter by using the product parameter. For example, if this SAP should only be used by Application Manager, then you may want to set this attribute to AM. Then, set product to AM on your radskman command line.		32 V
FILTER	Filter Expression [Obj.Var = Value] Use this attribute to filter the SAP based on any available object attribute. For example, if you only wanted to use this SAP for a specific service, specify APPINFO.ZOBJNAME=GS-CALC. Note: The ZSERVICE object is not available during installation. Use the APPINFO object instead. Appinfo, located in the service's LIB directory, is a copy of the service's instance from the ASERVICE object.		100 V
NETTTL	Network Time to Live (0-999) Controls the maximum number of network segment hops allowed		3 V

Attribute Name	Explanation	Value of Base Instance	Length Type
	when calculating the speed of network connections. If NETTTL is not available or blank, the default value of 3 is used.		
IPVER	IP Version [46/64/4/6] The IP version through which the communication happens. The value 46 signifies that the agent connects to the Satellite server specified in the priority list first through IPv4. If IPv4 is not found, the communication happens through IPv6. The value 64 signifies that the agent first connects to the Satellite server through IPv6. If IPv6 is not found, the agent connects through IPv4. The value 4 signifies that the agent connects to the Satellite server through IPv4. The value 6 signifies that the agent connects to the Satellite server through IPv6.	46	3 V

Server Connect Pool (SVRPOOL) Class

File:	PRIMARY
Domain:	CLIENT
Class:	SVRPOOL
Icon:	
Description:	The group of SERVER instances supporting up to 30 servers per server pool. <i>It is not recommended to make manual changes directly to the CSDB. All changes should be performed through the Sentient Core Console. If manual changes are required, they should be performed under the direction of Persistent Systems Support. Manual changes could impact UI operation.</i>

SVRPOOL Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
NAME	Display name The display name of the server pool.		80 V
DESCRIPT	Description The description of the server pool.		255 V
ENABLED	Enable/Disable [Y/N] Use this attribute to enable or disable device connections to the server pool. You can disable the server pool during the maintenance periods.	Y	1 V
ALWAYS	Connect to SERVER Specify the SERVER instance to connect to.		100 A

Subnet (SUBNET) Class

File:	PRIMARY
Domain:	CLIENT
Class:	SUBNET
Icon:	
Description:	A MACHINE instance connects to an EXCEPT1 instance, which chains up to EXCEPT6 and then connects to a SUBNET instance. The _ALWAYS_ connection for the SUBNET instance is a LOCATION instance. <i>Custom resolution logic should be added in the custom exception chain, before the exception pathway reaches the SUBNET instance. It is not recommended to manually change the SUBNET class or any class after this (LOCATION, SERVER, SVRPOOL, and SAP) in the exception pathway as it may introduce errors.</i>

SUBNET Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
NAME	Display name The display name of the subnet.		80 V
DESCRIPT	Description		255

Attribute Name	Explanation	Value of Base Instance	Length Type
	The description of the subnet.		V
EXCEPTC	Exception class name Do not change this value. This is used for reporting purposes only. The exception CLASS value that was used as a part of the connect process is reported. These values are automatically generated at client resolution time.	&(ZOBJCLAS)	32 V
EXCEPTI	Exception instance name Do not change this value. This is used for reporting purposes only. The exception INSTANCE value that was used as a part of the connect process is reported. These values are automatically generated at client resolution time.	&(ZOBJNAME)	32 V
ALWAYS	Connect to This instance always connects to the LOCATION instance.	CLIENT.LOCATION ._NULL_ INSTANCE_	100 A

Subnet For IPV6 (SUBNET6) Class

File:	PRIMARY
Domain:	CLIENT
Class:	SUBNET6
Icon:	
Description:	

SUBNET6 Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
NAME	Display name The display name of the subnet.		
IPADDR	IP Address with prefix		
LOCATION	Location		

The MOBILE Domain

The MOBILE domain contains information related to the mobile applications that are managed using Sentient. This domain also provides details on the methods that are used to deploy the mobile applications.

Android Black List (ABLCKLIS) Class

File: PRIMARY	
Domain:	MOBILE
Class:	ABLCKLIS
Icon:	
Description	Use an instance of this class to blacklist apps for Android devices.
Can connect to	ZSERVICE

ABLCKLIS Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
Name	Name of the application to blacklist.	NULL	255 V
DESCRIPT	Description of the mobile policy	NULL	255 V
PCKGID	Package ID of the application to blacklist	NULL	255 V
APPTYPE	Application Type (System/Other)		255 V
ZSTOP001	Stop Expression 01		255 E
ZSTOP002	Stop Expression 02	WORDPOS('AND', SUBSTR (EDMGETV(ZMASTER,ZOS),1,3))=0	255 E

Android Policies (ANDRDPOL) Class

File:	PRIMARY
Domain:	MOBILE
Class:	ANDRDPOL
Icon:	
Description	Use an instance of this class to create security profile for Android devices.
Can connect to	ZSERVICE

ANDRDPOL Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
Name	Friendly name of the profile.	-	255 V
CMPLXSYM	Number of complex password symbols		2 V
CMPLXUPP	Number of complex password upper		2 V
CMPLXLOW	Number of complex password lower		2 V
CMPLXNUM	Number of complex password number		2 V
DEVENCR	Set this attribute to enable device encryption		1 V
DESCRIPT	Description of the security profile.	-	255 V
ENBLCAM	Set this attribute to enable camera		1 V
ENBLWIFI	Set this attribute to enable wifi		1 V
ENBLBLU	Set this attribute to enable bluetooth		1 V

ENBLGPS	Set this attribute to enable GPS		1 V
ENBLTETR	Set this attribute to enable Wifi Tethering		1 V
ENBLMDAT	Set this attribute to enable mobile data		1 V
LOCKNOW	Set this attribute to lock the device at that instance.	-	1 V
MAXHIST	Maximum History		2 V
MAXAGE	Maximum Age		2 V
MAXPWDWP	The maximum number of failed password attempts, after which Sentient wipes the mobile device data.	-	255 V
MAXTMLCK	The maximum time a device can remain in idle state before the device is locked.	-	255 V
PWDEXPT	Password Expiration Timeout.	-	255 V
PMINLENT	Minimum length of the password.	-	255 V
PMINLETR	Minimum letter characters in the password.	-	255 V
PMINLOWC	Minimum lower-case letters in the password.	-	255 V
PMINNOLE	Minimum non-letter characters in the password.	-	255 V
PMINNUM	Minimum non-numeric characters in the password.	-	255 V
PMINSYMB	Minimum symbols in the password.	-	255

			V
PMINUPCS	Minimum upper-case letter characters in the password.	-	255 V
PQUALITY	The type of characters that the password must contain. This determines the strength and quality of the password.	-	255 V
PWDRQD	Set a value for this attribute if the managed mobile device user must have a password.	-	1 V
PWDRESET	Set a value for this attribute if the password must be reset.	-	1 V
RESRQD	Set a value for this attribute if the password restriction is to set.		1 V
WIPE	Set this attribute to wipe the mobile device data.	-	1 V
ZSTOP001	Governs the profile that is to be provisioned. Identifies that the profile is applicable only for Android devices.	WORDPOS('AND', SUBSTR(EDMGETV (ZMASTER,ZOS),1,3))=0	255 E
ZSTOP002	Stop Resolution Expression Expressions evaluating to "true" in ZSTOP attributes cause resolution of the instance to be skipped. If left blank, the instance is not skipped, and resolution continues. This is useful for assigning a version to a specific set of users. Use the CSDB Editor to set this attribute.		255 E

Application (ZSERVICE) Class

File:	PRIMARY
Domain:	MOBILE
Class:	ZSERVICE
Icon:	
Description	A ZSERVICE instance represents a unit of managed content that can be

File:	PRIMARY
	connected to many other class instances.
Can connect to	WORKGRP, STAGER

ZSERVICE Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ALWAYS (Four Attributes)	Contains Class connections determine the path of resolution of an agent's desired state. An _ALWAYS_ class connection is always resolved.	-	50 C
ALWAYS (Three Attributes)	Contains An _ALWAYS_ class connection is always resolved. An Includes connection is used to express inter-package relationships.	-	50 I
ALWAYS	Utility Resolution Method Any method you specify for this attribute is unconditionally executed by Client Automation when this instance is resolved. A valid method name, for example, <code>ZSYSTEM.ZMETHOD.PUT_ZERROR_HIST</code> .	-	51 M
ADAPTIVE	Auto Adaptability [Y/N]	-	1 V
APPCAT	Application Category.		30 V
APPTYPE	Application Type.	-	12 V
APSTYPE	Application Streaming.		12 V
AUTHOR	Author Name.	-	50 V
BCREATE	Service Pre-Installation Method.		100 M
BDELETE	Service Pre-Delete Method.	-	100

			M
BNDLEID	Required for iOS applications. The bundle ID is defined for each application, for example, <code>com.hpca</code> .	-	255 V
BREPAIR	Service Pre-Repair Method.	-	100 M
BUPDATE	Service Pre-Update Method.	-	100 M
CACHE	App Element Caching [Y/N].	N	1 V
CACHELIM	Percent Disk Limit For Cache	000	3 V
CACHELOC	CACHE Location on Client	_UNDEF_	254 V
CATGROUP	Catalog Group Name.	-	30 V
COMPSIZE	Application Size- Compressed.	-	12 V
DELDATE	Delete Date.	-	32 V
DESCRIPT	Application Description Description of the service that appears in the properties for the service in the Service List.	-	100 V
ERTYPE	Event Reporting Method [O/E/X].	O	1 V
EVENTS	Events to Report.	AI=B, AD=B, AU=B, AR=B, AV=F, VA=B, VD=B	50 V
INSTDATE	Installed Date.	-	32 V
ITEMS	Number of Component Instances.	-	7 V

LREPAIR	Local Repair [Y/N].	-	1 V
MCELIBL	System Multicast Eligible [Y/N]	Y	1 V
NAME	Friendly Name Friendly name for the service. This appears in the properties for the service in the Service List.	-	80 V
OWNER	Application Contact.	-	50 V
PRICE	Price.	-	10 V
PUBDATE	Published Date of Service.	-	32 V
REBOOT	Reboot Options.	-	64 V
RECONFIG	Reconfiguration Enabled [Y/N].	-	1 V
REMOVAL	Un-Managed Behavior [A/D/U].	D	1 V
RSTRSIZE	Download restart threshold (bytes).	0	11 V
RUNDLG	Dialog Processing [Y/N].	N	1 V
RUNLEVEL	Runlevel Support for Unix.	-	1 V
SCHEDOK	Update Schedule Locally [Y/N].	-	1 V
SIZE	Application Size- Uncompressed.	-	12 V
THROTYPE	[ADAPTIVE/RESERVERD/NONE]	-	8 V

UIOPTION	Progress Indicator [NONE/FULL/INFO].	-	4 V
UPDDATE	Upgrade Date (Programmatic).	-	32 V
UPGDATE	When Application was Upgraded on De.	-	32 V
URL	WEB URL Name.	-	100 V
VENDOR	Vendor Name.	-	24 V
VERDATE	Verified Date of Service.	-	32 V
VERSION	Version Description Version of the software. This appears in the properties for the service in the Service List.	-	50 V
ZAVIS	Available, Verified, Installed, Sync F.	-	4 V
ZBITARCH	Target Bit Architecture [/32/64].	-	8 V
ZBWAPCT	Adaptive Percentage.	-	3 V
ZBWPCT	Reserved Percentage.	-	3 V
ZCONTEXT	Component Context [M/U]	-	1 V
ZCREATE	Service Installation Method.	-	100 M
ZDELETE	Service Delete Method.	-	100 M
ZDISCONN	Disconnect on Install [Y/N]	Y	1 V

ZINIT	Service Initialization Method.	-	100 M
ZREPAIR	Service Repair Method.	-	100 M
ZSTOP000 to ZSTOP002	Stop Resolution Expression Expressions evaluating to "true" in ZSTOP attributes cause resolution of the instance to be skipped. If left blank, the instance is not skipped, and resolution continues. This is useful for assigning a version to a specific set of users. Use the CSDB Editor to set this attribute.		100 E
ZSTOP999	Stop Unless RCA Connect	"\(\UPPER (EDMGETV (ZMASTER, ZDOMNAME)) ='SOFTWARE'	50 E
ZSVCCAT	Service Visible in Catalog? [Y/N]	-	1 V
ZSVCCSTA	Service Status on Client (999).	999	3 V
ZSVCGRP	Service Group [Y/N/D]		1 V
ZSVCMO	Mandatory or Optional [M/O/MO/OM].	0	2 V
ZSVCMODE	Application Context [M/U/MU/EMU]	-	3 V
ZSVCNAME	Service Name	Unknown Service	80 V
ZSVCPRI	Service Create Ordering [01-99].	-	2 V
ZSVCTTYP	Application Target Type [A/S]	-	2 V
ZSYSACCT	Install under System Account [Y/N/M]	Y	1 V

ZUPDATE	Service Update Method.	-	100 M
ZVERIFY	Service Verify Method.	-	100 M

Application Packages (PACKAGE) Class

File:	PRIMARY
Domain:	MOBILE
Class:	PACKAGE
Icon:	
Description	An instance of the MOBILE Domain Application Packages (PACKAGE) class represents one software application or unit of other content to be managed on mobile devices.
Can connect to	ZSERVICE

PACKAGE Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ALWAYS	Connect to Class connections determine the path of resolution of an agent's desired state. An _ALWAYS_ class connection is always resolved.	-	50 C
ALWAYS	Package Includes Class connections determine the path of resolution of an agent's desired state. An _ALWAYS_ class connection is always resolved.	-	50 C
BEHAVIOR	Behavior.	-	255 I
COMPSIZE	Package Compressed Resource Size.	-	12 V
DESCRIPT	Application Description Description of the service that appears in the properties for the service in the Service List.	-	255 V
DESKTOP	Desktop.	-	255

			I
FILE	Includes connection to FILE Class instances.	-	255 I
INCLUDES	Package Includes.	-	50 I
ITEMS	Name of Component Instance.	-	7 V
NAME	Friendly name of the package.	-	80 V
OWNER	Application contact.	-	80 V
PATH	Path.	-	255 I
PKGNAME	Package Name.	-	80 V
REGISTRY	Registry.	-	255 I
RELEASE	Package Release.	-	80 V
REQUIRES	Required Package Level.	-	80 R
SIZE	Package Resource Size.	-	12 V
UNIXFILE	Unix File.	-	255 I
ZBITARCH	Target Bit Architecture [/32/64].	-	8 V
ZSTOP000 to ZSTOP003	Stop Resolution Expression Expressions evaluating to "true" in ZSTOP attributes cause resolution of the instance to be skipped. If left blank, the instance is not skipped, and resolution continues. This is useful for assigning a version to a specific set of users. Use the CSDB Editor to set this attribute.	-	255 E

Async Class

File:	PRIMARY
Domain:	MOBILE
Class:	ASYNC
Icon:	
Description	Use an instance of this class to create active sync configuration profile for mobile devices.

Async Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
NAME	Display name		255 V
DESCRIPT	Description		255 V
ADDRESS	Email Address		255 V
HOST	Specifies the Exchange Server Host		255 V
PDSYNC	number of past days		255 V
SSL	Optional. Default YES		1 V
DOMAIN	Domain name		255 V
USERNAME	Specifies Exchange Account		255 V
PASSWORD	The password of the account		255 V
MOVE	Prevent Move		1 V
APPSHEET	Prevent AppSheet		1 V

SMIME	S/MIME Enabled optional		1 V
MAILSYNC	Mail Recents-Syncing		1 V

File Resources (FILE) Class

File:	PRIMARY
Domain:	MOBILE
Class:	FILE
Icon:	
Description	An instance of this class lets you define and specify characteristics of an application such as the file name and file size.
Can connect to	ZSERVICE

FILE Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ALWAYS	Connect to Class connections determine the path of resolution of an agent's desired state. An _ALWAYS_ class connection is always resolved.		50 C
FVERSION	File Version	_UNDEF_	30 V
LOCATION	Location.	-	255
NAME	Friendly Name		80 V
OBJGUID	Instance GUID Object Global Unique Identification – <i>for future use</i> .	_UNDEF_	32 V
PVERSION	Promote Version.	_UNDEF_	30 V
SIGTYPE	File Signature Type.	-	3 V
ZCMPKEY	Compression Key	-	80

			V
ZCMPSIZE	Compressed File Size	-	11 V
ZCONTEXT	Component Context.	-	8 V
ZCREATE	Method to Install Resource	-	50 M
ZDELETE	Method to De-install Resource	RADREMF	50 M
ZFILETYP	File Type.		8 M
ZFILEUPD	Client File Update/Add Method	-	50 M
ZINIT	Resource Initialization Method	-	50 M
ZOBJPATCH	Perform Patching [Y/N]	Y	1 V
ZOPENERR	Client Method on File Open Error	RADLKM	50 M
ZPUBVER	Publisher Version	_UNDEF_	8 V
ZREPAIR	Client Instance Repair Method.	-	50 M
ZRSCCFIL	Resource File Name Specific file to be distributed. A generic pattern (a wildcard character, such as *) or an explicit filename.		255 V
ZRSCCKPT	Resource Checkpoint	-	1 V
ZRSCCONF	Confirm File Download [Y/N]	Y	1 V
ZRSCCRC	Resource CRC	-	8 V
ZRSCCSTA	Client File Status	999	3 V
ZRSCDATE	Resource Date Stamp - From Promote Date stamp on resource file from promote process.		8 V

ZRSCMEM	PDS Member Name	-	8 V
ZRSCMO	Mandatory/Optional on Client [M/O].	-	1 V
ZRSCNAME	Resource Name	&ZRSCCFIL	30 V
ZRSCOBJN	Persistent Object Name	-	8 V
ZRSCPADM	Admin ID	-	30 V
ZRSCPRI	Create Sequence Priority	50	2 V
ZRSCRASH	DOS File Attribute [R/A/S/H]	-	4 V
ZRSCRSTR	Restart [Y/N]	N	3 V
ZRSCSIG	File Signature	-	32 V
ZRSCSIZE	Resource Size- From Promote Byte count of resource file from promote process.		11 V
ZRSCSRC	Resource Source, i.e., Publisher Stores the source of the resource: If resource was retrieved from Configuration Server, the value would be M. If resource existed on the desktop, the value would be E. If resource was retrieved from Stager, value would be S. These values are filled in by the Sentient user during an application installation. When the user uninstalls an application, only the files that were retrieved from the Configuration Server or the Staging Server would be deleted. The files that existed on the desktop prior to an application install would not be deleted.		20 V
ZRSCSTYP	Server File Type [BINARY/TEXT]	BINARY	8 V
ZRSCSVRB	Version Skip Rebuild Flag [Y/N]	-	1 V

ZRSCTIME	Resource Time Stamp - From Promote Time stamp on resource file from promote process.		8 V
ZRSCVERS	Resource Version - From Promote Version number of resource from promotion process.		12 V
ZRSCVRFY	Verify Resource File on Connect	Y	1 V
ZUPDATE	Client Instance Update Method		50 M

iOS Profile (IOSPRFIL) Class

File:	PRIMARY
Domain:	MOBILE
Class:	IOSPRFIL
Icon:	
Description	Use an instance of this class to create security profile for iOS devices.
Can connect to	ZSERVICE

IOSPRFIL Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
RESRQD	Restriction required. Set if restrictions are required on the device.	0	1 V
NAME	Friendly name of the profile.	-	32 V
DESCRIPT	Description of the iOS profile	-	255 V
PWDRQD	Passcode required. Set a value for this attribute if password is required on the mobile device.	-	1 V
PWDSIM	Allow Simple Passcode. Set a value for this attribute if the password must not contain any special character.	-	1 V

PWDALNUM	Require Alphanumeric Passcode. Set a value for this attribute if the password must contain alphanumeric characters.	-	1 V
PMINLENT	Minimum Passcode Length. The value for the minimum length of the password.	-	255 V
PWDMINCX	Minimum Number of Complex character. Minimum number of non-alphanumeric characters that the password must contain.	-	255 V
PWDMXMAG	Maximum Passcode Age. The time after which the user must change the password (1-730 days).	-	3 V
MAXTMLCK	Auto-Lck for Mobile Devices. The time (in minutes) the device can remain in idle state, after which the device is locked automatically.	-	6 V
PWDHIST	Passcode History. The number of passwords that the device remembers. A user cannot set a password that matches the passwords available in the history list.	-	5 V
GRCPRD	The grace period for device lock.	-	3 V
MAXPWDWP	Maximum Failed Attempts. The maximum number of times a user can enter a wrong password, after which the device is wiped.	-	2 V
SECURITY	Security Control for Profile Remove. Set if a user can delete a security profile.	-	20 V
SCRTYCOD	Security Code for removing the Prof.	-	255 V
WIPE	Wipe Mobile Device. Set to wipe the mobile device data.	-	1 V
LOCKNOW	Lock Device. Set to lock the device at that instance.	-	1 V
PWDCLR	Clear the Passcode. Set to clear the password set for the device.	-	1 V

Reference GuideChapter 10: Configuration Server

ENBLITUN	Enable iTunes. Set to enable user to use iTunes store on the device.	1	1 V
ENBLAPPS	Enable Appstore. Set to enable user to allow application installation on the device.	1	1 V
INAPPPUR	Enable In-App Purchases [0/1]	1	1 V
ENBLINAP	Enable InApp Purchase [0/1]	1	1 V
ENBLVCDL	Enable Voice Dialing [0/1]	1	1 V
ASSIT	Enable Assistant [0/1]	1	1 V
ASSISTLCK	Enable Assistant when locked [0/1]	1	1 V
ENBLBOOK	Enable Bookstore [0/1]	1	1 V
BKSEROT	Allow Bookstore Erotica [0/1]	1	1 V
ENBLCAM	Enable Camera [0/1]	1	1 V
ENBLSHST	Enable Share Stream [0/1]	1	1 V
UNTRST	Enable Untrust Prompt [0/1]	1	1 V
VIDCONF	Allow Video Conferencing [0/1]	1	1 V
ENBLUTB	Enable Youtube [0/1]	1	1 V
FRCENCBK	Force Encrypted Backup [0/1]	1	1 V
FRCITUNP	forceITunesStorePasswordEntry [0/1]	1	1 V
FIGUNLCK	allowFingerprintForUnlock [0/1]	1	1 V
HOSTPAIR	allowHostPairing [0/1]	1	1 V

CNTRLSCR	allowLockScreenControlCentre [0/1]	1	1 V
NOTISCR	allowLockScreenNotificationsView [0/1]	1	1 V
TODAYSCR	allowLockScreenTodayView [0/1]	1	1 V
MG2UNMG	allowOpenFromManagedToUnmanaged	1	1 V
UNMG2MG	allowOpenFromUnmanagedToUnmanaged		1 V
PKIUP	OTAPKIUpdates [0/1]	1	1 V
PASSLCK	allowPassbookWhileLocked [0/1]	1	1 V
EXPCONT	allowExplicitContent [0/1]	1	1 V
ADGMFRD	allowAddingGameCentreFriends	1	1 V
ENBLSAF	Enable Safari [0/1]	1	1 V
DATAROAM	allowGlobalBackgroundFetchWhenRoaming	1	1 V
CONFPFROF	allowUIConfigurationProfileInstallation	1	1 V
ADTRCK	forceLimitAdTracking [0/1]	1	1 V
ENBLBACK	Enable to take backup on an iCloud account.	1	1 V
ENBLDOCS	Enable to sync documents with an iCloud account.	1	1 V
ENBLPHOT	Enable the use of photo stream functionality.	1	1 V
ENBLGAME	Enable Game Centre [0/1]	1	1 V
SCRNCAP	Enable Screen Capture [0/1]	1	1 V

ACCTMOD	allowAccountModification [0/1]	1	1 V
AIRDROP	Allow Air Drop [0/1]	1	1 V
MOB DATA	Allow Mobile Data [0/1]	1	1 V
ASISTCON	allowAssistantUserGeneratedContent [0/1]	1	1 V
CLDKEYSY	allow CloudKeychainSync [0/1]	1	1 V
DIAGSUB	AllowDiagnosticSubmission [0/1]	1	1 V
MYFRNDMD	FindMyFriendsModification [0/1]	1	1 V
ENBLMPLR	Enable Multiplayer Game [0/1]	1	1 V
ZSTOP001	Governs the profile that is to be provisioned. Identifies that the profile is applicable only for iOS devices.	WORDPOS('IOS', SUBSTR(EDMGETV (ZMASTER,ZOS),1,3))=0	255 E
ZSTOP002	Stop Resolution Expression Expressions evaluating to "true" in ZSTOP attributes cause resolution of the instance to be skipped. If left blank, the instance is not skipped, and resolution continues. This is useful for assigning a version to a specific set of users. Use the CSDB Editor to set this attribute.	-	255 E

Mail Class

File:	PRIMARY
Domain:	MOBILE
Class:	MAIL
Icon:	
Description	Use an instance of this class to create email configuration profile for mobile devices.

Mail Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
NAME	Friendly name		255 V
DESCRIPT	Description		255 V
ACCTNAME	Email Account Name		255 V
ADDRESS	Email Address		255 V
TYPE	Email Account Type		255 V
INAUTH	Incoming Mail Server Authentication		255 V
INHOST	Incoming Mail Server Host Name		255 V
INPREFIX	Incoming Mail Server IMAPPathPrefix		255 V
INPORT	Incoming Mail Server Port		255 V
INSSL	Incoming Mail Server Use SSL		1 V
INUSER	Incoming Mail Server User Name		255 V
INPASSWD	Incoming Mail Server Password		255 V
OAUTH	Outgoing Mail Server Authentication		255 V
OHOST	Outgoing Mail Server Host Name		255 V
OPORT	Outgoing Mail Server Port		6 V
OSSL	Outgoing Mail Server Use SSL		1 V
OUSER	Outgoing Mail Server User Name		255 V

OSAMEPWD	OutgoingPasswordSameAsIncomingPassw		1 V
OPASSWD	Outgoing Password		255 V
DISRSYNC	Disable Mail Recents Syncing		1 V
SMIME	SMIMEEnabled		1 V
MOVE	Prevent Move		1 V
APPSHEET	PreventAppSheet		1 V
ONLMAIL	Use Only in Mail		1 V
ENCRYPT	SMIMEEncryptionCertificateUUID		50 A
SIGN	SMIMESigningCertificateUUID		50 A

MDM (MDM) Class

File:	PRIMARY
Domain:	MOBILE
Class:	MDM
Icon:	
Description	An instance of this class lists the Sentient servers that are configured for mobile device management (MDM).

MDM Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
NAME	Friendly name of the Satellite server that provides MDM functionality.	NULL	32 V
SERVER	Host name of the server that provides MDM functionality.	NULL	255 V

DESCRIPT	Description of the full-service Satellite server that provides MDM functionality.	NULL	255 V

SCEP (SCEP) Class

File:	PRIMARY
Domain:	MOBILE
Class:	SCEP
Icon:	
Description	Use an instance of this class to add a SCEP server. For the current release, only base instance is supported; an administrator cannot create multiple instances of this class.

SCEP Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
NAME	Reserved for future use.	-	32
SCEPNAME	The name of the SCEP server.	-	255 V
URL	The URL of the SCEP server.	-	255 V
CHLNG	Challenge key got from the scep svr. The enrollment challenge password that you obtain from the SCEP server administrator.	-	255 V
RTRYDEL	Retry delay	-	255 V
ENABLED	SCEP Enabled	0	1 V
RETRY	Number of Retries.	-	255 V
DESCRIPT	Description	-	255 V
ZSTOP	ZSTOP expression	-	255 E

Scheduling (TIMER) Class

File:	PRIMARY
Domain:	MOBILE
Class:	TIMER
Icon:	
Description	Use the TIMER Class to schedule service deployments. For more information, see <i>Sentient Enterprise Console Online Help</i> .
Can connect to	ZSERVICE

TIMER Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
FREQ	The frequency at which the timer should be run (hours/days).	1	255 V
NAME	Friendly Name	Base Instance	80 V
ZSCHFREQ	Frequency [DAY/HOUR/MINUTE] Use this attribute to specify how often the timer should expire. DAY = expire one time HOUR = expire repeatedly	DAY	10 V
ZSCHTYPE	Type [IMMEDIATE/DEFERRED]	IMMEDIATE	10 V
ZSTOP	Stop Expression Expressions evaluating to "true" in ZSTOP attributes cause resolution of the instance to be skipped. If left blank, the instance is not skipped, and resolution continues. This is useful for assigning a version to a specific set of users. Use the CSDB Editor to set this attribute.	-	255 E

Settings (SETTINGS) Class

File:	PRIMARY
Domain:	MOBILE

File:	PRIMARY
Class:	SETTINGS
Icon:	
Description:	This class stores the MDM settings that are passed to the agent. The values for the base instance of this class are populated from the Sentient Core console.

SETTINGS Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ANRDPWD	Android default password. The password using which you can unlock a device after an administrator resets the password.	c2VjcmV0	255 V
LOG	Enable the Verbose Agent Logging	0	1 V
MDMPNAME	MDM configuration profile name. The name of the profile that a mobile device user views during the device enrollment process	RCA MDM Profile	255 V
NAME	Friendly Name The friendly name of the instance.	Base Instance	32 V
PAYLOADD	Configuration payload desc. The description of the security profile. This profile contains details on the security policies that an administrator has applied on the device.	RCA Sec Profile	255 V
PAYLOADN	Server configuration payload name. The name of the security profile. This profile contains details on the security policies that an administrator has applied on the device.	RCA Sec Profile	255 V
PROJCTID	Project ID. The project ID that you received when you created your Google API project. This is required to validate that the Sentient Android application is registered to send messages to the mobile device.	-	255 V

VPN Class

File:	PRIMARY
Domain:	MOBILE
Class:	VPN
Icon:	
Description	Use an instance of this class to create VPN configuration profile for mobile devices.

VPN Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
NAME	Friendly name		255 V
DESCRIPT	Description		255 V
USERNAME	User Define Name		255 V
OVERRIDE	Send all traffic through VPN		1 V
VPNTYPE	Settings type of vpn connection		255 V
AUTHNAME	Acc user name,used for L2TP,PPTP		255 V
PASSWORD	Used for L2TP and PPTP		255 V
CRADDRES	IP or Host name of VPN server		255 V
CCP40	ccpe enabled used for PPTP		1 V
CCP128	ccpe 128 enabled used for PPTP		1 V
CCP	Enable encryption on the connection		1 V

REMOTEAD	Remote Address IP or Host name		255 V
AUTHMETH	auth method shared secret or certif		255 V
XAUTH	User name for VPN acc for cisco		255 V
XAUTHENA	1 is xauth is on 0, if its off		255 V
LOCALID	Present only if authmethod is share		255 V
LOCALTYP	Present if authmethos is sharedsec		255 V
SHARED	Shared secret for vpn account		255 V
HYBRIDAU	Hybrid authentication		1 V
PROPSWD	Prompt for password		1 V
VPNPIN	Prompt for a PIN when connecting		1 V

Wifi Class

File:	PRIMARY
Domain:	MOBILE
Class:	WIFI
Icon:	
Description	Use an instance of this class to create WIFI configuration profile for mobile devices.

Wifi Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
NAME	Friendly name		255 V

DESCRIPT	Description		255 V
SSID	SSID of the WI-Fi network to be used		255 V
HIDDEN	Hidden network		1 V
AUTOJOIN	Auto joining of network. If true, network is auto joined.		1 V
ENCRYPT	Encryption type: WEP, WPA, and None.		255 V
PROXYTYP	Proxy type: none, manual, and auto.		255 V
PASSWORD	Optional Password		255 V
TLS	Accepted EAP Types		1 V
LEAP	Accepted EAP Types		1 V
PEAP	Accepted EAP Types		1 V
EAPFAST	Accepted EAP Types		1 V
EAPSIM	Accepted EAP Types		1 V
USEPAC	Configuration of PAC		1 V
FASTPAC	Configuration of PAC		1 V
FASTPACA	Configuration of PAC		1 V
USER	Username for connection to WIFI network		255 V
PASSWD	Password for connection to WIFI network		255 V
EAPCLIEN	EAPclientconfiguration dictionary		255 V
PROXYSER	Proxy server's network address.		255

			V
PROXYPOR	Proxy server's port.		255 V
PUSERNAM	Proxy user name		255 V
PPASSWOR	Proxy password		255 V
PROXYPAC	PAC file that defines proxy config.		255 V
PPACFALL	Proxy PAC fallback allowed		1 V

Windows Profile (WINPOL) Class

File:	PRIMARY
Domain:	MOBILE
Class:	WINPOL
Icon:	
Description	Use an instance of this class to create security profile for Windows Phone devices.
Can connect to	ZSERVICE

WINPOL Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
RESRQD	Restriction required	0	1 V
Name	Friendly Name	Base Instance	32 V
DESCRIPT	Description of the Windows Profile		255 V
PWDRQD	Passcode Required	1	1 V
PWDSIM	Allow Simple	1	1

	Passcode		V
PWDALNUM	Alphanumeric Passcode	0	1 V
PMINLENT	Minimum Passcode Length	4	255 V
PWDMINCX	Minimum Number of Complex Characters	1	255 V
PWDMXMAG	Minimum Passcode Age (0-730 days)	0	3 V
MAXTMLCK	Auto-Lck for the Mobile Devices		6 V
PWDHIST	Passcode History	0	5 V
MAXPWDWD	Maximum Failed Attempts	0	2 V
DEVENCR	Device Encryption [0/1]	1	1 V
ENBLSC	Enable Storage Card [0/1]	1	1 V
WIPE	Wipe Mobile Device		1 V
ZSTOP001	STOP Expression 001	WORDPOS ('WP8', SUBSTR (EDMGETV (ZMASTER, ZOS), 1, 3))=0	255 E
ZSTOP002	STOP Expression 002		255 E

The PRDMAINT Domain

Use the PRDMAINT Domain to store packages for self-maintenance.

Application (ZSERVICE) Class

File:	PRIMARY	Description:	A ZSERVICE instance represents a unit of managed content that can be connected to many other class instances. Defining an instance of the ZSERVICE Class within the PRDMAINT Domain should be exclusively for Agent and Admin self-maintenance purposes.
Domain:	PRDMAINT		
Class:	ZSERVICE		

Icon:		Can connect to:	PACKAGE, FILE, BEHAVIOR, DIALOG, DEPT, WORKGRP, TIMER, VGROUP, STAGER, HTTP, EXECUTE, ZSERVICE, PDACONFIG

ZSERVICE Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ALWAYS	Contains Class connections determine the path of resolution of an agent's desired state. An _ALWAYS_ class connection is always resolved.		50 C
ALWAYS	Contains An _ALWAYS_ class connection is always resolved. An Includes connection is used to express inter-package relationships.		50 I
ALWAYS	Utility Resolution Method Any method you specify for this attribute is unconditionally executed by Client Automation when this instance is resolved. A valid method name, for example, ZSYSTEM.ZMETHOD.PUT_ZERROR_HIST.		51 M
ADAPTIVE	Auto Adaptability [Y/N] Indicates whether the installed package is dependent on client settings that must be monitored periodically. If the settings change, the client must reconnect to the Configuration Server to get new or different components. Useful for "plug and play" services.		1 V
AUTHOR	Author Name Name of the author of the service that appears in the properties for the service in the Service List.		50 V
BDELETE	Service Pre-Delete Method Method to run before deleting the service.		100 M
BREPAIR	Service Pre-Update Method Method to run before updating the service.		100 M
BUPDATE	Service Pre-Repair Method Method to run before repairing the service.		100 M
CACHE	App Element Caching [Y/N]	N	1 V

CACHELIM	Percent Disk Limit for Cache Cache limit, which is defined as the percentage of used drive space. Type a number between 000 and 100. Note: Used for Windows Installer applications only. If the percentage of used space is greater than the cache limit, then all of the cached files for the product are removed and the cache folder is deleted. This is checked after every file is cached on the disk.	000	3 V
CACHELOC	CACHE Location on Client For Windows Installer applications only. Location of the folder on the agent computer that is used to cache the compressed application files needed for the product. Sentient support for Windows Installer tags the PRODGUID value to this value to create the folder. For example, If CACHELOC=c:\progra~1\PSL, and PRODGUID = 12345_XXXX\cache, the cache folder would be: c:\progra~1\PSL\12345_XXXX\cache. Note: The folder name \cache is automatically appended to PRODGUID. If you are not deploying a Windows Installer-enabled application, the files will be cached in IDMDATA.	_UNDEF_	254 V
CATGROUP	Catalog Group Name Type a name for the catalog to be displayed to subscribers when they click Properties in the Application Self-service Manager user interface.		30 V
COMPSIZE	Application Size – Compressed The size of the compressed application displayed to the subscribers when they click Properties in the Application Self-service Manager user interface. Cumulative value of the COMPSIZE defined in the PACKAGE Class.		12 V
DELDATE	Delete Date Indicates when the application was removed (in		32 V

	local time) from the agent computer. The Agent manages and maintains this attribute.		
DESCRIPT	Application Description Description of the service that appears in the properties for the service in the Service List. Value is set initially in the Long Description field in the New Application Wizard (used to create a service).		100 V
ERTYPE	Event Reporting Method [O/E/X] Set on the Configuration Server, this sends an APPEVENT object to the Configuration Server. O = Object Currently supports object format only.	O	1 V
EVENTS	Events to Report Indicates which events to report on. Event to report on: AI = Application Install AD = Application De-install AU = Application Update AR = Application Repair AV = Application Verify VA = Version Activation VD = Version Deactivation UM = Upgrdmaint Maintenance What (about the event) to report on: S = Success F = Failure B = Both Success and Failure N = None	AI=B AD=B AU=B AR=B AV=F VA=B VD=B UM=B	50 V
INSTDATE	Installed Date Indicates when the application was installed (in local time) on the agent computer. The Agent manages and maintains this attribute. The date is in text form, as in: Jan 28, 1998 16:10:00.		32 V
ITEMS	Number of Component Instances		7 V
LREPAIR	Local Repair [Y/N] Enables local repair of broken applications. If an application is broken because of missing files, the files (stored locally) can be used to repair the application.		1 V
MCELIBL	Service Multicast Eligible [Y/N] Indicates if the application is eligible for multi-casting.	Y	1 V

NAME	Friendly Name Friendly name for the service. This appears in the properties for the service in the Service List.		80 V
OWNER	Application Contact Package owner, as documented by the Admin Publisher.		50 V
PRICE	Price Type in the price of an application to be displayed to subscribers when they click Properties in the Application Self-service Manager user interface.		10 V
PUBDATE	Published Date of Service Date service was published.		32 V
REBOOT	Install/Update/Delete/Version Change Used to restart the agent computer after the service is installed. Event to report on: AI = Install AD = Deinstall AU = Update AR = Repair AV = Verify Type of reboot: S = Soft Boot (Default of type Y panel.) H = Hard Boot (Default of type A panel.) N = None Type of panel: Q = No panel. A = OK button only. Y = OK and Cancel button. Type of connect: None specified: Reboot on Machine connect (context = m). U = reboot on user connect only (context = u). MU = reboot when both machine and user parts of the service have been installed. Example: AI=S performs a soft boot on application installation.		64 V
RECONFIG	Reconfiguration Enabled [Y/N] Indicates whether an application can be relocated after it has been installed.		1 V
REMOVAL	Un-Managed Behavior [A/D/U] Controls removal of the service. If ZVCMO is set to M, set REMOVAL to: A = Abandon (deletes the objects, but not the components)	D	1 V

	D = Delete (deletes the objects and components) If ZVCMO is set to O, set REMOVAL to: A = Abandon (deletes the objects, but not the components) D = Delete (deletes the objects and components) U = Unmanage (does not delete the objects or components)		
RSTRSIZE	Download Restart Threshold (bytes)	0	11 V
RUNDLG	Dialog Processing [Y/N] Specifies whether to enable processing of dialog boxes during the installation of the service.	N	1 V
RUNLEVEL	Runlevel Support for UNIX.		1 V
SCHEDOK	Update Schedule Locally [Y/N] Specifies whether the subscribers are allowed to change the update schedule for the service locally. Y = Subscriber is allowed to change the schedule. N = Configuration Server controls the update schedule.		1 V
SIZE	Application Size – Uncompressed: The size of the uncompressed application displayed to the subscribers when they click Properties in the Application Self-service Manager user interface.		12 V
UIOPTION	Progress Indicator [NONE/FULL/INFO] Controls whether the status window appears. NONE = No interface appears. FULL = Interface appears and Cancel button is available. INFO = Interface appears with no option to cancel.		4 V
UIOPTMSI	MSI Progress Indicator [NONE/FULL/INFO] Controls whether the MSI status window appears. NONE = No interface appears. FULL = Displays what is happening on the agent computer and provides the subscriber		

	with controls to make changes. INFO = Displays what is happening on the agent computer, but disables all the controls so that the subscriber cannot make any changes. See the Setup log files and Event Viewer for warnings, errors, successes, failures, and other details.		
UPDDATE	Upgrade Date (Programmatic) Stores the date when the file was updated on the Configuration Server.		32 V
UPGDATE	When Application was Upgraded Stores the date when the application was last upgraded (in local time) on the agent computer. This value is set by the Sentient user after the dialogs have been installed. This date can be used for reporting purposes. The date is in text form, as in: Jan 28, 1998 16:10:00.		32 V
URL	WEB URL Name Address of a Web page where the subscriber can find additional information about the service.		100 V
VENDOR	Vendor Name Name of the vendor of the service.		24 V
VERDATE	Verified Date of Service Indicates when the application was last verified (in local time) on the agent computer. The Agent manages and maintains this attribute.		32 V
VERSION	Version Description: Version of the software. This appears in the properties for the service in the Service List.		50 V
ZAVIS	Available, Verified, Installed, Sync F. The Agent manages and maintains this attribute to show the different states of the application in the Service List. Y = Yes N = No X = Unknown The four states are: Available indicates whether a service is available from the Configuration Server. Verified indicates whether a service has been verified. Installed indicates whether the service has been installed. Synchronized indicates whether the installed	YXNX	4 V

	service has all of the latest changes from the Configuration Server.		
ZCONTEXT	Component Context [M/U]		1 V
ZCREATE	Service Installation Method Sentient-supplied method or fully qualified, user-defined method.		100 M
ZDELETE	Service Delete Method Sentient-supplied method or fully qualified, user-defined method.		100 M
ZDISCONN	Disconnect on Install [Y/N] Allows the client to disconnect from the Configuration Server if there is an open session with the Configuration Server. Y = Disconnects the agent from the Configuration Server. N = Does not disconnect from the agent from the Configuration Server.	Y	1 V
ZINIT	Service Initialization Method A fully qualified, user-defined method.		100 M
ZREPAIR	Service Repair Method Name of the method to repair the service.		100 M
ZSTOP000	Stop Resolution Expression Expressions evaluating to "true" in ZSTOP attributes cause resolution of the instance to be skipped. If left blank, the instance is not skipped, and resolution continues. This is useful for assigning a version to a specific set of users. Use the CSDB Editor to set this attribute.	EDMGETB (ZMASTER .ZOBJFLOR) ='PACKAGE'	100 E
ZSTOP001 to ZSTOP002	Expression Resolution Method - 001		100 E
ZSVCCAT	Service Visible in Catalog? [Y/N]		1 V
ZSVCCSTA	Service Status on Client (999) Status code for the service that leads you to the reason why files for a service may not be deployed correctly. 000 = All actions for service completed. 008 = At least one resource had a writer error. 014 = No disk space for at least one resource. 015 = ZDLIMIT exceeded for at least one	999	3 V

	resource. 024 = A method timed out. 036 = A create method set ZBRC>7. 900 = Not all files were verified. No work done for service.		
ZSVCMO	Mandatory/Optional Service [M/O/MO/OM] Indicates whether the service is mandatory or optional on the Agent. If you are using Application Manager <i>and</i> Application Self-Service Manager, you could also specify mandatory and then optional, MO, or optional then mandatory, OM. The first character indicates how the application should be handled before installation. The second character indicates how the application should be handled after installation.	O	1 V
ZSVCMODE	Application Context [M/U/MU] Specifies whether the service is to be installed in the Machine or User context. M = Machine U = User		2 V
ZSVCNAME	Service Name/Description	Unknown Service	24 V
ZSVCPRI	Service Create Ordering [01-99] Enables you to specify a priority level for this service. If disk space on the client desktop is not available, services will be downloaded based on the priority level set here. 01 = highest priority 10 = Default value 99 = Lowest priority		2 V
ZSVCRASH	Verify Attributes [Y/F/D/N] The agent will check permissions on files and/or directories when ZSVCRASH is set. Valid values are: Y – verify files and directories F – verify files only D – verify directories only N – do not verify permissions		1 V
ZSVCTTYP	Application Target Type [A/S] Indicates which agent will receive this application. A = Application Manager S = Application Self-service Manager		2 V
ZSYSACCT	Install Under System Account [Y/N/M] Specifies whether to install the service under	Y	1 V

	Y = The application is installed using the system rights. N = The application is installed using the rights of the user who is logged on. M = The application is installed using the system rights if the current connect is in the machine context (context =m).		
ZUPDATE	Service Update Method Sentient-supplied method or fully qualified, user-defined method.		130 M
ZVERIFY	Service Verify Method		100 M

Application Packages (PACKAGE) Class

File:	PRIMARY	Description:	An instance of the PRDMAINT Domain Application Packages (PACKAGE) class represents one software application or unit of other content to be managed on agent computers. A PACKAGE in the PRDMAINT Domain should be used exclusively to house self-maintenance applications.
Domain:	PRDMAINT		
Class:	PACKAGE		
Icon:		Can connect to:	DESKTOP, FILE, PATH, MSIFEATS, MSI, PACKAGE, PDACONFIG, REGISTRY

PACKAGE Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ALWAYS	Contains Class connections determine the path of resolution of an agent's desired state. An _ALWAYS_ class connection is always resolved.		50 C
ALWAYS	Package Includes Class connections determine the path of resolution of an agent's desired state. An _ALWAYS_ class connection is always resolved.		50 C
BEHAVIOR	Behavior		255 I
INCLUDES	Package Includes		50 I

COMPSize	Package Compressed Resource Size		12 V
DESCRIPT	Application Description Description of the service that appears in the properties for the service in the Service List. Value is set initially in the Long Description field in the New Application Wizard (used to create a service).		255 V
DESKTOP	Desktop		255 I
FILE	File		255 I
ITEMS	Number of Component Instances		7 V
NAME	Friendly Name		80 V
OWNER	Application Contact Package owner, as documented by the Admin Publisher.		80 V
PATH	Path		255 I
PKGNAME	Package Name		80 V
REGISTRY	Registry		255 I
RELEASE	Package Release Release level of the package, as documented by the Admin Publisher.		80 V
REQUIRES	Required Package Level		8 R
SIZE	Package Compressed Resource Size		12 V
UNIXFILE	UNIX File		255 I
ZSTOP000 to ZSTOP003	Stop Resolution Expression Expressions evaluating to "true" in ZSTOP attributes cause resolution of the instance to be skipped. If left blank, the instance is not skipped, and resolution continues. This is useful for assigning a version to a specific set of users. Use the CSDB Editor to set this attribute.		255 E

File Resources (FILE) Class

File:	PRIMARY	Description:	An instance of this class lets you define and specify characteristics of an application such as the file's name, and size. Instances of the PRDMAINT File Resources (FILE) class are specific to the applications used for administration and client self-maintenance.
Domain:	PRDMAINT		
Class:	FILE		
Icon:		Can connect to:	

FILE Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ALWAYS	Connect to Class connections determine the path of resolution of an agent's desired state. An _ALWAYS_ class connection is always resolved.		50 C
ALWAYS	Connect to	PRDMAINT .MAINTLOC .MAINTENANCE	50 C
FIXNUM	File Fix Number		8 V
FVERSION	File Version	_UNDEF_	30 V
LEVEL	Maintenance level [V,R,S,F] A collection of the FILE variables RELEASE, SPLEVEL and FIXNUM. For example: RELEASE = 4.0 SPLEVEL = 0 FIXNUM = 2091 LEVEL = 4.0.0.2091		20 V
LOCATION	Location		255 V
NAME	Friendly Name	Defaults	80 V
OBJGUID	Instance GUID Object Global Unique Identification – <i>for future use.</i>	_UNDEF_	32 V

PRODUCT	Product identifier		8 V
PVERSION	Product Version	_UNDEF_	30 V
RELEASE	Product release level		6 V
SIGTYPE	File Signature Type		3 V
SPEVEL	File service pack		3 V
ZCMPKEY	Compression Key	_UNDEF_	80 V
ZCMPSIZE	Compressed File Size From promotion process if the compressed file option was selected. The size of the compressed file.		11 V
ZCONTEXT	Component Context		8 V
ZCREATE	Method to Install Resource Sentient-supplied method or fully qualified, user-defined method.		50 M
ZDELETE	Method to De-install Resource Sentient-supplied method or fully qualified, user-defined method.	RADREMF	50 M
ZFILEUPD	Client File Update/Add Method If a file has been updated by an administrator, this method will be executed. Sentient-supplied method or fully qualified, user-defined method.		50 M
ZINIT	Resource Initialization Method A fully qualified, user-defined method.		50 M
ZOBJPTCH	Perform Patching [Y/N]	N	1 V
ZOPENERR	Client Method on File Open Error Sentient-supplied method or a fully qualified, user-defined method.	RADLKM	50 M
ZPUBVER	Publisher Version	_UNDEF_	8 V
ZREPAIR	Client Instance Repair Method		50

			M
ZRSCCFIL	Resource File Name Specific file to be distributed. A generic pattern (a wildcard character, such as *) or an explicit filename.		255 V
ZRSCCKPT	Resource Checkpoint		1 V
ZRSCCONF	Confirm File Download [Y/N] Y enables you to confirm that the resource was downloaded.	Y	1 V
ZRSCCRC	Resource CRC From the promotion process if the CRC option was selected. Eight-character calculated hexadecimal value.		8 V
ZRSCCSTA	Client File Status	999	3 V
ZRSCDATE	Resource Date Stamp – From Promote Date stamp on resource file from promote process.		8 V
ZRSCMMEM	PDS Member Name	_UNDEF_	8 V
ZRSCMO	Mandatory/Optional [M/O]	M	1 V
ZRSCNAME	Resource Name	&ZRSCCFIL	30 V
ZRSCOBJN	Persistent Object Name		8 V
ZRSCPADM	Admin ID		30 V
ZRSCPRI	Create Sequence Priority Enables you to specify a priority level for this resource. If disk space on the client desktop is not available, resources will be downloaded based on the priority set here. 01 = Highest priority 10 = Default value 99 = Lowest priority	50	2 V
ZRSCRASH	DOS File Attribute [R/A/S/H]		4 V
ZRSCRSTR	Restart [Y/N/R]	N	3

	Set to Y to restart. Set to R to bypass Windows file caching and write directly to disk (allowing checkpoint restart to operate correctly).		V
ZRSCSIG	File Signature		32 V
ZRSCSIZE	Resource Size – From Promote Byte count of resource file from promote process.		11 V
ZRSCSRC	Resource Source, i.e., Publisher		20 V
ZRSCSTYP	Server File Type[BINARY/TEXT]	BINARY	8 V
ZRSCSVRB	Version Skip Rebuild Flag[Y/N]		1 V
ZRSCTIME	Resource Time Stamp Time stamp on resource file from promote process.		8 V
ZRSCVERS	Resource Version - From Promote Version number of resource from promotion process.		12 V
ZRSCVRFY	Verify Resource File on Connect Verifies the file statistics of a resource file on the desktop during each agent connect process. If you are deploying versioned resources, this attribute is ignored. The rules for the E(xist) flag apply to both Sentient- and non-Sentient distributed files. (For example, an Sentient-managed application file or <code>CONFIG.SYS</code> file.). Thus, you must specify Y or N for an existing file to be distributed. Y = The file will be distributed if one or more of the following conditions is met: The file has not been distributed before. The date, time, or size of the file has changed. The file was re-promoted. N = The file will be distributed if: The file has not been distributed before, Or, the file was re-promoted. E = The file will be distributed only if it does	MN	8 V

	not currently exist on the desktop. M = Reserved for Sentient maintenance files. During the last step of the connect, the file is moved from the <code>EDMDATA</code> subdirectory to its proper location. Example: can be used to update <code>EDMGETRS.EXE</code>. S = This file is distributed if the file size has changed. T = The file is distributed if the file time has changed. D = File is distributed if the file date has changed. If the <code>ZRSCVRFY</code> attribute is set to <code>P</code> and deployment of this resource and the service are marked as failed due to SFP processing. If you do not enforce SFP, the file is marked as having passed verification.		
	L = Forces the lock method to be used. The file will be automatically queued up to be deployed on restart, and the reboot flag will be set. If the <code>L</code> flag is not specified, and the file to be deployed is currently in use (locked) the lock method is automatically used anyway. What the <code>L</code> flag does is force the lock method to be always used to deploy this file, whether or not it is currently in use (locked). The <code>L</code> flag will also skip decompression and dispatch the open error method (<code>RADLKM</code>). This is used for system dlls that must be replaced. Note that for the file to be verified during subsequent connects, you must also set this variable to <code>Y</code>. U = File is distributed only if the date and time on the <code>CONFIGURATION SERVER</code> is newer than the date and time on the desktop. R = Resource Stamp. For internal use only. V = (Windows only) Used in when normal date/time stamp check cannot be used. Usually when a <code>.dll</code> or <code>.exe</code> is involved and the date and time stamp are changing frequently. <code>Radconnect.exe</code> checks the internal version number of the target file and compares it with the one that Sentient has. If the incoming file has a higher		

	internal version number than it will be placed on the desktop over the existing file . P = (Windows only) Set during packaging, enforces Windows System File Protection (SFP). If the verification is successful, the resource is in the desired state and no further verification is necessary. If the verification fails, Sentient checks to see if System File Protection protects the file. If the file is protected, it will not be deployed.		
ZUPDATE	Client Instance Update Method		50 M

Mac File Resources (MACFILE) Class

File:	PRIMARY	Description:	Mac operating systems only. An instance of this class lets you define and specify characteristics of an application such as the file's name, and size. Instances of the PRDMAINT Mac File Resources (MACFILE) class are specific to the applications used for administration and agent self-maintenance.
Domain:	PRDMAINT		
Class:	MACFILE		
Icon:		Can connect to:	

MACFILE Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ALWAYS	Connect to Class connections determine the path of resolution of an agent's desired state. An _ALWAYS_ class connection is always resolved.		50 C
LINKDEST	Symbolic Link Destination		30 V
LOCATION	Location		255 V
NAME	Friendly Name	Defaults	80 V
OBJGUID	Instance GUID Object Global Unique Identification – <i>for future use</i> .	_UNDEF_	32 V

SIGTYPE	File Signature Type		3 V
ZCMPKEY	Compression Key		80 V
ZCMPSIZE	Compressed File Size From promotion process if the compressed file option was selected. The size of the compressed file.		11 V
ZCREATE	Method to Install Resource Sentient-supplied method or fully qualified, user-defined method.		50 M
ZDELETE	Method to De-install Resource Sentient-supplied method or fully qualified, user-defined method.	RADRMEF	50 M
ZFILEUPD	Client File Update/Add Method If a file has been updated by an administrator, this method will be executed. Sentient-supplied method or fully qualified, user-defined method.		50 M
ZINIT	Resource Initialization Method A fully qualified, user-defined method.		50 M
ZOBJPTCH	Perform Patching [Y/N]	Y	1 V
ZOPENERR	Client Method on File Open Error Sentient-supplied method or a fully qualified, user-defined method.		50 M
ZPERGID	File Group Name		32 V
ZPERUID	File Owner Name		32 V
ZREPAIR	Client Instance Repair Method		50 M
ZRSCCFIL	Resource File Name Specific file to be distributed. A generic pattern (a wildcard character, such as *) or an explicit filename.	_BASE_ _INSTANCE_	255 V
ZRSCCKPT	Resource Checkpoint		1 V
ZRSCCONF	Confirm File Download [Y/N] Y enables you to confirm that the resource was downloaded.	Y	1 V

ZRSCCRC	Resource CRC From the promotion process if the CRC option was selected. Eight-character calculated hexadecimal value.		8 V
ZRSCCSTA	Client File Status	999	3 V
ZRSCDATE	Resource Date Stamp Date stamp on resource file from promote process. A value up to eight digits long.		8 V
ZRSCMO	Mandatory/Optional [M/O]	M	1 V
ZRSCNAME	Resource Name	&ZRSCCFIL	30 V
ZRSCPADM	Admin ID		30 V
ZRSCPRI	Create Sequence Priority Enables you to specify a priority level for this resource. If disk space on the client desktop is not available, resources will be downloaded based on the priority set here. 01 = Highest priority 10 = Default value 99 = Lowest priority	50	2 V
ZRSCRASH	File Permissions		4 V
ZRSCRSTR	Restart [Y/N] Set to Y to restart.	Y	3 V
ZRSCSIG	File Signature		32 V
ZRSCSIZE	Resource Size Byte count of resource file from promote process.		11 V
ZRSCSRC	Resource Source, that is Publisher		20 V
ZRSCSTYP	Server File [BINARY/TEXT]	BINARY	8 V
ZRSCSVRB	Version Skip Rebuild Flag [Y/N]		1 V
ZRSCTIME	Resource Time Stamp Time stamp on resource file from promote		8 V

ZRSCVERS	Resource Version - From Promote Version number of resource from promotion process.		12 V
ZRSCVRFY	Verify Resource File on Connect Verifies the file statistics of a resource file on the desktop during each agent connect process. If you are deploying versioned resources, this attribute is ignored. The rules for the E(xist) flag apply to both Sentient- and non-Sentient distributed files. (For example, a Sentient-managed application file or <code>CONFIG.SYS</code> file.) Thus, you must specify Y or N for an existing file to be distributed. Y = The file will be distributed if one or more of the following conditions is met: The file has not been distributed before. The date, time, or size of the file has changed. The file was re-promoted. N = The file will be distributed if: The file has not been distributed before. Or, The file was re-promoted. E = The file will be distributed only if it does not currently exist on the desktop. M = Reserved for Sentient maintenance files. During the last step of the connect, the file is moved from the <code>EDMDATA</code> subdirectory to its proper location. Example: can be used to update <code>EDMGETRS.EXE</code>. S = This file is distributed if the file size has changed. T = The file is distributed if the file time has changed. D = File is distributed if the file date has changed.	Y	8 V
	L = Forces the lock method to be used. The file will be automatically queued up to be deployed on restart, and the reboot flag will be set. If the L flag is not specified, and the file to be deployed is currently in use (locked) the lock method is automatically used anyway. What the L flag does is force the lock method to be always used to deploy this file, whether or not it is currently in use (locked). The L flag will also skip decompression and dispatch the open error method (<code>RADLKM</code>). This is used for system dlls		

	that must be replaced. Note that for the file to be verified during subsequent connects, you must also set this variable to Y. U = File is distributed only if the date and time on the Configuration Server is newer than the date and time on the desktop. R = Resource Stamp. For internal use only. V = (Windows only) Used in when normal date/time stamp check cannot be used. Usually when a .dll or .exe is involved and the date and time stamp are changing frequently. Radconct.exe checks the internal version number of the target file and compares it with the one that Sentient has. If the incoming file has a higher internal version number than it will be placed on the desktop over the existing file. P = (Windows only) Set during packaging, enforces Windows System File Protection (SFP). If the verification is successful, the resource is in the desired state and no further verification is necessary. If the verification fails, Sentient checks to see if System File Protection protects the file. If the file is protected, it will not be deployed. If the ZRSCVRFY attribute is set to P and deployment of this resource and the service are marked as failed due to SFP processing. If you do not enforce SFP, the file is marked as having passed verification		
ZSTATUS	Client Status	999	3 V
ZUPDATE	Client Instance Update Method		50 M

Maintenance Location (MAINTLOC) Class

File:	PRIMARY	Description:	This class points to the instance that defines the self-maintenance options available in Sentient 3.0 and later.
Domain:	PRDMAINT		
Class:	MAINTLOC		
Icon:		Can connect to:	

MAINTLOC Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type

LOCATION	Maintenance Location	&(ZMASTER.ZROOTDRV)& (ZMASTER.ZROOTDIR)_MAINT_\\	255 V

Path (PATH) Class

File:	PRIMARY	Description:	This class stores the drive and directory required to install a resource. Packages can be relocated by updating instances of this class. Instances of the PRDMAINT Path (PATH) class are specific to the applications used for administration and client self-maintenance
Domain:	PRDMAINT		
Class:	PATH		
Icon:		Can connect to:	CATALOG, ZSERVICE

PATH Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
DRIVE	Drive	&(ZMASTER.ZSYSDRV)	20 V
DIR	Directory	&(ZMASTER.ZROOTDIR_\\ MAINT_\\	255 V
NAME	Friendly Name	Maintenance Staging Directory	255 V
OBJGUID	Instance GUID Object Global Unique Identification – <i>for future use.</i>	_UNDEF_	32 V
ZRSCPRI	Priority	5	2 V
ZRSCMO	Mandatory/Optional [M/O]	M	1 V

Product Maintenance (PRDMAINT) Class

File:	PRIMARY	Description:	Maintenance for the Agents is available from Technical Support. Maintenance includes import decks for the CSDB. New instances are created in the PRDMAINT Class. There is one PRDMAINT instance for each PRODUCT_PLATFORM_RELEASE combination. These instances are connected based on the agent's platform and current product level. Once you have decided to roll out the maintenance to the agent computers, you can add the service to the user's entitlements.
Domain:	PRDMAINT		
Class:	PRDMAINT		
Icon:		Can connect to:	PACKAGE, FILE, PATH

PRDMAINT Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ZSTOP000	Expression Resolution Method		100 E
ZSTOP001	Expression Resolution Method		100 E
NAME	Friendly Name		80 V
REQUIRES	Requires Connection		80 R

Scheduling (TIMER) Class

File:	PRIMARY	Description:	Use the TIMER Class to schedule service deployments. For more information, see <i>Sentient Enterprise Console Online Help</i> .
Domain:	PRDMAINT		
Class:	TIMER		
Icon:		Can connect to:	

TIMER Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type

ALWAYS	Connect to Class connections determine the path of resolution of an agent's desired state. An _ALWAYS_ class connection is always resolved.		50 C
APPSVC	Application		32 V
DOMAIN	Server Domain Name	&(ZMASTER.ZDOMNAME)	32 V
IPADDR	Server IP Address/Name	&(ZMASTER.ZIPADDR)	32 V
MGRNAME	Server Name	&(ZMASTER.ZMGRNAME)	32 V
NAME	Friendly Name		80 V
NETAVAIL	Check Network Availability [Y/N/W] Checks for network availability before executing the command		1 V
OBJGUID	Instance GUID Object Global Unique Identification – <i>for future use</i> .	_UNDEF_	32 V
PINGCNT	Number of Ping Attempts This attribute sets the number of ping attempts to be made by the Configuration Server.	3	2 V
PINGDLAY	Time between Pings (in msecs) This attribute sets the amount of time between pings in msecs.	2000	6 V
REQUEST	Application Request	"Apply Updates"	32 V
RUNSYNC	Synchronous Timer Execution [Y/N] This attribute sets the value of Yes or No for the synchronous timer execution. The default value is Yes.	Y	1 V
SOCKET	Server Socket Number	&(ZMASTER.ZDSTSOCK)	32 V
ZCHNNAME	Channel [DOMAIN] Name	&(ZSERVICE.ZOBJDOMN)	32

			V
ZCREATE	Client Create Method Sentient-supplied method or fully qualified, user-defined method. This value is inherited from the base instance and should not be changed.	radtimeq	80 M
ZDELETE	Client Delete Method Sentient-supplied method or fully qualified, user-defined method. This value is inherited from the base instance and should not be changed.	radtimeq	80 M
ZNOPING	Stop RCS Detection [Y/N/W/S/P] Controls the automatic sensing of a network connection between the agent computer and the Configuration Server. If set to S or P , additionally, it will ping SAP entries according to their priorities, along with the Configuration Server. An expired time will continually evaluate whether communications with the Configuration Server can be established. When communications are established, the command line associated with the time is executed. After executing the command line, the Scheduler service resumes normal evaluation of whether the timer has expired again. This attribute is very useful for mobile users. Note: In order to use this attribute, you must add it to the TIMER Class template.	Y	1 V
ZOBJPRI	Priority [00-99] Sets the priority for deployment of the ZTIMEQ object. The ZTIMEQ object is deployed relative to the other elements being deployed during the agent connect. The	90	2 V

	are deployed before the ZTIMEQ object. A value of 90 is inherited from the base instance and should not be changed.		
ZPRVNAME	Provider (Manager) Name	&(ZSERVICE.ZMGRNAME)	32 V
ZRSCCMDL	Command Line to Execute This attribute indicates the command line that is executed on the subscriber's computer when the timer expires.	radskman req=	255 V
ZSCHDEF	Time Parameter Indicates when the timer expires. The syntax varies depending on the frequency of expiration that can be DAILY, HOURLY, INTERVAL, NUMDAY, WEEKDAY, WEEKLY.		50 V
ZSCHFREQ	Frequency [PERIODIC/ONCE/RANDOM] Use this attribute to specify how often the timer should expire. ONCE = expire one time. PERIODIC = expire repeatedly. RANDOM = expire in random intervals.	PERIODIC	10 V
ZSCHMODE	Timer Owner [MANAGER/DEFAULT] This attribute specifies the timer owner. It is recommended that you leave the timer owner configuration value as DEFAULT.	DEFAULT	32 V
ZSCHTYPE	Type [IMMEDIATE/DEFERRED] <i>Used only when ZSCHFREQ = PERIODIC.</i> Set ZSCHTYPE to DEFERRED to indicate that the first time an event is attempted to be launched, it will be deferred until the <i>next</i> scheduled time, no matter when the timer instance is evaluated. This was designed to handle the case of a daily 4am (non-peak)	DEFERRED	10 V

	Example 1: Suppose you create and deploy a timer with the ZSCHDEF = DAILY (&ZSYSDATE, 4:00:00) If ZSCHTYPE = IMMEDIATE and it is: Before 4:00:00, the command in the instance will be executed the same day at 4:00:00 After 4:00:00, the command in the instance will be executed immediately If ZSCHTYPE = DEFERRED and it is: Before 4:00:00, the command in the instance will be executed the <i>next</i> day at 4:00:00 After 4:00:00, the command in the instance will be executed the <i>next</i> day at 4:00:00 Example 2: Suppose you create and deploy a timer with the ZSCHDEF = WEEKDAY(FRIDAY, 4:00:00) If ZSCHTYPE = IMMEDIATE and it is: Not Friday or Friday and before 4:00:00, the command in the instance will be executed on Friday at 4:00:00 Friday and after 4:00:00, the command in the instance will be executed immediately If ZSCHTYPE = DEFERRED and it is: Not Friday or Friday and before 4:00:00, the command in the instance will be executed a week later on Friday at 4:00:00 Friday and after 4:00:00, the command in the instance will be executed a week later on Friday at 4:00:00		
ZSTOP	Stop Expression Expressions evaluating to "true" in ZSTOP attributes cause resolution of the instance to be skipped. If left		255 E

ZSVCID	Service ID This attribute specifies the object ID of the Application instance that this Scheduling instance is connected to. This value is inherited from the base instance and should not be modified.	&(ZSERVICE.ZOBJID)	80 V
ZUPDATE	Client Update Method The Scheduler UPDATE method that runs on the agent computer. This value is inherited from the base instance and should not be changed.	radtimeq	80 M
ZVERIFY	Client Verify Method	radtimeq	80 M

Unix File Resources (UNIXFILE) Class

File:	PRIMARY	Description:	UNIX® operating systems only. The attributes of the PRDMAINT Unix File Resources (UNIXFILE) define the owner and group associations as well as the permissions of each published resource. Attributes of the PRDMAINT Unix File Resources (UNIXFILE) class are specific to the applications used for administration and client self-maintenance.
Domain:	PRDMAINT		
Class:	UNIXFILE		
Icon:		Can connect to:	

UNIXFILE Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ALWAYS	Connect to Class connections determine the path of resolution of an agent's desired state. An _ALWAYS_ class connection is always resolved.	PRDMAINT .MAINTLOC .MAINTENANCE	50 C
LOCATION	Location		255 V
NAME	Friendly Name	Defaults	80 V
OBJGUID	Instance GUID	_UNDEF_	32

	<i>for future use.</i>		V
SIGTYPE	File Signature Type		3 V
ZCMPKEY	Compression Key		80 V
ZCMPSIZE	Compressed File Size From promotion process if the compressed file option was selected. The size of the compressed file.		11 V
ZCREATE	Method to Install Resource Sentient-supplied method or fully qualified, user-defined method.		50 M
ZDELETE	Method to De-install Resource Sentient-supplied method or fully qualified, user-defined method.	RADRMEF	50 M
ZFILEUPD	Client File Update/Add Method If a file has been updated by an administrator, this method will be executed. Sentient-supplied method or fully qualified, user-defined method.		50 M
ZINIT	Resource Initialization Method A fully qualified, user-defined method.		50 M
ZOBJPTCH	Perform Patching [Y/N]	Y	1 V
ZOPENERR	Client Method on File Open Error Sentient-supplied method or a fully qualified, user-defined method.		50 M
ZPERGID	File Group Name		32 V
ZPERUID	File Owner Name		32 V
ZREPAIR	Client Instance Repair Method		50 M
ZRSCCFIL	Resource File Name Specific file to be distributed. A generic pattern (a wildcard character, such as *) or an explicit filename.	_BASE_ INSTANCE_	255 V
ZRSCCKPT	Resource Checkpoint		1 V

ZRSCCONF	Confirm File Download [Y/N] Y enables you to confirm that the resource was downloaded.	Y	1 V
ZRSCCRC	Resource CRC From the promotion process if the CRC option was selected. Eight-character calculated hexadecimal value.		8 V
ZRSCCSTA	Client File Status	999	3 V
ZRSCDATE	Resource Date Stamp Date stamp on resource file from promote process. A value up to eight digits long.		8 V
ZRSCMO	Mandatory/Optional [M/O]	M	1 V
ZRSCNAME	Resource Name	&ZRSCCFIL	30 V
ZRSCPADM	Admin ID		30 V
ZRSCPRI	Create Sequence Priority Enables you to specify a priority level for this resource. If disk space on the client desktop is not available, resources will be downloaded based on the priority set here. 01 = Highest priority 10 = Default value 99 = Lowest priority	50	2 V
ZRSCRASH	File Permissions		4 V
ZRSCRSTR	Restart [Y/N] Y to restart.	Y	3 V
ZRSCSIG	File Signature		32 V
ZRSCSIZE	Resource Size – From Promote Byte count of resource file from promote process.		11 V
ZRSCSRC	Resource Source, that is Publisher		20 V
ZRSCSTYP	Server File [BINARY/TEXT]	BINARY	8 V
ZRSCSVRB	Version Skip Rebuild Flag [Y/N]		1

			V
ZRSCTIME	Resource Time Stamp – From Promote Time stamp on resource file from promote process.		8 V
ZRSCVERS	Resource Version - From Promote Version number of resource from promotion process.		12 V
ZRSCVRFY	Verify Resource File on Connect Verifies the file statistics of a resource file on the desktop during each agent connect process. If you are deploying versioned resources, this attribute is ignored. The rules for the E(xist) flag apply to both Sentient- and non-Sentient distributed files. (For example, a Sentient-managed application file or <code>CONFIG.SYS</code> file.) Thus, you must specify Y or N for an existing file to be distributed. Y = The file will be distributed if one or more of the following conditions is met: The file has not been distributed before. The date, time, or size of the file has changed. The file was re-promoted. N = The file will be distributed if: The file has not been distributed before Or, the file was re-promoted. E = The file will be distributed only if it does not currently exist on the desktop. M = Reserved for Sentient maintenance files. During the last step of the connect, the file is moved from the <code>EDMDATA</code> subdirectory to its proper location. Example: can be used to update <code>EDMGETRS.EXE</code> . S = This file is distributed if the file size has changed. T = The file is distributed if the file time has changed. D = File is distributed if the file date has changed.	MU	8 V
	L = Forces the lock method to be used. The file will be automatically queued up to be deployed on restart, and the reboot flag will be set. If the L flag is not specified, and the		

	file to be deployed is currently in use (locked) the lock method is automatically used anyway. What the L flag does is force the lock method to be always used to deploy this file, whether or not it is currently in use (locked). The L flag will also skip decompression and dispatch the open error method (RADLKM). This is used for system dlls that must be replaced. Note that for the file to be verified during subsequent connects, you must also set this variable to Y. U = File is distributed only if the date and time on the Configuration Server is newer than the date and time on the desktop. R = Resource Stamp. For internal use only. V = (Windows only) Used in when normal date/time stamp check cannot be used. Usually when a .dll or .exe is involved and the date and time stamp are changing frequently. <code>Radconct.exe</code> checks the internal version number of the target file and compares it with the one that Sentient has. If the incoming file has a higher internal version number than it will be placed on the desktop over the existing file. P = (Windows only) Set during packaging, enforces Windows System File Protection (SFP). If the verification is successful, the resource is in the desired state and no further verification is necessary. If the verification fails, Sentient checks to see if System File Protection protects the file. If the file is protected, it will not be deployed. If the ZRSCVRFY attribute is set to P and deployment of this resource and the service are marked as failed due to SFP processing. If you do not enforce SFP, the file is marked as having passed verification.		
ZSTATUS	Client Status	999	3 V
ZUPDATE	Client Instance Update Method		50 M

The PATCH Domain

The PATCH Domain stores the list of available patches within the PATCH Class. See the *Sentient Enterprise Console Online Help* for additional information about this domain and the

Service Optimization function.

Patch (PATCH) Class

File:	PRIMARY	Description:	The PATCH Class contains one instance for each patch created when you use Service Optimization. The file signatures of the patch are used to identify the associated instance that contains the data for the patch.
Domain:	PATCH		
Class:	PATCH		
Icon:		Can connect to:	

Table Caption Outside Table: PATCH Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
FROMFDCI	Origin of FROM Resource: Original location of the From instance.		96 V
FROMOID	Object ID of FROM Resource: Object ID of the original From instance.		12 V
FROMSIG	Signature of FROM Resource: The content-based signature of the From file.		64 V
FROMSIZE	Size of FROM Resource: The size of the From file. This value is taken from the ZRSCSIZE of the From instance.		11 V
MESSAGE	Explanation of Status: Contains the explanation of the status. Reference this attribute to determine if a patch was built successfully.		255 V
NAME	Friendly Name: The value of ZRSCCFIL attribute in the To instance.		255 V
POS	OS where Patch was Built: The operating system on which the patch was built.		32 V
PVERSION	Patch Builder Version: The version of the patch builder.		3 V
SIGTYPE	Content-based Signature Type: For now, MD5 is the only signature type supported.		8 V
STATUS	Status of Patch: Queued, Rejected, or Built.		8 V
TOFDCI	Origin of TO Resource:		96

	The original location of the To instance.		V
TOOID	Object ID of TO Resource: The object ID of the original To instance.		12 V
TOSIG	Signature of TO Resource: The content-based signature of the To file.		64 V
TOSIZE	Size of TO Resource: The size of the From file. This value is taken from the ZRSCSIZE of the To instance.		11 V
ZCMPSIZE	Compressed File Size: From promotion process if the compressed file option was selected. The size of the compressed file.		11 V
ZCONTRL1	Patch Control 1: ZSTOPPCT is the percentage of target data when the patch will stop building the path.		64 V
ZCONTRL2	Patch Control 2: ZSTOPMAX is the maximum size of the patch.		64 V
ZCONTRL3	Patch Control 3: ZTIMEOUT is the time allotted to patch creation before aborting the Create.		64 V
ZRSCCSTA	Status of Patch Resource: Status of patch resource. The only valid value is 999.		3 V
ZRSCDATE	Date Patch was Built: Date stamp on patch file from promote process.		8 V
ZRSCSIG	Signature of Patch: The content-based signature of the patch type.		64 V
ZRSCSIZE	Size of Patch: Byte count of patch file from promote process.		11 V
ZRSCTIME	Time Patch was Built: Time stamp on patch file from promote process.		8 V
ZRSCTYPE	Type of Resource: The only valid value is PATCH.		8 V

The PATCHMGR Domain

The PATCHMGR Domain contains information about the software being managed and the methods used to deploy the software.

Application Package (PACKAGE) Class

File:	PRIMARY	Description:	The PACKAGE class contains the metadata for each vendor as well as the patch agent binaries.
Domain:	PATCHMGR		
Class:	PACKAGE		
Icon:		Can connect to:	

PACKAGE Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ALWAYS	Connect To Class connections determine the path of resolution of an agent's desired state. An _ALWAYS_ class connection is always resolved.		50C
ALWAYS	Package Includes Class connections determine the path of resolution of an agent's desired state. An _ALWAYS_ class connection is always resolved.		50C
INCLUDES	Package Includes		50I
REQUIRES	Requires Package Level		80R
OWNER	Application contact The Owner for the Package.		80V
RELEASE	Package Release The Release version for the package.		80V
PATH	Path Includes connection to PATH Class instances.		80I
DESKTOP	Desktop Includes connection to DESKTOP Class instances.		80I
FILE	File Includes connection to FILE Class instances.		80I
REGISTRY	Registry Includes connection to REGISTRY Class instances.		80I
REQUIRES	Required Package Level Generic Requires connection.		80R

BEHAVIOR	Behavior Includes connection to BEHAVIOR Class instances.		255 I
ZSTOP000 to ZSTOP003	Stop Resolution (000) Expressions evaluating to "true" in ZSTOP attributes cause resolution of the instance to be skipped. If left blank, the instance is not skipped, and resolution continues. This is useful for assigning a version to a specific set of users..		255 E
PKGNAME	Package Name		80 V
NAME	Friendly Name		80 V
DESCRIPT	Application Description Description of the service that appears in the properties for the service in the Service List.		255 V
SIZE	Package Resource Size Total size of the entire package.		12 V

AUTOPKG (AUTOPKG) Class

File:	PRIMARY	Description:	AUTOPKG class contains the connection to PACKAGE class for the patch agent scripts.
Domain:	PATCHMGR		
Class:	AUTOPKG		
Icon:		Can connect to:	

AUTOPKG Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
WIN32	Package Connection for WIN32 Includes connection to Windows PACKAGE Class instance.		80 A
HPUX	Package Connection for HPUX Includes connection to HPUX PACKAGE Class instance.		80 A
COMMON	Package Connection for COMMON Includes connection to Common PACKAGE Class instance.		80 A
SUSE	Package Connection for SUSE		80

	Includes connection to SUSE PACKAGE Class instance.		A
REDHAT	Package Connection for REDHAT Includes connection to RedHat PACKAGE Class instance.		80 A
LINUX	Package Connection for LINUX Includes connection to Linux PACKAGE Class instance.		80 A
SOLARIS	Package Connection for SOLARIS2 Includes connection to Solaris PACKAGE Class instance.		80 A
AIX	Package Connection for AIX Includes connection to AIX PACKAGE Class instance.		80 A

vBulletin (BULLETIN) Class

File:	PRIMARY	Description:	BULLETIN class contains the connection to PATCH and LOCATION class instance which contributes to the bulletin.
Domain:	PATCHMGR		
Class:	BULLETIN		
Icon:		Can connect to:	

BULLETIN Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
NAME	Bulletin ID Name for the bulletin.		32 V
ID	Unique Identifier CSDB ID for that bulletin, used for the connection to zservices.		12 V
TITLE	Title Title for the bulletin as per vendor documentation.		255 V
POSTED	Bulletin Posted On Bulletin posted date by vendor.		18 V
REVISED	Bulletin Revised On Bulletin revised date by vendor.		18 V

SUPPORT	Support [Y/N]		I V
URL	Bulletin URL Link to the bulletin page on the vendor site.		255 V
FAQURL	FAQ URL Link to the FAQ page on the vendor site		255 V
IMPACT	Impact ID		8 V
PREREQ	Prereq ID		8 V
MITIGATE	Mitigate ID		8 V
POPULAR	Popularity ID		8 V
CTIME	Bulletin Created on (ISO-8601) Date and time of the bulletin created in the CSDB		24 V
MTIME	Bulletin Modified on (ISO-8601) Date and time of the bulletin last modified in the CSDB		24 V
SOURCE	Source [MICROSOFT/NOVADIGM/CUSTOM] Source for the bulletin metadata. Usually Vendor name if bulletin metadata from the vendor site. NOVADIGM if Persistent Systems provided metadata. CUSTOM if its customized.		32 V
MANAGE	Manages Patches	SYSTEM. ZMETHOD. PATCH_ RESOLVE	80 M
PRELOAD	Preload of Patch Data Connection to the LOCATION class instance for the bulletin ID	LOCATION. &(ID)_*	32 V
TYPE	Bulletin Type Type of bulletin, usually "Security Updates"		40 V
PLATFORM	Platform Platform on which the bulletin has to be installed		32 V
VENDOR	Vendor		80

			V
CVENAME	CVE Name Common Vulnerability Exposure ID for bulletin		255 V
ALTIDS	Alternate IDs		255 V
ENABLED	Bulletin Enabled [Y/N] Set to Y if the bulletin has to be managed, and N otherwise.	Y	1 V
XMLMD5	MD5 of Bulletin XML File MD5 checksum value for the bulletin XML metadata file		32 V
ROLLBACK	Bulletin Rollback [Y/N] Set to Y if the bulletin can be uninstalled, and N otherwise		1 V
REVISION	Revision Number Revision number assigned by the vendor of the bulletin		32 V
RUNMODE	Install Mode Bulletin installation mode	AUTOMATIC	48 V
SYNC	Patch Connection	PATCH.&(ID)_ *	80 C
SYNC	Location Connection Connection to the LOCATION class instances.	LOCATION. &(ID)_*	80 C
SYNC	Patch Signature Connection to the PATCH class instances.	PATCHSIG. &(ID)_*	80 C
PSLPOSTED	Date when Persistent Systems Posted the Bulletin		32 V
PSLREVISED	Date when Persistent Systems Revised the Bulletin		32 V
ALLDATA	Is there Data for all Patches	N	1 V
RATING	Bulletin Rating Vendor rating for the bulletin		255 V

HTTP Codes (CODES) Class

File:	PRIMARY	Description:	CODES contains the various HTTP/FTP status codes.
Domain:	PATCHMGR		
Class:	CODES		
Icon:		Can connect to:	

CODES Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
CODE	Code HTTP Status Code.		32 V
TYPE	Type of Code It can be HTTP or FTP.		12 V
NAME	Friendly Name		32 V
DESCRIPT	Description		255 V

Client Method (CMETHOD) Class

File:	PRIMARY	Description:	CMETHOD contains the various client methods used managing the bulletin on the target machine.
Domain:	PATCHMGR		
Class:	CMETHOD		
Icon:		Can connect to:	

CMETHOD Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ZCREATE	Create Method Sentient-supplied method or fully qualified method. This is predefined and should not be changed.		150 V
ZDELETE	Delete Method		150

	Sentient-supplied method or fully qualified method. This is predefined and should not be change.		V
ZVERIFY	Verify Method Sentient-supplied method or fully qualified method. This is predefined and should not be change.		150 V
ZUPDATE	Update Method Sentient-supplied method or fully qualified method. This is predefined and should not be change.		150 V
ZREPAIR	Repair Method Sentient-supplied method or fully qualified method. This is predefined and should not be change.		150 V

Dialog Services (DIALOG) Class

File:	PRIMARY	Description:	DIALOG class contains the connection to the REXX packages in the PACKAGE.
Domain:	PATCHMGR		
Class:	DIALOG		
Icon:		Can connect to:	

DIALOG Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ZSTOP000	Expression Resolution Method		100 E
ZSTOP001	Expression Resolution Method - 001		100 E
ZSTOP002	Expression Resolution Method - 002		100 E
ZSTOP999	Stop Unless RCA Connect		50 E
ZSVCNAME	Service Name/Description	&(NAME)	50 V
ALWAYS	RCS Method		80 M
ZSVCTTYP	Application Target Type [A/S]		2 V
ZSVCMO	Mandatory or Optional Service	O	2

	[M/O]		V
ZSVCCSTA	Service Status on Client (999)	999	3 V
ZSVCPRI	Service Create Ordering [01-99]		2 V
ZCREATE	Service Installation Method		100 M
ZINIT	Service Initialization Method		100 M
ZDELETE	Service Delete Method		100 M
ZUPDATE	Service Update Method		100 M
ZVERIFY	Service Verify Method		100 M
ZREPAIR	Service Repair Method		100 M
ZAVIS	Available, Verified, Installed, Sync F	YXNX	4 V
PUBDATE	Published Date of Service		32 V
VERDATE	Verified Date of Service		32 V
UPGDATE	When Application was Upgraded on De		32 V
UPDDATE	Upgrade Date (Programmatic)		32 V
INSTDATE	Installed Date		32 V
DELDATE	Delete Date		32 V
AUTHOR	Author Name		50 V
DESCRIPT	Application Description		100 V
VENDOR	Vendor Name		24 V

URL	WEB URL Name		100 V
SIZE	Application Size - Uncompressed		12 V
COMPSIZE	Application Size - Compressed		12 V
PRICE	Price		10 V
SCHEDOK	Update Schedule Locally [Y/N]		1 V
VERSION	Version Description		50 V
NAME	Friendly Name	<Defaults>	80 V
OWNER	Application Contact		50 V
RUNDLG	Dialog Processing [Y/N]	N	1 V
REBOOT	Install/Update/Delete/Version Change		64 V
EVENTS	Events to Report	AI=B,AD=B,AU=B,AR=B, AV=F,VA=B,VD=B	50 V
ERTYPE	Event Reporting Method [O/E/X]	O	1 V
ADAPTIVE	Auto Adaptability [Y/N]		1 V
LREPAIR	Local Repair [Y/N]		1 V
REMOVAL	Un-Managed Behavior [A/D/U]	D	1 V
RECONFIG	Reconfiguration Enabled [Y/N]		1 V
ZSVCCAT	Service Visible in Catalog? [Y/N]		1 V
UIOPTION	Progress Indicator [NONE/FULL/INFO]	INFO	4 V
CACHE	App Element Caching [Y/N]	N	1

			V
CACHELOC	CACHE Location On Client	_UNDEF_	255 V
CACHELIM	Percent Disk Limit For Cache	000	3 V
ZDISCONN	Disconnect on Install [Y/N]	Y	1 V
ZSYSACCT	Install under System Account [Y/N/M]	Y	1 V
MCELIBL	Service Multicast Eligible[Y/N]	Y	1 V
RSTRSIZE	Download restart threshold (bytes)	0	11 V
CATGROUP	Catalog Group Name		30 V
ZSVCMODE	Application Context [M/U/MU]		2 V
ZSVCGRP	Service Group Dependency [Y/N/D/]		1 V
ZSVCTYPE	Service Type [I/M/D]	M	1 V
PRELOAD	Preload Connection		80 V

DOMVER (DOMVER) Class

File:	PRIMARY	Description:	The DOMVER class contains the Domain Version for the current PATCHMGR domain.
Domain:	PATCHMGR		
Class:	DOMVER		
Icon:		Can connect to:	

DOMVER Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
DOMVER	Domain Version		8

	Version of the installed Sentient Patch Manager.		V

File Change (FILECHG) Class

File:	PRIMARY	Description:	The FILECHG class contains the file probe information for the MSSecure data feed.
Domain:	PATCHMGR		
Class:	FILECHG		
Icon:		Can connect to:	

FILECHG Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
PATH	File Path Path to the file to be verified.		255 V
NAME	File Name		255 V
VERSION	Internal File Version Version of the file to be verified for the vulnerability detection.		32 V
GMTDATE	GMT Date File date in GMT.		8 V
GMTTIME	GMT Time File time in GMT.		8 V
CHECKSUM	Checksum of File Content		12 V
CRC32	CRC32 of File Content		12 V
SIZE	File Size		12 V
ZOBJPART	Parent Instance Parent Instance ID to group the FILECHG		128 V
REPORT	Report Threshold Report the status if the return code is beyond the threshold		5 V
DSTATE	Desired State The expected desired state of the component		32 V

USE	Variables to Use The list of variables to be used when evaluating the logical expression to determine the state of the file		50 V
OPTIONS	Standard Options The Class connection to the OPTIONS class instance	OPTIONS.F_ NEW_WARN	80 A

File Resources (FILE) Class

File:	PRIMARY	Description:	An instance of the FILE Class defines a file, or a group of files, to include in an package. An instance of this class lets you specify such characteristics about the file as the file's name, size, and date stamp.
Domain:	PATCHMGR		
Class:	FILE		
Icon:		Can connect to:	

FILE Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ZRSCNAME	Resource Name	&ZRSCCFIL	30 V
ZRSCCFIL	Resource File Name		255 V
ZRSCM	Mandatory/Optional on Client [M/O] Flag set to M if the file has to be mandatory on the target device. Otherwise, it is set to O.	M	1 V
ZRSCVRFY	Verify Source File on Connect	U	8 V
ZRSCCONF	Confirm File Download [Y/N]	Y	1 V
ZRSCCRASH	DOS File Attribute [R/A/S/H] File attribute to be set, R - Read, A- Archive, S - System, H - Hidden.		4 V
ZRSCSTYP	Server File Type [BINARY/TEXT] Type of the file.	BINARY	8 V
ZRSCMMEM	PDS Member Name		8 V
ZRSCDATE	Resource Date Stamp - From Promote		8

			V
ZRSCTIME	Resource Time Stamp - From Promote		8 V
ZRSCSIZE	Resource Size - From Promote		11 V
ZRSCVERS	Resource Version - From Promote		12 V
ZRSCCSTA	Client File Status	999	3 V
ZRSCCKPT	Resource Checkpoint		1 V
ZRSCCRC	Resource CRC		8 V
ZRSCRSTR	Restart [Y/N]	Y	3 V
ZRSCOBJN	Persistent Object Name		8 V
ZCMPSIZE	Compressed File Size		11 V
ZCMPKEY	Compression Key	_UNDEF_	80 V
ZRSCSVRB	Version Skip Rebuild Flag [Y/N]		1 V
ZRSCPRI	Create Sequence Priority	50	2 V
ZRSCPADM	Admin ID		30 V
ZRSCSRC	Resource Source, i.e. Publisher		20 V
ZINIT	Resource Initialization Method		50 M
ZCREATE	Method to Install Resource		50 M
ZDELETE	Method to De-install Resource	RADREMF	50 M
ZREPAIR	Client Instance Repair Method		50 M

ZUPDATE	Client Instance Update Method		50 M
ZFILEUPD	Client File Update/Add Method		50 M
ZOPENERR	Client Method on File Open Error	RADLKM	50 M
ALWAYS	Connect To		50 C
NAME	Friendly Name		80 V
LOCATION	Location		255 V
ZCONTEXT	Component Context		8 V
FVERSION	File Version		30 V
PVERSION	Promote Version		30 V
ZPUBVER	Publisher Version		8 V
OBJGUID	Instance GUID		32 V
ZOBJPTCH	Perform Patching [Y/N]		1 V
ZRSCSIG	File Signature		32 V
SIGTYPE	File Signature Type		3 V

HP-US Fileset (HPFSET) Class

File:	PRIMARY	Description:	HPFSET contains the file set and version information for HPUX bulletins.
Domain:	PATCHMGR		
Class:	HPFSET		
Icon:		Can connect to:	

HPFSET Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
NAME	Fileset Name		255 V
VERSION	Fileset Version		255 V
ZOJBPART	Parent Instance Parent Instance ID to group the HPFSET.		128 V

Location (LOCATION) Class

File:	PRIMARY	Description:	The instance of the LOCATION class defines the location for every patch instance.
Domain:	PATCHMGR		
Class:	LOCATION		
Icon:		Can connect to:	

LOCATION Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ZSTOP000	Stop Expression		255 E
ZSTOP001	Stop Expression		255 E
URL	URL Location		25 V
ID	Unique Identifier		12 V
STATUS	Status of Primary URL	999	8 V
DATE	GMT Date/Time		18 V
SIZE	Patch Size		12 V
CRC	CRC of Data		12 V

LOCATION	Installation Directory	&(ZMASTER.ZROOTDRV)& (ZMASTER.ZROOTDIR) PATCHES\& (BULLETIN.NAME)	255 V
ZRSCCFIL	Patch File Name		128 V
ZRSCDATE	Resource Date		8 V
ZRSCTIME	Resource Time		8 V
ZRSCVRFY	Resource Options		8 V
ZRSCSIZE	Resource Size		12 V
ZRSCCRC	Resource CRC		8 V
ZRSCSIG	Resource Signature		32 V
SIGTYPE	Signature Type		5 V
ZCMPsize	Compressed Size		12 V
ZRSCSTYP	Resource Type [BINARY/TEXT]	BINARY	8 V
ZOBJMO	Mandatory/Optional on Client [M/O]		1 V
ZRSCRASH	DOS File Attributes [R/A/S/H]		4 V
ZRSCRSTR	Restart [Y/N]		1 V
ZOBJPRI	Create Sequence Priority		2 V
CTIME	Data Creation Time		24 V
MTIME	Data Modification Time		24 V
ZRSCRSTR	Check Point Restart	Y	1 V

ZCREATE	Patch extraction command	50	255 V
ZUPDATE	Patch extraction command		255 V
ZDELETE	Patch delete method	RADREMF	50 M

Message Catalogs (MSGCAT) Class

File:	PRIMARY	Description:	MSGCAT class contains the file instance of Message Catalog files.
Domain:	PATCHMGR		
Class:	MSGCAT		
Icon:		Can connect to:	

MSGCAT Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ZRSCNAME	Resource Name		30 V
ZRSCCFIL	Resource File Name		255 V
ZRSCMO	Mandatory/Optional on Client [M/O]	M	1 V
ZRSCVRFY	Verify Resource File on Connect	U	8 V
ZRSCCONF	Confirm File Download [Y/N]	Y	1 V
ZRSCRASH	DOS File Attribute [R/A/S/H]		4 V
ZRSCSTYP	Server File Type [BINARY/TEXT]	BINARY	8 V
ZRSCMMEM	PDS Member Name		8 V
ZRSCDATE	Resource Date Stamp - From Promote		8 V
ZRSCTIME	Resource Time Stamp - From Promote		8

			V
ZRSCSIZE	Resource Size - From Promote		11 V
ZRSCVERS	Resource Version - From Promote		12 V
ZRSCCSTA	Client File Status	999	3 V
ZRSCCKPT	Resource Checkpoint		1 V
ZRSCCRC	Resource CRC		8 V
ZRSCSIG	Resource Signature		32 V
ZRSCRSTR	Restart [Y/N]	Y	3 V
ZRSCOBJN	Persistent Object Name		8 V
ZCMPSIZE	Compressed File Size		11 V
ZCMPKEY	Compression Key		80 V
ZRSCSVRB	Version Skip Rebuild Flag [Y/N]		1 V
ZRSCPRI	Create Sequence Priority	50	2 V
ZRSCPADM	Admin ID		30 V
ZRSCSRC	Resource Source, i.e. Publisher		20 V
ZINIT	Resource Initialization Method		50 M
ZCREATE	Method to Install Resource		50 M
ZDELETE	Method to De-install Resource	RADREMF	50 M
ZREPAIR	Client Instance Repair Method		50 M

ZUPDATE	Client Instance Update Method		50 M
ZFILEUPD	Client File Update/Add Method		50 M
ZOPENERR	Client Method on File Open Error	RADLKM	50 M
ALWAYS	Connect To		50 C
NAME	Friendly Name		80 V
LOCATION	Location		255 V
ZCONTEXT	Component Context		8 V
FVERSION	File Version		30 V
PVERSION	Promote Version		30 V
ZPUBVER	Publisher Version		8 V
OBJGUID	Instance GUID		32 V
ZOBJPTCH	Perform Patching [Y/N]	Y	1 V
SIGTYPE	File Signature Type		3 V

Metadata Resources (METADATA) Class

File:	PRIMARY	Description:	Metadata Class contains the patch manager metadata files.
Domain:	PATCHMGR		
Class:	METADATA		
Icon:		Can connect to:	

METADATA Class Attribute Reference

Attribute Name	Explanation	Value of Base	Length

		Instance	Type
ZRSCNAME	Resource Name	&ZRSCCFIL	30 V
ZRSCCFIL	Resource File Name		255 V
ZRSCMO	Mandatory/Optional on Client [M/O]		1 V
ZRSCVRFY	Verify Resource File on Connect		8 V
ZRSCCONF	Confirm File Download [Y/N]		1 V
ZRSCRASH	DOS File Attribute [R/A/S/H]		4 V
ZRSCSTYP	Server File Type [BINARY/TEXT]	BINARY	8 V
ZRSCMMEM	PDS Member Name		8 V
ZRSCDATE	Resource Date Stamp - From Promote		8 V
ZRSCTIME	Resource Time Stamp - From Promote		8 V
ZRSCSIZE	Resource Size - From Promote		11 V
ZRSCVERS	Resource Version - From Promote		12 V
ZRSCCSTA	Client File Status	999	3 V
ZRSCCKPT	Resource Checkpoint		1 V
ZRSCCRC	Resource CRC		8 V
ZRSCSIG	Resource Signature		32 V
ZRSCRSTR	Restart [Y/N]	Y	3 V
ZRSCOBJN	Persistent Object Name		8 V

ZCMPSIZE	Compressed File Size		11 V
ZCMPKEY	Compression Key		80 V
ZRSCSVRB	Version Skip Rebuild Flag [Y/N]		1 V
ZRSCPRI	Create Sequence Priority	50	2 V
ZRSCPADM	Admin ID		30 V
ZRSCSRC	Resource Source, i.e. Publisher		20 V
ZINIT	Resource Initialization Method		50 M
ZCREATE	Method to Install Resource		50 M
ZDELETE	Method to De-install Resource	RADREMF	50 M
ZREPAIR	Client Instance Repair Method		50 M
ZUPDATE	Client Instance Update Method		50 M
ZFILEUPD	Client File Update/Add Method		50 M
ZOPENERR	Client Method on File Open Error	RADLKM	50 M
ALWAYS	Connect To		50 C
NAME	Friendly Name	Default	80 V
LOCATION	Location		255 V
ZCONTEXT	Component Context		8 V
FVERSION	File Version	_UNDEF_	30 V
PVERSION	Promote Version	_UNDEF_	30

			V
ZPUBVER	Publisher Version	_UNDEF_	8 V
OBJGUID	Instance GUID	_UNDEF_	32 V
ZOBJPTCH	Perform Patching [Y/N]	Y	1 V
SIGTYPE	File Signature Type		3 V

Options (OPTIONS) Class

File:	PRIMARY	Description:	OPTIONS class contains the Report Threshold Desired.
Domain:	PATCHMGR		
Class:	OPTIONS		
Icon:		Can connect to:	

OPTIONS Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
REPORT	Report Threshold	1	2 V
DSTATE	Desired State Options		32 V
USE	Variables to use in compression		64 V

Patch (PATCH) Class

File:	PRIMARY	Description:	PATCH class contains the patch instance for each bulletin. This contains all the necessary information related to that patch.
Domain:	PATCHMGR		
Class:	PATCH		
Icon:		Can connect to:	

PATCH Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
LANG	ISO 639 Alpha(2) code2		5 V
ARCH	Architecture		8 V
PLATFORM	Platform [winnt/win9x]		16 V
OSVER	OS Version String		12 V
OSTYPE	Type [server/workstation]		16 V
OSSUITE	Suite [Datacenter/Blade]		16 V
LOCATION	Location Instance		32 V
PATCHSIG	Patch Signature Instance		32 V
RELEASE	Release Instance Name		32 V
NAME	Patch Name		80 V
REBOOT	Reboot [Y/N]		1 V
BULLETIN	Bulletin Name		32 V
QNUMBER	Q Number		32 V
ID	Unique Identifier		12 V
SUPERCED	Patch Superceded [Y/N]	N	1 V
SUSNAME	SUS Patch Name		80 V
MSSNAME	MS SECURE NAME		80 V
SUPERBU	Superceded bulletin name		32

			V
SUPERPA	Superceded Patch Instance Name		32 V
SUPERMSS	Superceded MS SECURE NAME		80 V
DATA	Patch Data file exists [Y/N]		1 V
PATCHURL	Patch URL		255 V
PATCHFN	Patch File Name		255 V
FORCEDL	Download on each acquisition [Y/N]	N	1 V
CTIME	Patch Creation Time		24 V
MTIME	Patch Modified Time		24 V
FIXEDIN	Fixed in Release		32 V
ROLLBACK	Patch Rollback [Y/N/]		1 V
RUNMODE	Install Mode	AUTOMATIC	48 V
MANAGE	Patch Signature	PATCHSIG.& (PATCHSIG)	80 C
OPTIONS	Options connection	OPTIONS.PATCH	80 A
ZSTOP000	Stop if not enabled		128 E
ZSTOP001	Stop if superceded		128 E
ZSTOP002	Stop if not managing		128 E
SYNC	Release Connections	RELEASE.&(RELEASE)	80 C
PATCHARG	Patch Arguments Instance		80 A

OTYPE	Object Type		80 V
OCREATE	Create Arguments	_NULL_	255 V
ODELETE	Delete Arguments	_NULL_	255 V
OVERIFY	Verification arguments		255 V
OASSERT	Assert arguments		255 V
OUPDATE	Update arguments		255 V
ENABLED	Patch Enabled [Y/N]	Y	1 V
INSORDER	Install Order		3 V
DELORDER	Delete Order		3 V
CPU	Number of CPU's [uni/multi]		8 V
PATCHNAM	Patch File Name from archive		255 V
REBOOT_U	Uninstall On Reboot		10 V
RATING	Patch Rating		255 V
TITLE	Patch Title		255 V

Patch Arguments (PATCHARG) Class

File:	PRIMARY	Description:	PATCHARG contains the patch arguments required for the different operating systems.
Domain:	PATCHMGR		
Class:	PATCHARG		
Icon:		Can connect to:	

PATCHARG Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
OCREATE	Patch Command Line	/z /q	255 V
ODELETE	Uninstall Command Line	/q/ /z	255 V

Patch Auth. Store (PASTORE) Class

File:	PRIMARY	Description:	PASTORE contains the Patch Status being acquired and published to the CSDB.
Domain:	PATCHMGR		
Class:	PASTORE		
Icon:		Can connect to:	

PASTORE Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
NAME	Friendly Name		80 V
ID	Unique Identifier		12 V
DVERSION	Data Version		16 V
LDUPDATE	Last Data Update Date		16 V
SVERSION	Schema Version		16 V
LSUPDATE	Last Schema Update Date		16 V
TVERSION	Tool Version		16 V
BULLTNUM	Bulletin Count		8 V
ZRSCCFIL	Resource File		255 V
LOCATION	Location of Resource File		80

Reference GuideChapter 10: Configuration Server

			V
ZRSCSIZE	Resource Size		12 V
ZCMPsize	Compressed Size		12 V
ZGMTDATE	Resource Date		8 V
ZGMTTIME	Resource Time		8 V
ZRSCSIG	MD5 Signature		32 V
SIGTYPE	Signature Type		5 V
ZSTATUS	Status	999	3 V
SRCURL	Source URL		255 V
JOBSTART	Start time of publishing job		32 V
JOBEND	End time of publishing job		32 V
MACHINE	Publisher Machine		80 V
AUTHOR	Author		32 V
PATCHNUM	Total Number of Patches in Data Store		8 V
PATCHERR	Number of Patches with Errors		8 V
ZRSCCRC	Resource CRC		8 V
ZRSCSTYP	Resource Type	BINARY	8 V
RVENDORS	Requested Vendor List		255 V
AVENDORS	Acquired Vendor List		255 V

ERRORS	List of Errors	PATCHMGR.PUBERROR.&(ID)_*	80 A

Patch Signature (PATCHSIG) Class

File:	PRIMARY	Description:	PATCHSIG contains the class connection to FILECHG and REGCHG.
Domain:	PATCHMGR		
Class:	PATCHSIG		
Icon:		Can connect to:	

PATCHSIG Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ZSTOP001	Stop if not needed	(EDMGETV(ZCURRENT,ZCURMSG) = 'MANAGE' & ((EDMGETV(PATCH,ZOBJRC) = '0') (EDMGETV(PATCH,ZOBJRC) = '2') (EDMGETV(PATCH,ZOBJRC)='999')))	255 E
NAME	Package Name		32 V
ID	Unique Identifier		12 V
CRC	CRC of FILECHGS and REGCHGS		8 V
CTIME	Creation Time		24 V
MTIME	Modification Time		24 V
FILECHGS	File Connection	FILECHG.&(ID)_*	80 A
REGCHGS	Registry Connection	REGCHG.&(ID)_*	80 A
HPFSETS	HP Fileset Connection	HPFSET.&(ID)_*	80 A

Path (PATH) Class

File:	PRIMARY	Description:	PATH class stores the drive and directory required to install a resource.
Domain:	PATCHMGR		
Class:	PATUH		
Icon:		Can connect to:	

PATH Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
DRIVE	Drive		20 V
DIR	Directory		255 V
NAME	Friendly Name		255 V
ZRSCPRI	Priority	5	2 V
ZRSCMO	Mandatory/Optional [M/O]	M	1 V
OBJGUID	Instance GUID	_UNDEF_	32 V

Product (PRODUCT) Class

File:	PRIMARY	Description:	PRODUCT Class contains the information on various products supported.
Domain:	PATCHMGR		
Class:	PRODUCT		
Icon:		Can connect to:	

PRODUCT Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
NAME	Friendly Name		255

			V
ID	Unique Identifier		12 V
TAG	Normalized Name		255 V
CVERSION	Current Product Version		32 V
RSUPPORT	Minimum Supported Release		255 V
RCURRENT	Current Release		255 V
CTIME	Creation Time		24 V
MTIME	Modification Time		24 V
RELEASE	Release Connection	RELEASE.&(ID)_*	80 C
OPTIONS	Options Connection	OPTIONS.PRODUCT	80 A
PATCHARG	Patch Arguments Instance		80 A
OTYPE	Object Type		80 V
OVERIFY	Verify Arguments		255 V
VENDOR	Vendor [MICROSOFT/REDHAT/SUSE/HPUX]		80 V
SYNC	Connection PG to PR	PG2PR.*_(ID)	80 C

Product Association (PG2PR) Class

File:	PRIMARY	Description:	PG2PR contains the class connection to the PRODUCT and RELEAE classes.
Domain:	PATCHMGR		
Class:	PG2PR		
Icon:		Can connect to:	

PG2PR Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
PROGROUP	Product Group Instance		12 V
PRODUCT	Product Instance		12 V
MEMBER	Product Connection	PRODUCT.&(PRODUCT)	80 A
SYNC	Product Group Connection	PROGROUP.&(PROGROUP)	80 C

Publisher Error (PUBERROR) Class

File:	PRIMARY	Description:	PUBERROR contains the errors if any during the publishing of the patches.
Domain:	PATCHMGR		
Class:	PUBERROR		
Icon:		Can connect to:	

PUBERROR Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
URL	Patch URL		255 V
PATCHNAM	Patch Name		80 V
BULLETIN	Bulletin		16 V
ERROR	Error Code		3 V
ERRORMSG	Error Message		255 V
ID	Job ID		32 V
LOCATION	Location Instance Name		32 V

VENDOR	Vendor Name		80 V

Registry Resources (REGCHG) Class

File:	PRIMARY	Description:	REGCHG class contains the registry related information required for the patch metadata..
Domain:	PATCHMGR		
Class:	REGCHG		
Icon:		Can connect to:	

REGCHG Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
PATH	Registry Key Name		255 V
VALUE	Value Data		255 V
NAME	Value Name		255 V
TYPE	Value Type[sz/dword]		12 V
ZOBJPART	Parent Instance Name		128 V
REPORT	Report Threshold		5 V
DSTATE	Desired State		32 V
USE	Variables to Use		50 V
OPTIONS	Options Connection	OPTIONS.R_EXACT	80 A

Release (RELEASE) Class

File:	PRIMARY	Description:	RELEASE contains the different releases of the PRODUCT, which patch manager supports.
Domain:	PATCHMGR		
Class:	RELEASE		
Icon:		Can connect to:	

RELEASE Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
PRODUCT	Product Instance Name		32 V
SP	SP Instance Name		32 V
ID	Unique Identifier		12 V
PROBE	Probe Name		32 V
ARGS	Probe Arguments		128 V
NAME	Release Name		255 V
CTIME	Creation Time		24 V
MTIME	Modification Time		24 V
PARENTC	Parent Class	PRODUCT	8 V
PARENTI	Parent Instance	&(PRODUCT)	32 V
CHILDC	Child Class	SP	8 V
CHILDI	Child Instance	&(SP)	32 V
PRODUCT	Product Connection	PRODUCT.&(PRODUCT)	80

			C
SP	SP Connection	SP.&(SP)	80 C
RELEASE	Children Release Instances	RELEASE.&(ID)_*	80 C
DISC_PA	Patch Connection	PATCH.*_&(ID)_*	80 C
OPTIONS	Options	OPTIONS.RELEASE	80 A
SYNC	Sync Connection to Fetch Product	PRODUCT.&(PRODUCT)	80 C
SYNC	Sync Connection to Fetch SP	SP.&(SP)	80 C

Services Packs (SP) Class

File:	PRIMARY	Description:	SP Class contains the Service Pack details for the product.
Domain:	PATCHMGR		
Class:	SP		
Icon:		Can connect to:	

SP Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
NAME	SP Name		255 V
ID	Unique Identifier		12 V
TAG	Normalized Name		255 V
POSTED	SP Post Date		18 V
REVISED	SP Revised Date		18 V
SUPPORT	Supported [Y/N]		1 V

URL	SP URL		255 V
FAQURL	FAQ URL		255 V
IMPACT	Impact ID		8 V
PREREQ	Prereq ID		8 V
MITIGATE	Mitigate ID		8 V
POPULAR	Popularity ID		8 V
CTIME	Creation Time		24 V
MTIME	Modification Time		24 V
OTYPE	Object Type		80 V
OASSERT	Assert Arguments		255 V
OVERIFY	Verify Arguments		255 V

Services (ZSERVICE) Class

File:	PRIMARY	Description:	ZSERVICE class contains the Bulletin, patches and any required files to define the Bulletin by the vendor.
Domain:	PATCHMGR		
Class:	ZSERVICE		
Icon:		Can connect to:	

ZSERVICE Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ZSTOP000	Expression Resolution Method		100 E
ZSTOP001	Expression Resolution		100

	Method - 001		E
ZSTOP002	Expression Resolution Method - 002		100 E
ZSTOP999	Stop Unless RCA Connect	\(UPPER(EDMGETV(ZMASTER, ZDOMNAME))='PATCH' POS ('PATCHMGR',EDMGETV (PATCHOBJ,FDCI))>0)	50 E
SYNC	Sync Connection	BULLETIN.&(ZOBJNAME)(SYNC)	80 C
ZSVCNAM E	Service Name/Description	&(NAME)	50 V
ALWAYS	RCS Method		80 M
ZSVCTTYP	Application Target Type [A/S]		2 V
ZSVCMO	Mandatory or Optional Service [M/O]	O	2 V
ZSVCCSTA	Service Status on Client (999)	999	3 V
ZSVCPRI	Service Create Ordering [01-99]		2 V
ALWAYS	Contains	BULLETIN.&(ZOBJNAME)(MANAGE)	50 A
ZCREATE	Service Installation Method		100 M
ZINIT	Service Initialization Method		100 M
ZDELETE	Service Delete Method		100 M
ZUPDATE	Service Update Method		100 M
ZVERIFY	Service Verify Method		100 M
ZREPAIR	Service Repair Method		100 M
ZAVIS	Available, Verified, Installed, Sync F	YXNX	4 V

Reference GuideChapter 10: Configuration Server

PUBDATE	Published Date of Service		32 V
VERDATE	Verified Date of Service		32 V
UPGDATE	When Application was Upgraded on De		32 V
UPDDATE	Upgrade Date (Programmatic)		32 V
INSTDATE	Installed Date		32 V
DELDATE	Delete Date		32 V
AUTHOR	Author Name		50 V
DESCRIPT	Application Description		100 V
VENDOR	Vendor Name		24 V
URL	WEB URL Name		100 V
SIZE	Application Size - Uncompressed		12 V
COMPSIZE	Application Size - Compressed		12 V
PRICE	Price		10 V
SCHEDOK	Update Schedule Locally [Y/N]		1 V
VERSION	Version Description		50 V
NAME	Friendly Name	<Defaults>	80 V
OWNER	Application Contact		50 V
RUNDLG	Dialog Processing [Y/N]	Y	1 V
REBOOT	Install/Update/Delete/Versio		64

Reference Guide

Chapter 10: Configuration Server

	n Change		V
EVENTS	Events to Report	AI=B,AD=B,AU=B,AR=B,AV=F,VA=B,VD=B	50 V
ERTYPE	Event Reporting Method [O/E/X]	O	1 V
ADAPTIVE	Auto Adaptability [Y/N]		1 V
LREPAIR	Local Repair [Y/N]		1 V
REMOVAL	Un-Managed Behavior [A/D/U]	O	1 V
RECONFIG	Reconfiguration Enabled [Y/N]		1 V
ZSVCCAT	Service Visible in Catalog? [Y/N]	N	1 V
UIOPTION	Progress Indicator [NONE/FULL/INFO]	INFO	4 V
CACHE	App Element Caching [Y/N]	N	1 V
CACHELOC	CACHE Location On Client		255 V
CACHELIM	Percent Disk Limit For Cache	000	3 V
ZDISCONN	Disconnect on Install [Y/N]	Y	1 V
ZSYSACCT	Install under System Account[Y/N/M]	Y	1 V
MCELIBL	Service Multicast Eligible [Y/N]	Y	1 V
RSTRSIZE	Download restart threshold (bytes)	0	11 V
CATGROUP	Catalog Group Name		30 V
ZSVCMODE	Application Context [M/U/MU]		2 V
ZSVCGRP	Service Group Dependency [Y/N/D/]		1 V

ZSVCTYPE	Service Type [I/M/D]	M	1 V
RUNMODE	Install Mode	AUTOMATIC	48 V
PRELOAD	Preload Connection	BULLETIN.&(ZOBJNAME) (PRELOAD)	80 CV

The POLICY Domain

Use the POLICY Domain to create users and groups, and to assign users to groups. For more information, see *Sentient Enterprise Console Online Help*.

Country / Region (COUNTRY) Class

File:	PRIMARY	Description:	Country / Region. Use an instance of the COUNTRY Class to define the time zone, date format, clock synchronization with the Configuration Server, and language for an agent computer. Do not assign services to this class.
Domain:	POLICY		
Class:	COUNTRY		
Icon:		Can connect to:	

COUNTRY Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ALWAYS	Utility Resolution Method: Any method you specify for this attribute is unconditionally executed by Client Automation when this object is resolved. A valid method name, for example, ZSYSTEM.ZMETHOD.PUT_ZERROR_HIST.		50 M
DATEFORM	Date Format: This attribute sets up the date format display. A valid date structure (for example, MM/DD/YY or DD/MM/YY).	MM/DD/YY	16 V
LANGUAGE	Language: This informational attribute informs other programs the language the client is using. Input the language that your enterprise uses while running Client Automation.	ENGLISH	16 V
NAME	Friendly Name		80 V
TIMEZONE	Time Zone: Informational attribute that lets you specify a time zone for an agent, e.g., EST, CST,	EST	8 V

	PST, and so forth.		
ZTIMEDIF	Time Zone Offset (in minutes): If you do not synchronize the agent to your site's server clock, you must enter an offset value for this attribute. Sentient time is set on Eastern Standard Time (EST). Thus, if your site's time zone is Central Standard Time (CST), enter 60 for this attribute. If your site's time zone is Pacific Standard Time (PST), enter 180 for this attribute, and so forth. Numeric value between 1 and 9999.		4 V

Departments (DEPT) Class

File:	PRIMARY	Description:	Use an instance of the DEPT Class to group subscribers into departments.
Domain:	POLICY		
Class:	DEPT		
Icon:		Can connect to:	ZSERVICE, COUNTRY, STAGER, HTTP, MULTICAST

DEPT Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ALWAYS	Offers: Class connections determine the path of resolution of an agent's desired state. An _ALWAYS_ class connection is always resolved.		50 C
ALWAYS	Utility Resolution Method: Any method you specify for this attribute is unconditionally executed by Sentient when this instance is resolved. A valid method name, for example, ZSYSTEM.ZMETHOD.PUT_ZERROR_HIST.		51 M
ACCTNO	Account Number: Attribute that lets you specify the account number for the department. Site-specific account number.		12 V
NAME	Friendly Name		80 V

Machine Manufacturer (MANUFACT) Class

File:	PRIMARY	Description:	Used to assign policy based on the machine manufacturer.
Domain:	POLICY		
Class:	MANUFACT		
Icon:		Can connect to:	

MANUFACT Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ALWAYS	Utility Method		U
ALWAYS	Service Connection		C
NAME	Friendly Name		V

Machine Models (MODEL) Class

File:	PRIMARY	Description:	Used to assign policy based on the machine model.
Domain:	POLICY		
Class:	MODEL		
Icon:		Can connect to:	

MODEL Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ALWAYS	Utility Method		U
ALWAYS	Service Connection		C
NAME	Friendly Name		V

Machine Roles (ROLE) Class

File:	PRIMARY	Description:	Used to assign policy based on the machine role.
Domain:	POLICY		
Class:	ROLE		
Icon:		Can connect to:	

ROLE Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ALWAYS	Utility Method		U
ALWAYS	Service Connection		C
NAME	Friendly Name		V

Machine Subnets (SUBNET) Class

File:	PRIMARY	Description:	Used to assign policy based on the machine subnet.
Domain:	POLICY		
Class:	SUBNET		
Icon:		Can connect to:	

SUBNET Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ALWAYS	Utility Method		U
ALWAYS	Service Connection		C
NAME	Friendly Name		V

Mobile Device Config (MBLCONFIG) Class

File:	PRIMARY	Description:	This class defines the parameters for mobile device configuration when using the Mini Management Server.
Domain:	POLICY		
Class:	MBLCONFIG		

Icon:		Can connect to:	

MBLCONFIG Class

Attribute Name	Explanation	Value of Base Instance	Length Type
RPSADR1	RCA Proxy Server #1 [IP Addr:Port]: Specifies the first Proxy Server's IP address and port number, in the format of <IP Address>:<Port>. For example: 192.168.0.1:3466		255 V
RPSADR2	RCA Proxy Server #2 [IP Addr:Port]: Specifies the second Proxy Server's IP address and port number, in the format of <IP Address>:<Port>. For example: 192.168.0.2:3466		255 V
RMDSADR1	RCA Mobile Dist [IP Addr:Port]: Specifies the Mini Management Server's IP address and port number, in the format of <IP Address>:<Port>. For example: 192.168.0.3:3466		255 V
RIMADR	RCA Inventory Mgr [IP Addr:Port]: Specifies the Inventory Manager Server's IP address and port number, in the format of <IP Address>:<Port>. For example: 192.168.0.4:3466		255 V
RMPADR	RCA Mgmt Portal [IP Addr:Port]: Specifies the Portal's IP address and port number, in the format of <IP Address>:<Port>. For example: 192.168.0.5:3466		255 V
ZMBLSGNO	Mobile Signon Panel [Y/N]: Specifies whether to display the mobile device's sign on panel. Use Y to display the panel and N to suppress the panel.	N	1 V
SETUSER	Set Userid on Mobile Device [Y/N]: Set this to Y if you want the userID set to the serial number of the mobile device. Set this to N if you want this to be left as user entry.	N	1 V
STORUSER	Store Userid & Password [Y/N]: Set to Y if you want to store the User/Password in the UPDATE . INI file. Set this to N if you want to clear the User/Password in the UPDATE . INI file.	Y	1 V

Attribute Reference

Multicast (MULTICAST) Class

File:	PRIMARY	Description:	Use the MULTICAST Class to configure agent computers to use multicasting.
Domain:	POLICY		
Class:	MULTICAST		
Icon:		Can connect to:	

MULTICAST Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ALWAYS	Manager REXX Method A Configuration Server REXX method.	SYSTEM .ZMETHOD .MULTICAST	255 M
ADDRESS	Broadcast or Multicast Address Specifies a valid broadcast/multicast address. Additionally, this attribute is important to the dynamic multicast feature.	229.0.0.0	19 V
ALTADDRM	Multicast Server IP The IP address of Multicast Server for object exchange.		32 V
ALTPORTM	Multicast Server Port The port of Multicast Server for object exchange. Note: This is the TCP/IP port that the Multicast Server listens on, not the multicast transmission port. The default is 3463 . This port was chosen to avoid conflicts with known Client Automation ports (such as the Configuration Server port). Therefore, be prepared to adjust accordingly, as other network software might also conflict.	3463	5 V
BYPASCON	User to Prime Multicast Server Limits which clients send which data to the Multicast Server. BYPASCON = blank - All clients will exchange a control object and a needs list object with the Multicast Server. BYPASCON = User - The client with this user		255 V8

	ID will send <i>both</i> objects to the Multicast Server, and the other clients will send <i>only</i> the control object. BYPASCON = User 2 (a user ID, followed by a space and the numeral 2) - The client with this user ID will send <i>both</i> objects to the Multicast Server; no other clients will be part of the object exchange. Note: This option requires that the local time on the clock of the machine that houses the Multicast Server be in synchronization with all multicast-eligible clients.		
CGMTDATE	Collect Start Date (YYYYMMDD) The start date of the collection period. The format is YYYYMMDD. Additionally, this attribute is important to the dynamic multicast feature.		8 V
CGMTTIME	Collect Start Time GMT (HH:MM:SS) The start time of the collection period. The format is HH:MM:SS. Additionally, this attribute is important to the dynamic multicast feature.		8 V
CLASS	Class Name Specifies the class name.	&(ZOBJCLASS)	12 V
CWINDOW	Duration of Collection Window (min) Duration (in minutes) of collection phase for all clients to register their list of required files. Additionally, this attribute is discussed as it relates to the dynamic. Note: When specifying the duration of your collection window, be sure it is sufficient to collect all the files in the request list.		3 V
DELAYBP	Delay between Packets (msec) Delay (milliseconds) between packets.	40	5 V
DELAYFP	Delay after Packets (msec) Delay (msec) after the first packet is sent.	40	5 V
DOMAIN	Domain Name Specifies the domain name.	&(ZOBJDOMN)	32 V

INSTANCE	Instance Name Specifies the instance name.	&(ZOBJNAME)	32 V
MCAST	Enable Multicast [Y/N] A flag to indicate if multicast is enabled. The default is Y.	Y	1 V
MCORDER	Service Installation Order [B/A/S] Enables an administrator to specify when to download and install mandatory services that are not multicast eligible. B - Download the service and install it on the client <i>before</i> the multicast session. A - Download the service and install it on the client <i>after</i> the multicast session. S - SPLIT the process so that it downloads the service to the client <i>before</i> the multicast, and installs the service <i>after</i> the multicast session. This is the default setting.	S	1 V
MDELAY	Delay before Multicast Start (min) Delay between close of collection and start of multicast (minutes).		3 V
MINREF	Minimum Number of Clients Requesting File The minimum number of clients that must request a file in order for it to be considered for multicast.	1	7 V
MINSIZE	Minimum File Size for Multicast The minimum size a file must be in order to be considered for multicast	1024	10 V
MODE	Broadcast or Multicast [M/B] Broadcast or Multicast. The default is M. Note: If MODE = M, the ADDRESS variable should be specified in the standard Internet 'dotted-decimal' format, and should be between 225.0.0.0 and 239.255.255.255. (Multicast addresses are defined as IP class D addresses in this range. Avoid using the extreme low end of the range (224.0.0.1 – 224.255.255.255), as most of these are reserved for specific purposes.) The B option is for those network configurations (primarily satellite and single sub-area networks) that don't support IGMP.	M	1 V

MWINDOW	Duration of Multicast Window (min) Controls the amount of time (in minutes) for a client to run before terminating its multicast "receive" activity. The default is 0 , which allows a client to maintain the receive mode for as long as is necessary to receive all the requested data.	0	3 V
PORT	Broadcast or Multicast UDP Port Broadcast or Multicast UDP port.	9512	5 V
RESENDS	Number of Resends Number of re-sends. Note: Multicast is designed to re-send all packets if RESENDS > 0. It will send STORE number of packets and then re-send that group for RESENDS times.	1	5 V
STORE	Backup Packets to Store for Resend Number of packets to buffer for resends. Note: Multicast is designed to re-send all packets if RESENDS > 0. It will send STORE number of packets and then re-send that group for RESENDS times.	20	5 V
TTL	Number of Router Hops	3	3 V

PDACONFIG (PDACONFIG) Class

File:	PRIMARY	Description:	Defines the parameters for PDA configuration and is used to obtain inventory reports of cradled-PDAs
Domain:	POLICY		
Class:	PDACONFIG		
Icon:		Can connect to:	

PDACONFIG Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
MODE	Operation Mode	RCA	10 V
RIS_ADDR	Tcl Web Server Address		50

			V
RIS_PORT	Tcl Web Server Port	3466	4 V
SVC_NTIFY	Service Notify Flag [Y/N]	N	1 V

POLPRMS (POLPRMS) Class

File:	PRIMARY	Description:	Use an instance of this class to restrict the policy resolution for a single-service.
Domain:	POLICY		
Class:	POLPRMS		
Icon:		Can connect to:	None

USER Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
XDOMAIN	Single Service Resolution: The domain names that will be used for single-service policy resolution. For more information on how to implement single service policy resolution, see the section <i>"Optimization for Single-Service Policy Resolution" on page 1305</i> .	None	32 V

Server Stagers (STAGER) Class

File:	PRIMARY	Description:	Use to define Staging Servers or Proxy Servers within your distribution network. In this table staging server can refer to either a Proxy Server or Staging Server.
Domain:	POLICY		
Class:	STAGER		
Icon:		Can connect to:	

STAGER Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type

ZDATAURL	Sentient Proxy Server URL Address: This parameter is for use with the Proxy Server only. Use the Proxy Server if you would like to use http and dynamic caching.		80 V
ZNETNDIR	Alternative Install Directory: Specify a path to a folder on the agent computer's ZNETNDRV drive where the Staging Server's resources are located. The location on the server where staged data is stored is created by a concatenation of the values of the ZNETNDRV and ZNETNDIR attributes. Add a leading and trailing slash, for example, \STAGER\.		64 V
ZNETNDRV	Alternative Install Drive: Type a drive letter defined on the agent computer followed by a colon, for example, D: . This can be a local drive, such as a CD-ROM drive, or a mapped network drive.		19 V
ZPCPROTL	Stager Protocol [TCP/HTTP]: The protocol to be used for communications between the Staging Requestor and the Staging Server. Select one of the following: NETB – Net BIOS TCP – TCP/IP SPX – Novell NOS	TCP	4 V
ZSTGDESC	Stager Description: Specify a text string that subscribers will identify with the Staging Server.	RCA Stager	30 V
ZSTGENV	Stager Environment [S/M]: S if the Staging Server is on a machine separate from the Configuration Server. M if the Staging Server is co-resident with the Configuration Server. S = separate M = located on Configuration Server.		1 V
ZSTGLOCS	Type-Server/Network/Both [S/N/B] Specify S if the subscriber is to retrieve files from the Staging Server via a TCP/IP session. Specify N if the subscriber is to retrieve files from an addressable location on the agent computer such as a mapped network drive or a local CD-ROM drive. Specify B if the subscriber is to retrieve files from both. If both (B), first the addressable location on the agent computer is checked for the necessary resources. If the files are not found, then the Staging Server is requested to supply them.	S	4 V

	. This can be a local CD-ROM drive or a network share. The resources are unavailable from this source if the ZNETNDRV and ZNETNDIR are blank, the location specified is invalid, a CD-ROM drive is identified that has no CD-ROM loaded, or the files cannot be found on the CD-ROM. The location specified by ZSTGSDRV and ZSTGSDIR. This can be a dedicated Staging Server, or a Staging Server co-resident with a Configuration Server. The resources are unavailable from this source if the ZSTGSDRV and ZSTGSDIR are blank, or if the Staging Server cannot supply them from the designated location. 3. The Configuration Server.		
ZSTGOPT	Perform Staging [1/0] 1 = enable staging on the agent. 0 = disable staging on the agent.	1	1 V
ZSTGPORT	TCP/IP Stager Port: If the agent connects to the Staging Server via a TCP/IP connection, type the TCP/IP port through which the communication will take place. The default port is 3461. This is a well-known port that has been registered for this purpose. For non-TCP/IP connections, leave blank.		4 V
ZSTGSDIR	Stager Directory: Specify a path to a folder on the Staging Server's ZSTGSDRV where the Staging Server will store staged data. The location on the server where staged data is stored is created by a concatenation of the values of the ZSTGSRDV and ZSTGSDIR attributes. Specify a leading and trailing slash, e.g., \STAGER\, /opt/novadigm/Stager/, \nvdm\radstage\. If the specified folder does not exist, Client Automation creates it automatically when data is first copied to the Staging Server.		64 V
ZSTGSDRV	Stager Drive: Specify the drive on the Staging Server where Client Automation will store staged data. Specify the drive letter followed by a colon, e.g., C:, UNIX:, SYS:. Note: The drive letter is the local drive designation on the stager.		19 V
ZSTGSRVR	Stager Name or IP Address: If the agent connects to the Staging Server via a TCP/IP connection, type the IP address or host name of the Staging Server. Otherwise, type the Staging Server's network name.		50 V

Users (USER) Class

File:	PRIMARY	Description:	Use to define individual subscribers or agent computers.
Domain:	POLICY		
Class:	USER		
Icon:		Can connect to:	COUNTRY, DEPT, HTTP, STAGER, , MULTICAST, WORKGRP, ZSERVICE

USER Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ALWAYS	Utility Method		64 M
ALWAYS	Member of: Class connections determine the path of resolution of an agent's desired state. An _ALWAYS_ class connection is always resolved.	POLICY.WORKGRP.DEFAULT	50 C
ALWAYS	Member of: Class connections determine the path of resolution of an agent's desired state. An _ALWAYS_ class connection is always resolved.		50 C
ALWAYS	Member of: Class connections determine the path of resolution of an agent's desired state. An _ALWAYS_ class connection is always resolved.	PRDMAINT .ZSERVICE .CLIENT	50 C
EMAIL	E-Mail Address		128 V
MSITRACE	Activates MSI Verbose Trace-*vpath		255 V
NAME	Friendly Name: User-specific ID.		80 V
RETRYINT	Retry Interval (in seconds)		2 V
RETRYLIM	Number of Retries		2 V
SELPIND	Self Maintenance Display [Y/N]: Controls the self-maintenance status display screen.	N	1 V

SLFINTVL	Self Maintenance Interval (hours): Sets the frequency for self-maintenance (how often).	0	3 V
TYPSEL	Type Selection-Calc. Pack Size	Typical	10 V
UNAME	Name		30 V
USERID	Enterprise User ID: A valid eight-character alphanumeric User ID that identifies the Client Automation user. This must be the enterprise User ID that is already administered and authenticated by security software at your site. Eight-character Client Automation User ID.		8 V
ZCONFIG	Collect Hardware Information [Y/N]: Specify Y to allow hardware auditing information to be collected from the user's desktop. Specify N, and you will not be able to audit hardware assets. Note that this attribute value does not specify what is audited on a user's desktop, only that it can be.	Y	1 V
ZDLIMIT	Maximum Disk Space	0	11 V
ZGRPINFO	Gather Group Membership Information [Y/N]	N	1 V
ZOBJPFUE	Free Unused Pool Elements	Y	1 V
ZOBJPTCH	Perform Patching [Y/N]	N	1 V
ZPRIORIT	Execute Priority: For multiprocessing platforms, establishes the priority at which the agent connect process will execute Create methods. Numeric value between 1 and 99.	000	3 V
ZSETMSGA	Send Message to Audit Resource: If you insert (Y) in the ZAUDIT attribute (above), this value will determine the auditing process frequency: HOURLY, DAILY, MONTHLY.	DAILY	8 V
ZSHOW	Display Status Indicator [Y/N]:	N	1

	Enables or disables batch indicator bars. Y = Display batch indicator bars. N = Do not display batch indicator bars.		V
ZTIMEO	Client Time-out (seconds): Time that the Agent will wait for Configuration Server response, before the agent program is deactivated (timed out). Numeric value between 0 and 240 (in seconds).	240	3 V
ZTRACE	Trace On or Off [Y/N]: Enables log to include communications buffer information; also generates unique logs for create methods. Y = Trace on N = Trace off	N	1 V
ZTRACEL	Trace Log Level [0-999]: Level of tracing generated in the client log files. 0 = none 40 = acceptable for most activity 999 = maximum	040	3 V
ZVERDT	Verify Desktop [Y/N]: Use this variable to verify that user has all of his links and desktop settings. Y = Verify desktop, registry, and ini files N = Do not verify	Y	1 V

Workgroups (WORKGRP) Class

File:	PRIMARY	Description:	Use to group subscribers into functional groups. For example, a project team may be made up of subscribers from several different departments.
Domain:	POLICY		
Class:	WORKGRP		
Icon:		Can connect to:	COUNTRY, DEPT, ZSERVICE, STAGER, HTTP, MULTICAST

WORKGRP Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ALWAYS	Offers: Class connections determine the path of resolution of an agent's desired state. An _ALWAYS_ class connection is always resolved.		50 C

ALWAYS	Utility Resolution Method: Any method you specify for this attribute is unconditionally executed by Client Automation when this instance is resolved. A valid method name, for example, ZSYSTEM.ZMETHOD.PUT_ZERROR_HIST.		50 M
NAME	Friendly Name		80 V
ZSTOP	Stop Resolution Expression: Expressions evaluating to "true" in ZSTOP attributes cause resolution of the instance to be skipped. If left blank, the instance is not skipped, and resolution continues. This is useful for assigning a version to a specific set of users. Use the CSDB Editor to set this attribute.		255 E

The SECURITY Domain

The SECURITY Domain contains the tools that are used to collect vulnerability management information from devices in your environment. Once collected, this information is available for reporting purposes. See the *Sentient Enterprise Console Online Help* for more information.

Alert/Defer (ALERTDEF) Class

File:	PRIMARY	Description:	Use an instance of the ALERTDEF Class to define an application alert message or deferral.
Domain:	SECURITY		
Class:	ALERTDEF		
Icon:		Can connect to:	ZSERVICE

ALERTDEF Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ALERTMSG	Alert Message (! = High Priority) An exclamation point (!) preceding "Service Alert Message" denotes a high priority message.		100 V
DM	Alert Mode [I/U/B] Set to I for Install. Set to U for Update. Set to B for either Install or Update.	B	1 V
DN	Number of deferrals allowed Maximum number of deferrals allowed before the	0	2 V

DT	Network Threshold Network bandwidth threshold in bytes.	86000	12 V
DBT	Below Threshold Alert Size (bytes)	50000	12 V
DAT	Above Threshold Alert Size (bytes)	0	12 V
DTO	Alert Timeout (in seconds) Specify in seconds the amount of time to display the Defer Alert dialog box. After the timeout is reached, the action specified in the DA (Action on timeout) variable, will be taken.	120	4 V
DA	Action on timeout Cont/Defer [C/D] Specify C to continue with the specified action if the subscriber does not respond to the defer alert dialog box by the value specified in the DTO (Alert Timeout). Specify D to defer the specified action.	C	1 V
DI	Allow Install Deferral up to [DATE] After this date is reached, the option to defer installation will no longer be available. The application will be installed. This should be in the format YYYYMMDD.	NA	20 V
DU	Allow Update Deferral up to [DATE] After this date is reached, the option to defer installation will no longer be available. The application will be updated. This should be in the format YYYYMMDD.	NA	20 V
NAME	Friendly Name Friendly Name for the instance.	Default	80 V
DELOPTNS	Defer Options (Do Not Modify) This attribute is used to resolve the values of the other attributes of this class. <i>Do not modify.</i>	&(DM),& (DN),& (DT), &(DBT),& (DAT), &(DTO),& (DA), &(DI),& (DU)	12 V

Application (ZSERVICE) Class

File:	PRIMARY	Description:	Use an instance of the ZSERVICE Class to define an application and its characteristics to Client Automation. An instance of the ZSERVICE Class always includes the resource files that comprise it.
Domain:	SECURITY		
Class:	ZSERVICE		
Icon:		Can connect to:	PACKAGE, FILE, BEHAVIOR, DIALOG, DEPT, WORKGRP, TIMER, VGROUP, STAGER, HTTP, EXECUTE, ZSERVICE, PDACONFIG

ZSERVICE Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ALWAYS	Contains Class connections determine the path of resolution of an agent's desired state. An _ALWAYS_ class connection is always resolved.		50 C
ALWAYS	Contains An _ALWAYS_ class connection is always resolved. An Includes connection is used to express inter-package relationships.		50 I
ALWAYS	Utility Resolution Method Any method you specify for this attribute is unconditionally executed by Client Automation when this instance is resolved. A valid method name, for example, ZSYSTEM.ZMETHOD.PUT_ZERROR_HIST.		51 M
ADAPTIVE	Auto Adaptability [Y/N] Indicates whether the installed package is dependent on client settings that must be monitored periodically. If the settings change, the client must reconnect to the Configuration Server to get new or different components. Useful for plug-and-play services.		1 V
AUTHOR	Author Name Name of the author of the service that appears in the properties for the service in the Service List.		50 V
BDELETE	Service Pre-Delete Method Method to run before deleting the service.		100 M
BREPAIR	Service Pre-Update Method		100

			M
BUPDATE	Service Pre-Repair Method Method to run before repairing the service.		100 M
CACHE	App Element Caching [Y/N]	N	1 V
CACHELIM	Percent Disk Limit for Cache Cache limit, which is defined as the percentage of used drive space. Type a number between 000 and 100. Note: Used for Windows Installer applications only. If the percentage of used space is greater than the cache limit, then all of the cached files for the product are removed and the cache folder is deleted. This is checked after every file is cached on the disk.	000	3 V
CACHELOC	CACHE Location on Client For Windows Installer applications only. Location of the folder on the agent computer that is used to cache the compressed application files needed for the product. Sentient support for Windows Installer tags the PRODGUID value to this value to create the folder. For example, If CACHELOC=c:\progra~1\PSL, and PRODGUID = 12345_XXXX\cache, the cache folder would be: c:\progra~1\PSL\12345_XXXX\cache. Note: The folder name \cache is automatically appended to PRODGUID. If you are not deploying a Windows Installer-enabled application, the files will be cached in IDMDATA.	_UNDEF_	254 V
CATGROUP	Catalog Group Name Type a name for the catalog to be displayed to subscribers when they click Properties in the Application Self-service Manager user interface.		30 V
COMPSIZE	Application Size – Compressed The size of the compressed application displayed to the subscribers when they click Properties in the Application Self-service Manager user interface.		12 V

	Cumulative value of the COMPSIZE defined in the PACKAGE Class.		
DELDATE	Delete Date Indicates when the application was removed (in local time) from the agent computer. The Agent manages and maintains this attribute.		32 V
DESCRIPT	Application Description Description of the service that appears in the properties for the service in the Service List. Value is set initially in the Long Description field in the New Application Wizard (used to create a service).		100 V
ERTYPE	Event Reporting Method [O/E/X] Set on the Configuration Server, this sends an APPEVENT object to the Configuration Server. O = Object Currently supports object format only.	O	1 V
EVENTS	Events to Report Indicates which events to report on. Event to report on: AI = Application Install AD = Application De-install AU = Application Update AR = Application Repair AV = Application Verify VA = Version Activation VD = Version Deactivation What (about the event) to report on: S = Success F = Failure B = Both Success and Failure N = None	AI=B AD=B AU=B AR=B AV=F VA=B VD=B	50 V
INSTDATE	Installed Date Indicates when the application was installed (in local time) on the agent computer. The Agent manages and maintains this attribute. The date is in text form, as in: Jan 28, 1998 16:10:00.		32 V
ITEMS	Number of Component Instances Cumulative numeric value of the component instances of all the attached packages. (See ITEMS from PACKAGE Class.)		7 V
LREPAIR	Local Repair [Y/N] Enables local repair of broken applications. If an application is broken because of missing files, the files (stored locally) can be used to repair the application.		1 V
MCELIBL	Service Multicast Eligible [Y/N]	Y	1

	Indicates if the application is eligible for multi-casting.		V
NAME	Friendly Name Friendly name for the service. This appears in the properties for the service in the Service List.		80 V
OWNER	Application Contact Package owner, as documented by the Admin Publisher.		50 V
PRICE	Price Type the price of an application to be displayed to subscribers when they click Properties in the Application Self-service Manager user interface.		10 V
PUBDATE	Published Date of Service Date service was published.		32 V
REBOOT	Install/Update/Delete/Version Change Used to restart the agent computer after the service is installed. Event to report on: AI = Install AD = Deinstall AU = Update AR = Repair AV = Verify Type of reboot: S = Soft Boot (Default of type Y panel.) H = Hard Boot (Default of type A panel.) N = None Type of panel: Q = No panel. A = OK button only. Y = OK and Cancel button. Type of connect: None specified: Reboot on Machine connect (context = m). U = reboot on user connect only (context = u). MU = reboot when both machine and user parts of the service have been installed. Example: AI=S performs a soft boot on application installation.		64 V
RECONFIG	Reconfiguration Enabled [Y/N] Indicates whether an application can be relocated after it has been installed.		1 V
REMOVAL	Un-Managed Behavior [A/D/U] Controls removal of the service. If ZVCMO is set to M, set REMOVAL to:	D	1 V

	A = Abandon (deletes the objects, but not the components) D = Delete (deletes the objects and components) If ZVCMO is set to O, set REMOVAL to: A = Abandon (deletes the objects, but not the components) D = Delete (deletes the objects and components) U = Unmanage (does not delete the objects or components)		
RSTRSIZE	Download Restart Threshold (in bytes) Use the RSTRSIZE attribute in the appropriate ZSERVICE Class instance to control which files are enabled for check point restart based on the amount of data being downloaded (in bytes). This is for Windows Installer-Enabled Applications (MSI)	0	11 V
RUNDLG	Dialog Processing [Y/N] Specifies whether to enable processing of dialog boxes during the installation of the service.	N	1 V
RUNLEVEL	Runlevel Support for UNIX.		1 V
SCHEDOK	Update Schedule Locally [Y/N] Specifies whether the subscribers are allowed to change the update schedule for the service locally. Y = Subscriber is allowed to change the schedule. N = Configuration Server controls the update schedule.		1 V
SIZE	Application Size – Uncompressed The size of the uncompressed application displayed to the subscribers when they click Properties in the Application Self-service Manager user interface.		12 V
THROTYPE	[ADAPTIVE/RESERVED/NONE/] Type of Bandwidth throttling to use. Set to ADAPTIVE to yield to other services that are using the network. Set to RESERVED to allow for a specific reservation of the bandwidth. It is the maximum percentage of network bandwidth to use. Set to NONE for no bandwidth throttling, and use the maximum available bandwidth. NONE is the default.		8 V
UIOPTION	Progress Indicator [NONE/FULL/INFO] Controls whether the status window appears. NONE = No interface appears. FULL = Interface appears and Cancel button is available.		4 V

	INFO = Interface appears with no option to cancel.		
UIOPTMSI	MSI Progress Indicator [NONE/FULL/INFO] Controls whether the MSI status window appears. NONE = No interface appears. FULL = Displays what is happening on the agent computer and provides the subscriber with controls to make changes. INFO = Displays what is happening on the agent computer, but disables all the controls so that the subscriber cannot make any changes. See the Setup log files and Event Viewer for warnings, errors, successes, failures, and other details.		
UPDDATE	Upgrade Date (Programmatic) Stores the date when the file was updated on the Configuration Server.		32 V
UPGDATE	When Application was Upgraded Stores the date when the application was last upgraded (in local time) on the agent computer. This value is set by the Sentient user after the dialogs have been installed. This date can be used for reporting purposes. The date is in text form, as in: Jan 28, 1998 16:10:00.		32 V
URL	WEB URL Name Address of a Web page where the subscriber can find additional information about the service.		100 V
VENDOR	Vendor Name Name of the vendor of the service.		24 V
VERDATE	Verified Date of Service Indicates when the application was last verified (in local time) on the agent computer. The Agent manages and maintains this attribute.		32 V
VERSION	Version Description Version of the software. This appears in the properties for the service in the Service List.		50 V
ZAVIS	Available, Verified, Installed, Sync F The Agent manages and maintains this attribute to show the different states of the application in the Service List. Y = Yes N = No X = Unknown The four states are:	YXNX	4 V

	Available indicates whether a service is available from the Configuration Server. Verified indicates whether a service has been verified. Installed indicates whether the service has been installed. Synchronized indicates whether the installed service has all of the latest changes from the Configuration Server.		
ZBITARCH	Target Bit Architecture [/32/64] Set to 32 to deploy to 32-bit architectures. Set to 64 to deploy to 64-bit architectures. Set to blank to deploy to all architectures.		8 V
ZBWAPCT	Adaptive Percentage Adaptive minimum percentage of bandwidth used when using bandwidth throttling.		3 V
ZBWPCT	Reserved Percentage Reserved maximum percentage bandwidth used when using bandwidth throttling.		3 V
ZCONTEXT	Component Context [M/U]		8 V
ZCREATE	Service Installation Method Sentient-supplied method or fully qualified, user-defined method.		255 M
ZDELETE	Service Delete Method Sentient-supplied method or fully qualified, user-defined method.		255 M
ZDISCONN	Disconnect on Install [Y/N] Allows the client to disconnect from the Configuration Server if there is an open session with the Configuration Server. Y = Disconnects the client from the Configuration Server. N = Does not disconnect from the client from the Configuration Server.	Y	1 V
ZINIT	Service Initialization Method A fully qualified, user-defined method.		100 M
ZREPAIR	Service Repair Method Name of the method to repair the service.		255 M
ZSTOP000 to ZSTOP002	Stop Resolution Expression Expressions evaluating to "true" in ZSTOP attributes cause resolution of the instance to be skipped. If left blank, the instance is not skipped, and resolution continues. This is useful for		100 E

ZSTOP999	Stop Unless RCA Connect		50 E
ZSVCCAT	Service Visible in Catalog? [Y/N]		1 V
ZSVCCSTA	Service Status on Client (999) Status code for the service that leads you to the reason why files for a service may not be deployed correctly. 000 = All actions for service completed. 008 = At least one resource had a writer error. 014 = No disk space for at least one resource. 015 = ZDLIMIT exceeded for at least one resource. 024 = A method timed out. 036 = A create method set ZBRC>7. 900 = Not all files were verified. No work done for service.	999	3 V
ZSVCMO	Mandatory/Optional Service [M/O/MO/OM] Indicates whether the service is mandatory or optional on the Agent. If you are using Client Automation Application Manager <i>and</i> Application Self-service Manager, you could also specify mandatory and then optional, MO , or optional then mandatory, OM . The first character indicates how the application should be handled before installation. The second character indicates how the application should be handled after installation.	O	1 V
ZSVCMODE	Application Context [M/U/MU] Specifies whether the service is to be installed in the Machine or User context. M = Machine U = User		2 V
ZSVCNAME	Service Name/Description	Unknown Service	24 V
ZSVCPRI	Service Create Ordering [01-99] Enables you to specify a priority level for this service. If disk space on the agent desktop is not available, services will be downloaded based on the priority level set here. 01 = Highest priority 10 = Default value 99 = Lowest priority		2 V
ZSVCTTYP	Application Target Type [A/S] Indicates which Agent will receive this application. A = Application Manager S = Application Self-service Manager		2 V

ZSYSACCT	Install Under System Account [Y/N/M] Specifies whether to install the service under the system account or the user's account. Y = The application is installed using the system rights. N = The application is installed using the rights of the user who is logged on. M = The application is installed using the system rights if the current connect is in the machine context (context =m).	Y	1 V
ZUPDATE	Service Update Method Sentient-supplied method or fully qualified, user-defined method.		255 M
ZVERIFY	Service Verify Method		255 M

Application Control Class (APPCNTRL)

File:	PRIMARY	Description:	The Application Control Class is used to publish and execute Application Management Profiles (AMPs).
Domain:	SECURITY		
Class:	APPCNTRL		
Icon:		Can connect to:	

Application Control Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
AUTHOR	Author Name		255 V
PMODDT	Parent Modified date/time (ISO8601)		32 V
PCREDT	Parent Created date/time (ISO8601)		32 V
MODDT	Modified Date/Time (ISO8601)		32 V
CREDT	Created Date/Time (ISO8601)		32 V
TITLE	Title		255 V

TYPE	Type of Profile [CAP/MAP]		8 V
LANGUAGE	Language		8 V
SCHEMAVER	Schema Version		32 V
REVISION	Revision		32 V
PRODUCT	Product Name		255 V
PRODVER	Product Version		32 V
ZRSCCFIL	File Name		255 V
LOCATION	Resource Location		255 V
ZRSCSIZE	Resource Size		12 V
ZRSC TYP	Resource Type		32 V
ZRSCSIG	Resource		32 V
SIGTYPE	Signature Type		5 V
ZRSCRSTR	Restart options		1 V
ZCMPSIZE	Resource compressed size		12 V
ZRSCDATE	Resource Date		8 V
ZRSC TIME	Resource Time		8 V
ZRSCVRFY	Verify options		32 V
ZRSCCRC	Resource CRC		8 V
ZSTATUS	Instance Status	999	3

			V
ZOBJPRI	Instance Priority		3 V
ZCONTEXT	Instance Context		1 V

Application Packages (PACKAGE) Class

File:	PRIMARY	Description:	The PACKAGE Class contains an instance with associated references used to identify an application component model. An administrator uses the Admin Publisher to create a package instance.
Domain:	SECURITY		
Class:	PACKAGE		
Icon:		Can connect to:	DESKTOP, FILE, PATH, MSIFEATURES, MSI, PACKAGE, PDACONFIG, REGISTRY

PACKAGE Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ALWAYS	Connect to Class connections determine the path of resolution of an agent's desired state. An _ALWAYS_ class connection is always resolved.		50 C
ALWAYS	Package Includes Class connections determine the path of resolution of an agent's desired state. An _ALWAYS_ class connection is always resolved.		50 C
BEHAVIOR	Behavior Includes connection to BEHAVIOR Class instances		255 I
COMPSize	Package Compressed Resource Size Compressed size of package (in bytes).		12 V
DESCRIPT	Application Description Description of the service that appears in the properties for the service in the Service List. Value is set initially in the Long Description field in the New Application Wizard (used to create a service).		255 V
DESKTOP	Desktop Includes connection to DESKTOP Class instances.		255 I
FILE	File		255

	Includes connection to FILE Class instances.		I
INCLUDES	Package Includes Generic Includes connection.		50 I
ITEMS	Number of Component instances Number of component instances in a package at time of publishing.		7 V
NAME	Friendly Name		80 V
OWNER	Application Contact Package owner, as documented by the Admin Publisher.		80 V
PATH	Path Includes connection to PATH Class instances.		255 I
PKGNAME	Package Name		80 V
REGISTRY	Registry Includes connection to REGISTRY Class instances.		255 I
RELEASE	Package Release Release level of the package, as documented by the Admin Publisher.		80 V
REQUIRES	Required Package Level Generic Requires connection.		80 R
SIZE	Package Resource Size– Uncompressed Size of package (in bytes).		12 V
UNIXFILE	UNIX File Includes connection to UNIXFILE Class instances.		255 I
ZBITARCH	Target Bit Architecture [/32/64] Set to 32 to deploy to 32-bit architectures. Set to 64 to deploy to 64-bit architectures. Set to blank to deploy to all architectures.	_UNDEF_	8 V
ZSTOP000 to ZSTOP003	Stop Resolution Expression Expressions evaluating to "true" in ZSTOP attributes cause resolution of the instance to be skipped. If left blank, the instance is not skipped, and resolution continues. This is useful for assigning a version to a specific set of users. Use the CSDB Editor to set this attribute.		255 E

Auto Run (EXECUTE) Class

File:	PRIMARY	Description:	An instance of the EXECUTE Class defines a file, or a group of files, that will be launched after an application has been installed on the agent desktop.
Domain:	SECURITY		
Class:	EXECUTE		
Icon:		Can connect to:	

EXECUTE Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ARGUMENT	Arguments passed to the program Stores the arguments that have to be passed to the program to be launched.		255 V
EXECPATH	Fully qualified executable name Stores the fully qualified name of the executable to be launched after an application install.		255 V
WORKDIR	Starting Directory Stores the name of the directory/folder from which the executable is started.		255 V
ZCREATE	Create Method Stores the Client Automation method name that is used to create a list of programs that need to be executed after an application is installed.	EXECUTE	50 M
ZDELETE	Delete Method Sentient-supplied method or fully qualified, user-defined method.	EXECUTE	50 M
ZVERIFY	Verify Method	EXECUTE	50 M

Behavior Services (BEHAVIOR) Class

File:	PRIMARY	Description:	Defining an instance of the BEHAVIOR Class lets you define a file, or a group of files, to include in an application service. An instance of this class lets you specify such characteristics about the file as the file's name, size, and date stamp.
Domain:	SECURITY		
Class:	BEHAVIOR		
Icon:		Can connect to:	PACKAGE

BEHAVIOR Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ALWAYS	Connect to Class connections determine the path of resolution of an agent's desired state. An _ALWAYS_ class connection is always resolved.		50 C
CMPMD5	Compressed File MD5 Signature	_UNDEF_	32 V
FILEMD5	File MD5 Signature	_UNDEF_	32 V
LOCATION	Location		255 V
NAME	Friendly Name Name that appears in the CSDB Editor for this class.		80 V
OBJGUID	Instance GUID Object Global Unique Identification – <i>for future use</i> .	_UNDEF_	32 V
ZCMPKEY	Compression Key Not used.		80 V
ZCMPSIZE	Compressed File Size From promotion process if the compressed file option was selected. The size of the compressed file.		11 V
ZCREATE	Method to Install Behavior Name of method to install a resource file. Sentient-supplied method or fully qualified, user-defined method.		50 M
ZDELETE	Method to De-install Behavior Sentient-supplied method or a fully qualified, user-defined method.	EDMREMF	50 M
ZFILEUPD	Client File Update/Add Method If a file has been updated by an administrator, this method will be executed. Sentient-supplied method or fully qualified, user-defined method.		50 V
ZINIT	Resource Initialization Method Sentient-supplied method or a fully qualified, user-defined method.		50 M
ZOPENERR	Client Method on File Open Error Sentient-supplied method or a fully qualified,	EDMLKM	50

	user-defined method.		
ZRSCCFIL	Behavior File Name Specific file to be distributed. A generic pattern (a wildcard character, such as *) or an explicit file name.		255 V
ZRSCCKPT	Resource Checkpoint: Not used. Replaced by ZRSCRSTR.		1 V
ZRSCCONF	Confirm File Download [Y/N] Y enables you to confirm that the resource was downloaded.	N	1 V
ZRSCCRC	Resource CRC From the promotion process if the CRC option was selected. Eight-character calculated hexadecimal value.		8 V
ZRSCCSTA	Client File Status Resource status code. 000 = Successfully downloaded 001 = Data can be found in EDMDATA directory 004 = Resource with no data 008 = Could not create the file 009 = No disk space for file (ZRSCMO=O) 010 = ZDLIMIT was exceeded (ZRSCMO=O) 011 = No data in RCA for resource 014 = No disk space (ZRSCMO=M) 015 = ZDLIMIT was exceeded (ZRSCMO=M) 024 = Time-out waiting for method. 036 = A method was set ZBRC>7 999 = No work done for resource	999	3 V
ZRSCDATE	Behavior Date Stamp Date stamp on resource file from promote process. A value up to eight digits long.		8 V
ZRSCMDAT	Manager Stores Data [Y/N]	Y	1 V
ZRSCMMEM	PDS Member Name PDS where resource actually resides. PDS name where resource resides in Manager (MVS only).		8 V
ZRSCMO	Mandatory/Optional [M/O]	M	1 V
ZRSCNAME	Behavior Name Name or description of resource. Valid name of resource to distribute to the Agent.	&ZRSCCFIL	30

ZRSCOBJN	Persistent Object Name The name of the in-storage object. Valid only when the ZLOCMGR location is STORAGE.		8 V
ZRSCPRI	Create Sequence Priority Enables you to specify a priority level for this resource. If disk space on the client desktop is not available, resources will be downloaded based on the priority set here. 01 = Highest priority 10 = Default value 99 = Lowest priority	10	2 V
ZRSCRASH	DOS File Attribute [R/A/S/H] Enables you to specify file attributes for a resource. Read-only Archive System Hidden.		4 V
ZRSCRSTR	Restart [Y/N] You can either restart a download process at the point it was interrupted (Yes), or resend the entire object (No).	N	3 V
ZRSCSIZE	Behavior Size Byte count of resource file from promote process. A value up to 11 digits long.		11 V
ZRSCSRC	Resource Source,that is Publisher		20 V
ZRSCSRC	Resource Source		1 V
ZRSCSTYP	Server File Type [BINARY/TEXT] File format for file download: Binary, Text, or Object.	BINARY	8 V
ZRSCSVRB	Version Skip Rebuild Flag [Y/N] Enables you to skip the reactivation of version resources if the version is already active. Y = Skip the rebuilding of version resources N = Rebuild version resources	Y	1 V
ZRSCTIME	Behavior Time Stamp Time stamp on resource file from promote process.		8 V
ZRSCVERS	Resource Version - From Promote Version number of resource from promotion process.		12 V

ZRSCVRFY	Verify Resource File on Connect Verifies the file statistics of a resource file on the desktop during each agent connect process. If you are deploying versioned resources, this attribute is ignored. The rules for the E(xist) flag apply to both Client Automation- and non-Sentient distributed files. (For example, a Client Automation-managed application file or CONFIG.SYS file.) Thus, you must specify Y or N for an existing file to be distributed. Y = The file will be distributed if one or more of the following conditions is met: The file has not been distributed before. The date, time, or size of the file has changed. The file was re-promoted. N = The file will be distributed if: The file has not been distributed before. Or, The file was re-promoted. E = The file will be distributed only if it does not currently exist on the desktop. M = Reserved for Client Automation maintenance files. During the last step of the connect, the file is moved from the EDMDATA subdirectory to its proper location. Example: can be used to update EDMGETRS . EXE. S = This file is distributed if the file size has changed. T = The file is distributed if the file time has changed. D = File is distributed if the file date has changed. L = Forces the lock method to be used. The file will be automatically queued up to be deployed on restart, and the reboot flag will be set. If the L flag is not specified, and the file to be deployed is currently in use (locked) the lock method is automatically used anyway.	Y	1 V
	What the L flag does is force the lock method to be always used to deploy this file, whether or not it is currently in use (locked). The L flag will also skip decompression and dispatch the open error method (RADLKM). This is used for system dlls that must be replaced. Note that for the file to be verified during subsequent connects, you must also set this variable to Y. U = File is distributed only if the date and time		

	on the Configuration Server is newer than the date and time on the desktop. R = Resource Stamp. For internal use only. V = (Windows only) Used in when normal date/time stamp check cannot be used. Usually when a <code>.dll</code> or <code>.exe</code> is involved and the date and time stamp are changing frequently. <code>Radconct.exe</code> checks the internal version number of the target file and compares it with the one that Client Automation has. If the incoming file has a higher internal version number than it will be placed on the desktop over the existing file. P = (Windows only) Set during packaging, enforces Windows System File Protection (SFP). If the verification is successful, the resource is in the desired state and no further verification is necessary. If the verification fails, Client Automation checks to see if System File Protection protects the file. If the file is protected, it will not be deployed. If the ZRSCVRFY attribute is set to P and deployment of this resource and the service are marked as failed due to SFP processing. If you do not enforce SFP, the file is marked as having passed verification.		
ZUPDATE	Client Instance Update Method If an object (resource instance) has been updated by an administrator, this method will be executed.		50 M
ZUSERID	Admin ID		30 V

Class Defaults (METACLAS) Class

File:	PRIMARY	Description:	An instance of the METACLAS Class defines prerequisites.
Domain:	SECURITY		
Class:	METACLAS		
Icon:		Can connect to:	PACKAGE

METACLAS Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type

ALWAYS	Connect to Class connections determine the path of resolution of an agent's desired state. An _ALWAYS_ class connection is always resolved.		50 C
CCREATE	Client Install Method		128 V
CDATAUPD	Client File Update/Add Method		128 V
CDELETE	Client De-install Method		128 V
CERROR	Client Error Method		128 V
CINIT	Client Initialization Method		128 V
COPENERR	Client Method on File Open Error		128 V
CUPDATE	Client Update Method		128 V
CVERIFY	Client Verify Method		128 V
NAME	Friendly Name		80 V

Desktop (DESKTOP) Class

File:	PRIMARY	Description:	Instances of this class specify those resources which are icons appearing in the Windows environment. The attributes of this class also specify the properties of the Client Automation resources. Some of the icons could store just-in-time executables, which would be updated by connecting to the Configuration Server before they are run.
Domain:	SECURITY		
Class:	DESKTOP		
Icon:		Can connect to:	CATALOG, ZSERVICE

DESKTOP Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type

ALWAYS	Connect to Class connections determine the path of resolution of an agent's desired state. An _ALWAYS_ class connection is always resolved.		50 C
JITARGS	Application Manager Arguments Stores the arguments to be passed to the just-in-time executable.		255 V
JITLNK	Application Manager Installation [Y/N]		1 V
JITPATH	Application Manager Path Stores the fully qualified name of the just-in-time executable. Should be a fully qualified path name with the drive and directory. If the directory name contains embedded spaces then the whole value should be in double quotes, e.g., "c:\program files\PSL\test.exe".		255 V
MACHUSER	Machine User		32 V
NAME	Friendly Name		80 V
VERIFY	Verify [Y/E/N]		1 V
ZCONTEXT	Component Context		8 V
ZCREATE	Method to Create Link Sentient-supplied method or fully qualified, user-defined method.	RADCLNK	50 M
ZDELETE	Method to Delete Link Sentient-supplied method or fully qualified, user-defined method.	RADDLNK	50 M
ZINSTALL	Method CLNK Processing [Y/N] Stores if the links/icons have to be installed.		1 V
ZOBJPRI	Priority [09-99]	60	2 V
ZPUBVER	Publisher Version	_UNDEF_	8 V
ZRSCLFIL	Link Path Stores the fully qualified name for the link. The links are usually created in the Windows folder,		255 V

	create method to store the value of the current user.		
ZRSCMO	Mandatory/Optional [M/O]	o	1 V
ZSTATUS	Client Status	999	3 V
ZW95ARGS	Command Line Parameters Stores the arguments to be passed to the executable specified in the ZW95PATH field.		255 V
ZW95DESC	Icon Description A brief description for the icon/link.		60 V
ZW95HOTK	Hot Key Assigned by MS Windows		12 V
ZW95ICON	Icon File		128 V
ZW95INDX	Index of Icon		60 V
ZW95PATH	Command Line Stores the name of the executable associated with this icon/link.		128 V
ZW95SHOC	Visibility Flag [00000000/00000001]	00000001	8 V
ZW95WDIR	Working Directory Stores the starting directory for the program associated with the icon or link. Should be a fully qualified path name with the drive and directory. If the directory name contains embedded spaces then the whole value should be in double quotes, e.g., "c:\program files\PSL".		50 V

Dialog Services (DIALOG) Class

File:	PRIMARY	Description:	Defining an instance of the DIALOG Class lets you define an install process for an application and its characteristics to Client Automation. An instance of the DIALOG Class always includes the resource files that comprise it, which could be either the panels or scripts. The panels and scripts are used to initiate an interactive session, to gather information pertinent to installing a particular application.
Domain:	SECURITY		
Class:	DIALOG		

Icon:		Can connect to:	BEHAVIOR, INSTALL, PACKAGE, PANEL

DIALOG Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ALWAYS	Connect to Class connections determine the path of resolution of an agent's desired state. An _ALWAYS_ class connection is always resolved.		50 C
ALWAYS	Utility Resolution Method Any method you specify for this attribute is unconditionally executed by Sentient when this object is resolved. A valid method name, for example, ZSYSTEM.ZMETHOD.PUT_ZERROR_HIST.		51 M
AUTHOR	Author Name Name of the author of the dialogs.		50 V
CATALOG	Catalog Name The catalog name for this dialog.	&(CATALOG.CATNAME)	32 V
COMPSize	Application Size- Compressed Stores the compressed size of all the dialog components (informational). This value is filled in by the Admin Publisher when the dialogs are stored in the CSDB.		12 V
DESCRIPT	Application Description Description of the dialog that appears in the properties for the service in the Service List. Value is set initially in the Long Description field in the New Application Wizard (used to create a service).		255 V
INSTDATE	Installed Date Stores the date and time when the dialog was installed on the client desktop. This value is set by the Sentient user after a dialog has been installed. It is useful for reporting purposes. (informational). The date is in text form, as in: Jan 28, 1998 16:10:00.		32 V
NAME	Friendly Name Stores the dialog name.		80 V
PRICE	Price		10 V

PUBDATE	Published Date of Service Date service was published.		32 V
SCHEDOK	Update Schedule Locally [Y/N] Specifies whether the subscribers are allowed to change the update schedule for the service locally. Y = Subscriber is allowed to change the schedule. N = Configuration Server controls the update schedule.		1 V
SIZE	Application Size - Uncompressed		12 V
UPDDATE	Upgrade Date (Programmatic) Stores the date when the file was updated on the Configuration Server.		12 V
UPGDATE	When Application was Upgraded Stores the date when the application was last upgraded (in local time) on the agent computer. This value is set by the Sentient user after the dialogs have been installed. This date can be used for reporting purposes. The date is in text form, as in: Jan 28, 1998 16:10:00.		12 V
URL	WEB URL Name Stores the address of a Web page, where additional information is available about this application. http://www.company_name.com		255 V
VENDOR	Vendor Name		24 V
VERDATE	Verified Date of Service		32 V
ZAVIS	Available, Verified, Installed, Sync F The Agent manages and maintains this attribute to show the different states of the application in the Service List. Y = Yes N = No X = Unknown The four states are: Available indicates whether a service is available from the Configuration Server. Verified indicates whether a service has been verified. Installed indicates whether the service has been installed.	YXNX	4 V

	Synchronized indicates whether the installed service has all of the latest changes from the Configuration Server.		
ZCREATE	Service Installation Method Sentient-supplied method or fully qualified, user-defined method.	50 M	
ZDELETE	Service Delete Method Sentient-supplied method or fully qualified, user-defined method.		50 M
ZINIT	Service Initialization Method A fully qualified, user-defined method.		50 M
ZSVCCONF	Confirm Service Install [Y/N] Enables you to confirm the service was downloaded.	N	1 V
ZSVCCSTA	Service Status on Client (999) Service status code. 000 = All actions for service completed 008 = At least one resource had a writer error 014 = No disk space for at least one resource 015 = ZDLIMIT exceeded for at least one resource 024 = A method timed out 036 = A create method set ZBRC>7 900 = Not all files were verified. No work done for service	999	3 V
ZSVCINST	Service Installation Date/Time		32 V
ZSVCMO	Mandatory/Optional Service [M/O/MO/OM] Indicates whether the service is mandatory or optional on the Agent. If you are using Application Manager <i>and</i> Application Self-service Manager, you could also specify mandatory and then optional, MO , or optional then mandatory, OM . The first character indicates how the application should be handled before installation. The second character indicates how the application should be handled after installation.	M	1 V
ZSVCNAME	Service Name/Description Name of the service, e.g., VB Calculator.		24 V
ZSVCPRI	Service Create Ordering [01-99] Enables you to specify a priority level for this service. If disk space on the agent desktop is not available, services will be downloaded based on		2 V

	the priority level set here. 01 = Highest priority 10 = Default value 99 = Lowest priority		
ZSVCSEL	Update Force Bypass Remove [U/F/B/R] If ZSVCMO is optional, this attribute can be Update, Force, Remove, or Bypass. Update Force Remove Bypass		1 V
ZSVCUPD	Service Update Date/Time		32 V
ZSVCVERS	Version Activation Level Version number of resources or service from the promotion process.		12 V
ZUPDATE	Service Update Method Sentient-supplied method or fully qualified, user-defined method.		50 M

File Resources (FILE) Class

File:	PRIMARY	Description:	An instance of the FILE Class defines a file, or a group of files, to include in an application service. An instance of this class lets you specify such characteristics about the file as the file's name, size, and date stamp.
Domain:	SECURITY		
Class:	FILE		
Icon:		Can connect to:	

FILE Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ALWAYS	Connect to Class connections determine the path of resolution of an agent's desired state. An _ALWAYS_ class connection is always resolved.		50 C
FVERSION	File Version	_UNDEF_	30 V
LOCATION	Location Stores the location of the file. The location of the drive and directory is encapsulated in the PATH		255 V

	object. The value is filled by the publisher and contains value of the form &P (D001ABCDEF01_12345678) , which is an indirect reference to the PATH object. The &P triggers a special kind of variable substitution which looks for the drive and directory from the PATH object.		
NAME	Friendly Name		80 V
OBJGUID	Instance GUID Object Global Unique Identification – <i>for future use</i> .	_UNDEF_	32 V
PVERSION	Promote Version	_UNDEF_	30 V
SIGTYPE	File Signature Type		3 V
ZCMPKEY	Compression Key Not used.	_UNDEF_	80 V
ZCMPSIZE	Compressed File Size From promotion process if the compressed file option was selected. The size of the compressed file.		11 V
ZCONTEXT	Component Context		8 V
ZCREATE	Method to Install Resource Sentient-supplied method or fully qualified, user-defined method.		50 M
ZDELETE	Method to De-install Resource Sentient-supplied method or fully qualified, user-defined method.	RADREMF	50 M
ZFILEUPD	Client File Update/Add Method Method that is executed if a file has been updated by an administrator. Sentient-supplied method or fully qualified, user-defined method		50 M
ZINIT	Resource Initialization Method Sentient-supplied method or fully qualified, user-defined method.		50 M
ZOBJPATCH	Perform Patching [Y/N]	Y	1 V
ZOPENERR	Client Method on File Open Error Sentient-supplied method or fully qualified,	RADLKM	50 M

ZPUBVER	Publisher Version	_UNDEF_	8 V
ZREPAIR	Client Instance Repair Method		50 M
ZRSCCFIL	Resource File Name Specific file to be distributed. A generic pattern (a wildcard character, such as *) or an explicit filename.		255 V
ZRSCCKPT	Resource Checkpoint Not used. Replaced by ZRSCRSTR.		1 V
ZRSCCONF	Confirm File Download [Y/N] Y enables you to confirm that the resource was downloaded.	Y	1 V
ZRSCCRC	Resource CRC From the promotion process if the CRC option was selected. Eight-character calculated hexadecimal value.		8 V
ZRSCCSTA	Client File Status Resource status code. 000 = Successfully downloaded 001 = Data can be found in EDMDATA directory 004 = Resource with no data 008 = Could not create the file 009 = No disk space for file (ZRSCMO=O) 010 = ZDLIMIT was exceeded (ZRSCMO=O) 011 = No data in Sentient for resource 014 = No disk space (ZRSCMO=M) 015 = ZDLIMIT was exceeded (ZRSCMO=M) 024 = Time-out waiting for method 036 = A method was set ZBRC>7 999 = No work done for resource	999	3 V
ZRSCDATE	Resource Date Stamp Date stamp on resource file from promote process.		8 V
ZRSCMMEM	PDS Member Name PDS where resource actually resides. PDS name where resource resides in Manager (MVS only).		8 V
ZRSCMO	Mandatory/Optional [M/O]	M	1 V
ZRSCNAME	Resource Name Name or description of resource. Valid name of	&ZRSCCFIL	30 V

ZRSCOBJN	Persistent Object Name The name of the in-storage object. Valid only when the ZLOCMGR location is STORAGE. A valid in-storage object name.		8 V
ZRSCPADM	Admin ID		30 V
ZRSCPRI	Create Sequence Priority Enables you to specify a priority level for this resource. If disk space on the client desktop is not available, resources will be downloaded based on the priority set here. 01 = Highest priority 10 = Default value 99 = Lowest priority	50	2 V
ZRSCRASH	DOS File Attribute [R/A/S/H] Enables you to specify file attributes for a resource. Read-only Archive System and/or Hidden		4 V
ZRSCRSTR	Restart [Y/N/R] Restart a download process at the point it was interrupted (Y), or resend the entire object (No). Set to (R) to bypass Windows file caching and write directly to disk (allowing checkpoint restart to operate correctly).	N	3 V
ZRSCSIG	File Signature		32 V
ZRSCSIZE	Resource Size Byte count of resource file from promote process.		11 V
ZRSCSRC	Resource Source, i.e., Publisher Stores the source of the resource: If resource was retrieved from Configuration Server, the value would be M. If resource existed on the desktop, the value would be E. If resource was retrieved from Stager, value would be S. These values are filled in by the Sentient user during an application installation. When the user uninstalls an application, only the files that were retrieved from the Configuration Server or the		20 V

	Staging Server would be deleted. The files that existed on the desktop prior to an application install would not be deleted.		
ZRSCSTYP	Server File Type [BINARY/TEXT] File format for file download: Binary, Text, or Object.	BINARY	8 V
ZRSCSVRB	Version Skip Rebuild Flag [Y/N] Enables you to skip the reactivation of version resources if the version is already active. Y = Skip the rebuilding of version resources N = Rebuild version resources		1 V
ZRSCTIME	Resource Time Stamp Time stamp on resource file from promote process.		8 V
ZRSCVERS	Resource Version - From Promote Version number of resource from promotion process.		12 V
ZRSCVRFY	Verify Resource File on Connect Verifies the file statistics of a resource file on the desktop during each agent connect process. If you are deploying versioned resources, this attribute is ignored. The rules for the E(xist) flag apply to both Client Automation- and non-Sentient distributed files. (For example, a Client Automation-managed application file or CONFIG.SYS file.) Thus, you must specify Y or N for an existing file to be distributed. Y = The file will be distributed if one or more of the following conditions is met: The file has not been distributed before. The date, time, or size of the file has changed. The file was re-promoted. N = The file will be distributed if: The file has not been distributed before. Or, The file was re-promoted. E = The file will be distributed only if it does not currently exist on the desktop. M = Reserved for Client Automation maintenance files. During the last step of the agent connect, the file is moved from the EDMDATA subdirectory to its proper location. Example: can be used to update EDMGETRS.EXE. S = This file is distributed if the file size has	Y	1 V

	changed. T = The file is distributed if the file time has changed. D = File is distributed if the file date has changed. L = Forces the lock method to be used. The file will be automatically queued up to be deployed on restart, and the reboot flag will be set. If the L flag is not specified, and the file to be deployed is currently in use (locked) the lock method is automatically used anyway. What the L flag does is force the lock method to be always used to deploy this file, whether or not it is currently in use (locked). The L flag will also skip decompression and dispatch the open error method (RADLKM). This is used for system dlls that must be replaced.		
	Note that for the file to be verified during subsequent connects, you must also set this variable to Y. U = File is distributed only if the date and time on the CONFIGURATION SERVER is newer than the date and time on the desktop. R = Resource Stamp. For internal use only. V = (Windows only) Used in when normal date/time stamp check cannot be used. Usually when a .dll or .exe is involved and the date and time stamp are changing frequently. Radconct.exe checks the internal version number of the target file and compares it with the one that Client Automation has. If the incoming file has a higher internal version number than it will be placed on the desktop over the existing file. P = (Windows only) Set during packaging, enforces Windows System File Protection (SFP). If the verification is successful, the resource is in the desired state and no further verification is necessary. If the verification fails, Client Automation checks to see if System File Protection protects the file. If the file is protected, it will not be deployed. If the ZRSCVRFY attribute is set to P and deployment of this resource and the service are marked as failed due to SFP processing. If you do not enforce SFP, the file is marked as having passed verification.		
ZUPDATE	Client Instance Update Method		50

	If an object (resource instance) has been updated by an administrator, this method will be executed.		M

HTTP Proxy (HTTP) Class

File:	PRIMARY	Description:	
Domain:	SECURITY		
Class:	HTTP		
Icon:		Can connect to:	

HTTP Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
HTTPADDR	WEB Server Host		255 V
HTTPLOG	HTTP Transaction Log [Y/N]	N	1 V
HTTPNVDS	Using Novadigm HTTP Server	N	1 V
HTTPPORT	Web Server Port	80	4 V
HTTPPROXO	Use Web server if proxy fails	N	1 V
HTTPPROXP	Proxy Server Port	80	4 V
HTTPPROXY	Proxy Server Host		255 V
HTTPVDIR	Web Server Virtual Directory	/RADIADB	255 V
ZPCPROTL	Protocol [HTTP]		4 V
ZSTGOPT	Enable Staging	1	1 V

IBM AIX Packages (AIX) Class

File:	PRIMARY	Description:	AIX operating systems only. This class is required by the Native Packaging Feature of the Batch Publisher.
Domain:	SECURITY		
Class:	AIX		
Icon:		Can connect to:	

AIX Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ALWAYS	Connect To Class connections determine the path of resolution of an agent's desired state. An _ALWAYS_ class connection is always resolved.		50 A
CMPMD5	Compressed File MD5 Signature	_UNDEF_	32 V
COMMIT	Commit Update [Y/N]	N	1 V
COREQ	AIX Package Corequisites		255 V
FILEMD5	File MD5 Signature	_UNDEF_	32 V
FORCE	Force (overwrite) installation [Y/N]	N	1 V
IFREQ	AIX Package If-Requisites		255 V
INSTREQ	AIX Package Installed Requisites		255 V
LOCATION	Location		255 V
NAME	Friendly Name	Defaults	80 V
OBJGUID	Instance GUID Object Global Unique Identification – <i>for future use.</i>	_UNDEF_	32 V
PKGDESC	Package Description		80

			V
PREREQ	AIX Package Prerequisites		255 V
RELYONDB	Verify Error Code? [Y/N]		1 V
ZCMPSIZE	Compressed File Size From promotion process if the compressed file option was selected. The size of the compressed file.		11 V
ZCREATE	Method to Install Resource Sentient-supplied method or fully qualified, user-defined method.	hide nvdkit method	60 M
ZDELETE	Method to De-Install resource	hide nvdkit method	50 M
ZFILEUPD	Client File Update/Add Method		50 M
ZINIT	Resource Initialization Method		50 M
ZOPENERR	Client Method on File Open Error Sentient-supplied method or a fully qualified, user-defined method.		50 M
ZPERGUID	File Group Name		32 V
ZPERUID	File Owner Name		32 V
ZREPAIR	Method to Repair Resource	hide nvdkit method	60 M
ZRSCCFIL	AIX Package File Name Specific file to be distributed. A generic pattern (a wildcard character, such as *) or an explicit filename.	_BASE _INSTANCE_	255 V
ZRSCCKPT	Resource Checkpoint		1 V
ZRSCCONF	Confirm File Download [Y/N]	Y	1 V
ZRSCCRC	Resource CRC		8 V
ZRSCCSTA	Client File Status	999	3

			V
ZRSCDATE	Resource Date Stamp – From Promote		8 V
ZRSCFLEV	Filesets + Levels		255 V
ZRSCFSET	Filesets in the Package		255 V
ZRSCMO	Mandatory/Optional on client [M/O]	M	1 V
ZRSCNAME	AIX Package Name	_BASE _INSTANCE_	255 V
ZRSCPADM	Admin ID		30 V
ZRSCPRI	Create Sequence Priority	50	2 V
ZRSCRASH	File Permissions		4 V
ZRSCRSTR	Restart [Y/N]	Y	3 V
ZRSCSIZE	Resource Size – From Promote		11 V
ZRSCSRC	Resource Source, i.e. Publisher		20 V
ZRSCSTYP	Server File Type [BINARY/TEXT]	BINARY	8 V
ZRSCSVRB	Version Skip Rebuild Flag [Y/N]		1 V
ZRSCTIME	Resource Time Stamp – From Promote		8 V
ZRSCVERS	Resource Version – From Promote		12 V
ZRSCVRFY	Verify Resource File on Connect	Y	8 V
ZSTATUS	Client Status	999	3 V
ZUPDATE	Client Instance Update Method	hide nvdkit method	60 M

ZVERIFY	Method to Verify Resource	hide nvdkit method	40 M

Install Options (INSTALL) Class

File:	PRIMARY	Description:	Defining an instance of the INSTALL Class lets you activate and configure the presentation of dialog panels which are used for the purposes of personalizing the delivery of an application along with its associated components.
Domain:	SECURITY		
Class:	INSTALL		
Icon:		Can connect to:	

INSTALL Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
APPNAME	Application Name		30 V
AUDIT	Search for Application File [Y/N]	N	1 V
COMP1SEL	Component 1 Selected [Y/N]	Y	1 V
COMP1TXT	Component 1 Description		20 V
COMP2SEL	Component 2 Selected [Y/N]	Y	1 V
COMP2TXT	Component 2 Description		20 V
COMP3SEL	Component 3 Selected [Y/N]	Y	1 V
COMP3TXT	Component 3 Description		20 V
COMP4SEL	Component 4 Selected [Y/N]	Y	1 V
COMP4TXT	Component 4 Description		20 V
COMPPIC	Components Picture (bmp)	&(WELCPIC)	12 V

COPYRITE	Copyright Text	@Copyright 1998	30 V
EULAPIC	EULA Picture (bmp)	&(WELCPIC)	12 V
EULATEXT	End User License Agreement Text	This software is protected by copyright laws. By accepting, you are agreeing with all terms included in the original program documentation.	200 V
IDIRNAME	Install Directory Name		100 V
IDIRPIC	Install Directory Picture (bmp)	&(WELCPIC)	12
KEYFILE	Search for File Name		12 V
NAME	Friendly Name		80 V
OPTS1SEL	Option 1 Selected [Y/N]	Y	1 V
OPTS1TXT	Option 1 Description		20 V
OPTS2SEL	Option 2 Selected [Y/N]	Y	1 V
OPTS2TXT	Option 2 Description		20 V
OPTS3SEL	Option 3 Selected [Y/N]	Y	1 V
OPTS3TXT	Option 3 Description		20 V
OPTS4SEL	Option 4 Selected [Y/N]	Y	1 V
OPTS4TXT	Option 4 Description		20 V
OPTSPIC	Options Picture (bmp)	&(WELCPIC)	12 V
SCANDRVS	Search Drives		32 V

SHOWCOMP	Show Components [Y/N]	N	1 V
SHOWEULA	Show End User License Agreement [Y/N]	Y	1 V
SHOWIDIR	Show Install Directory [Y/N]	N	1 V
SHOWOPTS	Show Options [Y/N]	N	1 V
SHOWTYPE	Show Install Type [Y/N]	N	1 V
SHOWUINF	Show User Info [Y/N]	N	1 V
SHOWWELC	Show Welcome (N disables all) [Y/N]	Y	1 V
TYPE1BTN	Type 1 Button Caption	Typical	10 V
TYPE1TXT	Type 1 Description	Program will be installed with the most common options. Recommended for most users.	100 V
TYPE2BTN	Type 2 Button Caption	Custom	10 V
TYPE2TXT	Type 2 Description	You may choose the options you want to install. Recommended for advanced users.	100 V
TYPE3BTN	Type 3 Button Caption		10 V
TYPE3TXT	Type 3 Description		100 V
TYPEPIC	Install Type Picture (bmp)	&(WELCPIC)	12 V
TYPESEL	Type Selection	Typical	10 V
UINF1LBL	User Info 1 Label		12 V

UINF1TXT	User Info - Text 1		30 V
UINF2LBL	User Info 2 Label		12 V
UINF2TXT	User Info - Text 2		30 V
UINF3LBL	User Info 3 Label		12 V
UINF3TXT	User Info - Text 3		30 V
UINFNAME	User Info - Name		30 V
UINFPIC	User Info Picture (bmp)	&(WELCPIC)	12 V
WELCPIC	Welcome Picture (bmp)	instpic.bmp	12 V
ZOBJDATA	Search Operation is Scan [SCN]	SCN	3 V
ZRSCCDIR	Search Directory	*	255 V
ZRSCCDRV	Search Drive	&(ZMASTER.ZWINDRV)	19 V
ZRSCSTYP	Search File Type [BINARY]	BINARY	6 V

Linux RPM Packages (RPM) Class

File:	PRIMARY	Description:	Linux operating systems only. This class is required by the Native Packaging Feature of the Batch Publisher.
Domain:	SECURITY		
Class:	RPM		
Icon:		Can connect to:	

RPM Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type

ALWAYS	Connect To Class connections determine the path of resolution of an agent's desired state. An _ALWAYS_ class connection is always resolved.		50 A
CMPMD5	Compressed File MD5 Signature	_UNDEF_	32 V
CONTENTS	Required Packages Included in Tar		255 V
FILEMD5	File MD5 Signature	_UNDEF_	32 V
INSTOPTS	Package Install Options		60 V
LOCATION	Location		255 V
NAME	Friendly Name		80 V
OBJGUID	Instance GUID Object Global Unique Identification – <i>for future use.</i>	_UNDEF_	32 V
PKGARCH	RPM Package Architecture		32 V
PKGEPOCH	RPM Package EPOCH		32 V
PKGNAME	RPM Package Name		32 V
PKGREL	RPM Package Release		32 V
PKGSUMM	RPM Package Summary		100 V
PKGVER	RPM Package Version		32 V
RELYONDB	Verify Error Code? [Y/N]		1 V
REQCMDS	Requires Commands		255 V
REQLIBS	Requires Shared Libraries		255 V
REQPKGS	Requires Package		255 V
VRFYOPTS	Package Verify Options		60 V
ZCMPSIZE	Compressed File Size From promotion process if the compressed file option was selected. The size of the compressed file.		11 V
ZCREATE	Method to Install Resource Sentient-supplied method or fully qualified, user-defined method.	hide nvdkit method	60 M
ZDELETE	Method to De-Install resource	hide nvdkit method	50 M
ZFILEUPD	Client File Update/Add Method		50 M

ZINIT	Resource Initialization Method		50 M
ZOPENERR	Client Method on File Open Error Sentient-supplied method or a fully qualified, user-defined method.	RADLKM	50 M
ZPERGUID	File Group Name		32 V
ZPERUID	File Owner Name		32 V
ZREPAIR	Method to Repair Resource	hide nvdkit method	60 M
ZRSCCFIL	RPM Package File Name Specific file to be distributed. A generic pattern (a wildcard character, such as *) or an explicit filename.	_BASE_ _INSTANCE_	255 V
ZRSCCKPT	Resource Checkpoint		1 V
ZRSCCONF	Confirm File Download [Y/N]	Y	1 V
ZRSCCRC	Resource CRC		8 V
ZRSCCSTA	Client File Status	999	3 V
ZRSCDATE	Resource Date Stamp – From Promote		8 V
ZRSCMO	Mandatory/Optional on client [M/O]	M	1 V
ZRSCNAME	RPM Package	_BASE_ _INSTANCE_	255 V
ZRSCPADM	Admin ID		30 V
ZRSCPRI	Create Sequence Priority	50	2 V
ZRSCRASH	File Permissions		4 V
ZRSCRSTR	Restart [Y/N]	Y	3 V
ZRSCSIZE	Resource Size – From Promote		11 V
ZRSCSRC	Resource Source, that is Publisher		20 V
ZRSCSTYP	Server File Type [BINARY/TEXT]	BINARY	8 V
ZRSCSVRB	Version Skip Rebuild Flag [Y/N]		1 V
ZRSCTIME	Resource Time Stamp – From Promote		8 V
ZRSCVERS	Resource Version – From Promote		12 V
ZRSCVRFY	Verify Resource File on Connect	Y	8 V
ZSTATUS	Client Status	999	3 V

ZUPDATE	Client Instance Update Method	hide nvdkit method	60 M
ZVERIFY	Method to Verify Resource	hide nvdkit method	40 M

Mac File Resources (MACFILE) Class

File:	PRIMARY	Description:	Mac operating systems only. An instance of the MACFILE Class defines a file, or a group of files, to include in an application service. An instance of this class lets you specify such characteristics about the file as the file's name, size, and date stamp.
Domain:	SECURITY		
Class:	MACFILE		
Icon:		Can connect to:	

MACFILE Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ALWAYS	Connect to Class connections determine the path of resolution of an agent's desired state. An _ALWAYS_ class connection is always resolved.		50 C
LINKDEST	Symbolic Link Destination		80 V
LOCATION	Location		255 V
NAME	Friendly Name	Defaults	80 V
OBJGUID	Instance GUID Object Global Unique Identification – <i>for future use</i> .	_UNDEF_	32 V
SIGTYPE	File Signature Type		3 V
ZCMPKEY	Compression Key		80 V
ZCMPSIZE	Compressed File Size From promotion process if the compressed file option was selected. The size of the compressed file.		11 V

ZCONTEXT	Component Context		8 V
ZCREATE	Method to Install Resource Sentient-supplied method or fully qualified, user-defined method.		50 M
ZDELETE	Method to De-install Resource Sentient-supplied method or fully qualified, user-defined method.	RADREMF	50 M
ZFILEUPD	Client File Update/Add Method If a file has been updated by an administrator, this method will be executed. Sentient-supplied method or fully qualified, user-defined method.		50 M
ZINIT	Resource Initialization Method A fully qualified, user-defined method.		50 M
ZOBJPTCH	Perform Patching [Y/N]	Y	1 V
ZOPENERR	Client Method on File Open Error Sentient-supplied method or a fully qualified, user-defined method.		50 M
ZPERGID	File Group Name		32 V
ZPERUID	File Owner Name		32 V
ZREPAIR	Client Instance Repair Method		50 V
ZRSCCFIL	Resource File Name Specific file to be distributed. A generic pattern (a wildcard character, such as *) or an explicit filename.	_BASE_ INSTANCE_	255 V
ZRSCCKPT	Resource Checkpoint		1 V
ZRSCCONF	Confirm File Download [Y/N] Y enables you to confirm that the resource was downloaded.	Y	1 V
ZRSCCRC	Resource CRC From the promotion process if the CRC option was selected. Eight-character calculated hexadecimal value.		8 V
ZRSCCSTA	Client File Status	999	3 V

ZRSCDATE	Resource Date Stamp Date stamp on resource file from promote process.		8 V
ZRSCMO	Mandatory/Optional [M/O]	M	1 V
ZRSCNAME	Resource Name Name or description of resource.	&ZRSCCFIL	30 V
ZRSCPADM	Admin ID		30 V
ZRSCPRI	Create Sequence Priority Enables you to specify a priority level for this resource. If disk space on the client desktop is not available, resources will be downloaded based on the priority set here. 01 = Highest priority 10 = Default value 99 = Lowest priority	50	2 V
ZRSCRASH	File Permissions		4 V
ZRSCRSTR	Restart [Y/N/R] Y to restart. Set to R to bypass Windows file caching and write directly to disk (allowing checkpoint restart to operate correctly).	Y	3 V
ZRSCSIG	File Signature		32 V
ZRSCSIZE	Resource Size Byte count of resource file from promote process.		11 V
ZRSCSRC	Resource Source, that is Publisher		20 V
ZRSCSTYP	Server File Type [BINARY/TEXT]	BINARY	8 V
ZRSCSVRB	Version Skip Rebuild Flag[Y/N]		1 V
ZRSCTIME	Resource Time Stamp Time stamp on resource file from promote process.		8 V
ZRSCVERS	Resource Version - From Promote Version number of resource from promotion process.		12 V

ZRSCVRFY	Verify Resource File on Connect Verifies the file statistics of a resource file on the desktop during each agent connect process. If you are deploying versioned resources, this attribute is ignored. The rules for the E(xist) flag apply to both Client Automation- and non-Sentient distributed files. (For example, a Client Automation-managed application file or <code>CONFIG.SYS</code> file.) Thus, you must specify Y or N for an existing file to be distributed. Y = The file will be distributed if one or more of the following conditions is met: - The file has not been distributed before. - The date, time, or size of the file has changed. - The file was re-promoted. N = The file will be distributed if: - The file has not been distributed before, - Or, the file was re-promoted. E = The file will be distributed only if it does not currently exist on the desktop. M = Reserved for Client Automation maintenance files. During the last step of the connect, the file is moved from the <code>EDMDATA</code> subdirectory to its proper location. Example: can be used to update <code>EDMGETRS.EXE</code>. S = This file is distributed if the file size has changed. T = The file is distributed if the file time has changed. D = File is distributed if the file date has changed. L = Forces the lock method to be used. The file will be automatically queued up to be deployed on restart, and the reboot flag will be set. If the L flag is not specified, and the file to be deployed is currently in use (locked) the lock method is automatically used anyway. What the L flag does is force the lock method to be always used to deploy this file, whether or not it is currently in use (locked). The L flag will also skip decompression and dispatch the open error method (RADLKM). This is used for system dlls that must be replaced.	Y	8 V
	Note that for the file to be verified during subsequent connects, you must also set this variable to Y. U = File is distributed only if the date and time		

	on the Configuration Server is newer than the date and time on the desktop. R = Resource Stamp. For internal use only. V = (Windows only) Used in when normal date/time stamp check cannot be used. Usually when a .dll or .exe is involved and the date and time stamp are changing frequently. Radconct.exe checks the internal version number of the target file and compares it with the one that Client Automation has. If the incoming file has a higher internal version number, then it will be placed on the desktop over the existing file. P = (Windows only) Set during packaging, enforces Windows System File Protection (SFP). If the verification is successful, the resource is in the desired state and no further verification is necessary. If the verification fails, Client Automation checks to see if System File Protection protects the file. If the file is protected, it will not be deployed. If the ZRSCVRFY attribute is set to P and deployment of this resource and the service are marked as failed due to SFP processing. If you do not enforce SFP, the file is marked as having passed verification.		
ZSTATUS	Client Status	999	3 V
ZUPDATE	Client Instance Update Method		50 M

Mobile File Resources (RMMFILE) Class

File:	PRIMARY	Description:	Instances of this class are created when publishing files for Mobile Management using the Mobile Management Publisher.
Domain:	SECURITY		
Class:	RMMFILE		
Icon:		Can connect to:	

RMMFILE Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ALWAYS	Connect To		50

	Class connections determine the path of resolution of an agent's desired state. An <code>_ ALWAYS_</code> class connection is always resolved.		C
FVERSION	File Version	<code>_UNDEF_</code>	30 V
LOCATION	Location		255 V
NAME	Friendly Name	Defaults	80 V
OBJGUID	Instance GUID Object Global Unique Identification – <i>for future use.</i>	<code>_UNDEF_</code>	32 V
PVERSION	Promote Version	<code>_UNDEF_</code>	30 V
SIGTYPE	File Signature Type		3 V
UPROIMPT	User Prompt [Y/N]	N	1 V
ZAPPNAME	PDA Application Name		50 V
ZCMPKEY	Compression Key	<code>_UNDEF_</code>	80 V
ZCMPSIZE	Compressed File Size From promotion process if the compressed file option was selected. The size of the compressed file.		11 V
ZCONTEXT	Component Context		8 V
ZCREATE	Method to Install Resource Sentient-supplied method or fully qualified, user-defined method.		50 M
ZDELETE	Method to De-install Resource	RADREMF	50 M
ZFILEUPD	Client File Update/Add Method		50 M
ZINIT	Resource Initialization Method		50 M

ZOBJPTCH	Perform Patching [Y/N]	Y	1 V
ZOPENERR	Client Method on File Open Error Sentient-supplied method or a fully qualified, user-defined method.	RADLKM	50
ZPDAVERS	PDA Resource Internal Version		20 V
ZPUBVER	Publisher Version	_UNDEF_	8 V
ZREPAIR	Client Instance Repair Method		50 M
ZRSCCFIL	Resource File Name Specific file to be distributed. A generic pattern (a wildcard character, such as *) or an explicit filename.		255 V
ZRSCCKPT	Resource Checkpoint		1 V
ZRSCCONF	Confirm File Download [Y/N]	Y	1 V
ZRSCCPADM	Admin ID		30 V
ZRSCCRC	Resource CRC		8 V
ZRSCCSTA	Client file Status	999	3 V
ZRSCDATE	Resource Date Stamp – From Promote		8 V
ZRSCMMEM	PDS Member Name		8 V
ZRSCMO	Mandatory/Optional on Client [M/O]	M	1 V
ZRSCNAME	Resource Name	&ZRSCCFIL	30 V
ZRSCOBJN	Persistent Object Name		8 V
ZRSCPRI	Create Sequence Priority	50	2 V
ZRSCRASH	DOS File Attribute [R/A/S/H]		4

			V
ZRSCRSCR	Restart [Y/N]	N	3 V
ZRSCSIG	File Signature		32 V
ZRSCSIZE	Resource Size – From Promote		11 V
ZRSCSRC	Resource Source, i.e., Publisher		20 V
ZRSCSTYP	Server File Type [BINARY/TEXT]	BINARY	8 V
ZRSCSVRB	Version Skip Rebuild Flag [Y/N]		1 V
ZRSCTIME	Resource Time Stamp – From Promote		8 V
ZRSCVERS	Resource Version – From Promote		12 V
ZRSCVRFY	Verify Resource File on Connect	Y	8 V
ZUPDATE	Client Instance Update Method		50 M

MSI Basic Resources (MSIBASIC) Class

File:	PRIMARY	Description:	Packages created with the Extensions for Windows Installer have a connection to the MSIBASIC class in the CSDB. When the New Application Wizard is used to create a service from an application with one of these connections, you have the opportunity to define the MSI basic behavior methods.
Domain:	SECURITY		
Class:	MSIBASIC		
Icon:		Can connect to:	

MSIBASIC Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ALWAYS	Connect To		50

	Class connections determine the path of resolution of an agent's desired state. An _ALWAYS_ class connection is always resolved.		C
BCREATE	Method Before Install		50 M
BDELETE	Method Before De-Install		50 M
BREPAIR	Method Before Repair		50 M
BUPDATE	Method Before Update		50 M
BVERIFY	Method Before Verify		50 M
FVERSION	File Version	_UNDEF_	30 V
LOCATION	Location		255 V
NAME	Friendly Name		80 V
PRODGUID	MSI Product GUID		50 V
PVERSION	Promote Version	_UNDEF_	30 V
SIGTYPE	File Signature Type		3 V
ZCMPSIZE	Compressed File Size From promotion process if the compressed file option was selected. The size of the compressed file.		11 V
ZCONTEXT	Component Context		8 V
ZCREATE	Method to Install Resource Sentient-supplied method or fully qualified, user-defined method.		50 M
ZDELETE	Method to De-install Resource	RADREMF	50 M
ZINIT	Resource Initialization Method		50 M

ZOBJPTCH	Perform Patching [Y/N]	Y	1 V
ZOPENERR	Client Method on File Open Error Sentient-supplied method or a fully qualified, user-defined method.		50 M
ZREPAIR	Client Instance Repair Method		50 M
ZRSCCFIL	Resource File Name Specific file to be distributed. A generic pattern (a wildcard character, such as *) or an explicit file name.		255 V
ZRSCCRC	Resource CRC		8 V
ZRSCCSTA	Client file Status	999	3 V
ZRSCDATE	Resource Date Stamp – From Promote		8 V
ZRSCMO	Mandatory/Optional on Client [M/O]	M	1 V
ZRSCPRI	Create Sequence Priority	50	2 V
ZRSCRASH	DOS File Attribute [R/A/S/H]		4 V
ZRSCRSCR	Restart [Y/N]	N	3 V
ZRSCSIG	File Signature		32 V
ZRSCSIZE	Resource Size – From Promote		11 V
ZRSCSRC	Resource Source, i.e., Publisher		20 V
ZRSCSTYP	Server File Type [BINARY/TEXT]	BINARY	8 V
ZRSCTIME	Resource Time Stamp – From Promote		8 V
ZRSCVRFY	Verify Resource File on Connect	Y	8 V
ZUPDATE	Client Instance Update Method		50

			M
ZVERIFY	Client Instance Verify Method		50 M

MSI Features (MSIFEATS) Class

File:	PRIMARY	Description:	An instance of the MSIFEATS Class defines MSI feature states.
Domain:	SECURITY		
Class:	MSIFEATS		
Icon:		Can connect to:	

MSIFEATS Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ASTATES	Allowed States		100 V
DISABLED	Disabled [Y/N]		1 V
DSTATE	Default State		10 V
FEATID	Feature ID		38 V
FEATNAME	Feature Name		50 V
FEATSIZE	Feature Size		12 V
HIDE	Hide Features in CP A/R Dialog		1 V
ISTATE	Install State		10 V
OBJGUID	Instance GUID Object Global Unique Identification – <i>for future use.</i>	_UNDEF_	32 V
PARENTID	Parent Feature ID		38 V
SEQNO	Sequence Number		4 V

MSI Properties (MSIPROPS) Class

File:	PRIMARY	Description:	The MSIPROPS Class stores MSI public properties and their values.
Domain:	SECURITY		
Class:	MSIPROPS		
Icon:		Can connect to:	

MSIPROPS Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
DEFAULT	Default Property Value		255 V
DISABLED	Disabled [Y/N]		1 V
PROPNAME	Property Name		255 V
VALUE	Property Value		255 V
VERIFY	Verify Property Value [Y/N]	N	1 V

MSI Resources (MSI) Class

File:	PRIMARY	Description:	Instances in this class are used to define MSI resources.
Domain:	SECURITY		
Class:	MSI		
Icon:		Can connect to:	

MSI Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ALWAYS	Connect to Class connections determine the path of resolution of an agent's desired state. An _ALWAYS_ class connection is always resolved.		50 C
DATAURL	Administrative Control Point		255 V
LOCATION	Location		255 V

MSIDRVR	Use Local AIP [Y/N]		1 V
NAME	Friendly Name	Defaults	80 V
OBJGUID	Instance GUID Object Global Unique Identification – <i>for future use.</i>	_UNDEF_	32 V
ORGNAME	Organization Name		100 V
PRODGUID	MSI Product GUID	_UNDEF_	50 V
PRODVER	Product Version	_UNDEF_	10 V
SALVAGE	Salvage Cache Files[U/M/B/N] Salvage existing Windows Installer-enabled content. U – Unmanaged application salvage. Set to U if you want to salvage content that Sentient is not currently managing. M – Managed application salvage. Set to M to salvage content that Sentient is currently managing. B – Both. Set to B to salvage both managed and unmanaged applications. This is useful when you have clients with resources both managed and unmanaged by Sentient. N – No content is salvaged.	N	1 V
SAVETEMP	Preserve Internet Cache [Y/N]	N	1 V
SEQNO	Feature Sequence Number		3 V
SHOWINCP	Show in CP Add/Remove	Y	1 V
SIGTYPE	File Signature Type		3 V
USERCNTL	User Controls MSI Features [Y/N]		1 V
ZCMPKEY	Compression Key		80 V
ZCMPSIZE	Compressed File Size		11 V

ZCONTEXT	Component Context [M/U]		8 V
ZCREATE	Service Installation Method Sentient-supplied method or fully qualified, user-defined method.		50 M
ZDELETE	Service Delete Method Sentient-supplied method or fully qualified, user-defined method.	_NONE_	50 M
ZFILEUPD	Client File Update/Add Method If a file has been updated by an administrator, this method will be executed. Sentient-supplied method or fully qualified, user-defined method.		50 M
ZINIT	Service Initialization Method A fully qualified, user-defined method.		50 M
ZOBJPTCH	Perform Patching [Y/N]	Y	1 V
ZOPENERR	Client Method on File Open Error Sentient-supplied method or a fully qualified, user-defined method.	_UNDEF_	50 M
ZPUBVER	Publisher Version		8 V
ZREPAIR	Service Repair Method		50 M
ZRSCCFIL	Resource File Name Specific file to be distributed. A generic pattern (a wildcard character, such as *) or an explicit filename.		255 V
ZRSCCKPT	Resource Checkpoint		1 V
ZRSCCONF	Confirm File Download [Y/N] Y enables you to confirm that the resource was downloaded.	Y	1 V
ZRSCCRC	Resource CRC From the promotion process if the CRC option was selected. Eight-character calculated hexadecimal value.		8 V
ZRSCCSTA	Client File Status	999	3 V
ZRSCDATE	Resource Date Stamp Date stamp on resource file from promote		8 V

ZRSCMEM	PDS Member Name		8 V
ZRSCMO	Mandatory/Optional [M/O]	M	1 V
ZRSCNAME	Resource Name Name or description of resource.	&ZRSCCFIL	30 V
ZRSCOBJN	Persistent Object Name		8 V
ZRSCPADM	Admin ID		30 V
ZRSCPRI	Create Sequence Priority Enables you to specify a priority level for this resource. If disk space on the client desktop is not available, resources will be downloaded based on the priority set here. 01 = Highest priority 10 = Default value 99 = Lowest priority	50	2 V
ZRSCRASH	DOS File Attribute [R/A/S/H]	N	4 V
ZRSCRSTR	Restart [Y/N]		3 V
ZRSCSIG	File Signature		32 V
ZRSCSIZE	Resource Size Byte count of resource file from promote process.		11 V
ZRSCSRC	Resource Source, that is Publisher		20 V
ZRSCSTYP	Server File Type [BINARY/TEXT]	BINARY	8 V
ZRSCSVRB	Version Skip Rebuild Flag [Y/N]		1 V
ZRSCTIME	Resource Time Stamp Time stamp on resource file from promote process.		8 V
ZRSCVERS	Resource Version - From Promote Version number of resource from promotion process.		12 V

ZRSCVRFY	Verify Resource File on Connect Verifies the file statistics of a resource file on the desktop during each agent connect process. If you are deploying versioned resources, this attribute is ignored. The rules for the E (exist) flag apply to both Sentient- and non-Sentient distributed files. (For example, a Client Automation-managed application file or <code>CONFIG.SYS</code> file.) Thus, you must specify Y or N for an existing file to be distributed. Y = The file will be distributed if one or more of the following conditions is met: - The file has not been distributed before. - The date, time, or size of the file has changed. - The file was re-promoted. N = The file will be distributed if: The file has not been distributed before, Or, the file was re-promoted. E = The file will be distributed only if it does not currently exist on the desktop. M = Reserved for Client Automation maintenance files. During the last step of the connect, the file is moved from the <code>EDMDATA</code> subdirectory to its proper location. Example: can be used to update <code>EDMGETRS.EXE</code>. S = This file is distributed if the file size has changed. T = The file is distributed if the file time has changed. D = File is distributed if the file date has changed. L = Forces the lock method to be used. The file will be automatically queued up to be deployed on restart, and the reboot flag will be set. If the L flag is not specified, and the file to be deployed is currently in use (locked) the lock method is automatically used anyway. What the L flag does is force the lock method to be always used to deploy this file, whether or not it is currently in use (locked). The L flag will also skip decompression and dispatch the open error method (RADLKM). This is used for system dlls that must be replaced.		8 V
	Note that for the file to be verified during subsequent connects, you must also set this variable to Y. U = File is distributed only if the date and time		

	on the Configuration Server is newer than the date and time on the desktop. R = Resource Stamp. For internal use only. V = (Windows only) Used in when normal date/time stamp check cannot be used. Usually when a .dll or .exe is involved and the date and time stamp are changing frequently. Radconct.exe checks the internal version number of the target file and compares it with the one that Client Automation has. If the incoming file has a higher internal version number than it will be placed on the desktop over the existing file. P = (Windows only) Set during packaging, enforces Windows System File Protection (SFP). If the verification is successful, the resource is in the desired state and no further verification is necessary. If the verification fails, Client Automation checks to see if System File Protection protects the file. If the file is protected, it will not be deployed. If the ZRSCVRFY attribute is set to P and deployment of this resource and the service are marked as failed due to SFP processing. If you do not enforce SFP, the file is marked as having passed verification.		
ZUPDATE	Service Update Method		50 M
ZVERIFY	Service Verify Method		50 M

Panel Services (PANEL) Class

File:	PRIMARY	Description:	The PANEL Class contains the Client Automation panel objects used during Client Automation installation dialog processing. This class contains shipping dialog panel examples. Publish all custom dialog panels to this class.
Domain:	SECURITY		
Class:	PANEL		
Icon:		Can connect to:	

PANEL Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ALWAYS	Connect to		50

	Class connections determine the path of resolution of an agent's desired state. An <code>_ ALWAYS _</code> class connection is always resolved.		C
CMPMD5	Compressed File MD5 Signature	<code>_UNDEF_</code>	32 V
FILEMD5	File MD5 Signature	<code>_UNDEF_</code>	32 V
NAME	Friendly Name	Defaults	80 V
OBJGUID	Instance GUID Object Global Unique Identification – <i>for future use.</i>	<code>_UNDEF_</code>	32 V
ZCMPKEY	Compression Key Not used.		80 V
ZCMPSIZE	Compressed File Size From promotion process if the compressed file option was selected. The size of the compressed file.		11 V
ZCREATE	Install Behavior Sentient-supplied method or fully qualified, user-defined method.		50 M
ZDELETE	Method to De-install Behavior Sentient-supplied method or fully qualified, user-defined method.	RADREMF	50 M
ZFILEUPD	Client File Update/Add Method If a file has been updated by an administrator, this method will be executed. Sentient-supplied method or fully qualified, user-defined method.		50 M
ZINIT	Resource Initialization Method Sentient-supplied method or fully qualified, user-defined method.		50 M
ZOPENERR	Client Method on File Open Error Sentient-supplied method or fully qualified, user-defined method.	RADLKM	50 M
ZRSCCFIL	Panel Name Specific file to be distributed. A generic pattern (a wildcard character, such as <code>*</code>) or an explicit filename.		255 V
ZRSCCKPT	Resource Checkpoint Not used. Replaced by ZRSCRSTR.		1 V

ZRSCCONF	Confirm File Download [Y/N] Y enables you to confirm that the resource was downloaded.	N	1 V
ZRSCCRC	Resource CRC From the promotion process if the CRC option was selected. Eight-character calculated hexadecimal value.		8 V
ZRSCCSTA	Client File Status Resource status code. 000 = Successfully downloaded 001 = Data can be found in EDMDATA directory 004 = Resource with no data 008 = Could not create the file 009 = No disk space for file (ZRSCMO=O) 010 = ZDLIMIT was exceeded (ZRSCMO=O) 011 = No data in Sentient for resource 014 = No disk space (ZRSCMO=M) 015 = Time-out waiting for method 036 = A method was set ZBRC>7 999 = No work done for resource	999	3 V
ZRSCDATE	Panel Date Stamp Date stamp on resource file from promote process.		8 V
ZRSCMDAT	Manager Stores Data [Y/N]	Y	1 V
ZRSCMMEM	PDS Member Name PDS where resource actually resides. PDS name where resource resides in Manager (MVS only).		8 V
ZRSCMO	Mandatory/Optional [M/O]	M	1 V
ZRSCNAME	Panel Name Name or description of resource. Valid name of resource to distribute to Agent.	&ZRSCCFIL	30 V
ZRSCOBJN	Persistent Object Name The name of the in-storage object. Valid only when the ZLOCMGR location is STORAGE.		8 V
ZRSCPRI	Create Sequence Priority Enables you to specify a priority level for this resource. If disk space on the client desktop is not available, resources will be downloaded based on the priority set here. 01 = Highest priority		2 V

	10 = Default value 99 = Lowest priority		
ZRSCRASH	DOS File Attribute Enables you to specify file attributes for a resource. Read-only Archive System and/or Hidden		4 V
ZRSCRSTR	Restart [Y/N] You can either restart a download process at the point it was interrupted (Yes), or resend the entire object (No).	N	3 V
ZRSCSIZE	Panel Size Byte count of resource file from promote process.		11 V
ZRSCSRC	Resource Source, that is Publisher		20 V
ZRSCSRC	Resource Source		1 V
ZRSCSTYP	Server File Type [BINARY/TEXT] File format for file download: Binary, Text, or Object.	BINARY	8 V
ZRSCSVRB	Version Skip Rebuild Flag [Y/N] Enables you to skip the reactivation of version resources if the version is already active. Y = Skip the rebuilding of version resources. N = Rebuild version resources.	Y	1 V
ZRSCTIME	Panel Time Stamp Time stamp on resource file from promote process.		8 V
ZRSCVERS	Resource Version - From Promote Version number of resource from promotion process.		12 V
ZRSCVRFY	Verify Resource File on Connect Verifies the file statistics of a resource file on the desktop during each agent connect process. If you are deploying versioned resources, this attribute is ignored. The rules for the E (exist) flag apply to both Sentient- and non-Sentient distributed files. (For example, a Sentient-managed application file or CONFIG.SYS file.)	Y	1 V

	Thus, you must specify Y or N for an existing file to be distributed. Y = The file will be distributed if one or more of the following conditions is met: - The file has not been distributed before. - The date, time, or size of the file has changed. - The file was re-promoted. N = The file will be distributed if: The file was re-promoted. E = The file will be distributed only if it does not currently exist on the desktop. M = Reserved for Client Automation maintenance files. During the last step of the agent connect, the file is moved from the <code>EDMDATA</code> subdirectory to its proper location. Example: can be used to update <code>EDMGETRS.EXE</code>. S = This file is distributed if the file size has changed. T = The file is distributed if the file time has changed. D = File is distributed if the file date has changed. L = Forces the lock method to be used. The file will be automatically queued up to be deployed on restart, and the reboot flag will be set. If the L flag is not specified, and the file to be deployed is currently in use (locked) the lock method is automatically used anyway. What the L flag does is force the lock method to be always used to deploy this file, whether or not it is currently in use (locked). The L flag will also skip decompression and dispatch the open error method (RADLKM). This is used for system dlls that must be replaced.		
	Note that for the file to be verified during subsequent connects, you must also set this variable to Y. U = File is distributed only if the date and time on the Configuration Server is newer than the date and time on the desktop. R = Resource Stamp. For internal use only. Automation has. If the incoming file has a higher internal version number than it will be placed on the desktop over the existing file. P = (Windows only) Set during packaging, enforces Windows System File Protection (SFP). If the verification is successful, the resource is in the desired state and no further		

	verification is necessary. If the verification fails, Client Automation checks to see if System File Protection protects the file. If the file is protected, it will not be deployed. If the ZRSCVRFY attribute is set to P and deployment of this resource and the service are marked as failed due to SFP processing. If you do not enforce SFP, the file is marked as having passed verification.		
ZUPDATE	Client Instance Update Method If an object (resource instance) has been updated by an administrator, this method will be executed.		50 M
ZUSERID	Admin ID		30 V

Path (PATH) Class

File:	PRIMARY	Description:	This class stores the drive and directory required to install a resource. Packages can be relocated by updating instances of this class.
Domain:	SECURITY		
Class:	PATH		
Icon:		Can connect to:	CATALOG, ZSERVICE

PATH Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
DIR	Directory Stores the directory name where the application is installed. The directory name could be selected by the administrator, or it could be selected by the client during the dialog sessions that precede an application install. The value can be substituted from an agent object, or it can be a fully qualified name.		255 V
DRIVE	Drive Stores the drive where the application would be installed. The drive name could be selected by the administrator or it could be selected by the client during the dialog sessions that precede an application install. The value can be substituted from an agent object, or it can be a valid drive name on the Agent machine.		20 V

NAME	Friendly Name		255 V
OBJGUID Path (PATH) Class	Instance GUID Object Global Unique Identification – <i>for future use.</i>	_UNDEF_	32 V
ZRSCMO	Mandatory/Optional [M/O]	M	1 V
ZRSCPRI	Priority Stores the Create priority. The path object should be one of the first objects built on the Agent during an application installation so that all other files, desktop instances, can refer to it. The lower priority instances, or classes, are installed first. Numeric value between 1 and 99.	5	2 V

Registry Resources (REGISTRY) Class

File:	PRIMARY	Description:	An instance of the REGISTRY Class defines a registry resource that is part of a PACKAGE.
Domain:	SECURITY		
Class:	REGISTRY		
Icon:		Can connect to:	

REGISTRY Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ALWAYS	Connect to Class connections determine the path of resolution of an agent's desired state. An _ALWAYS_ class connection is always resolved.		50 C
DELENUM	Delete Device Key under Enum\Root When set to Y, will delete the device key under ENUM which otherwise will not be deleted due to the appearance of two new values after reboot; values that are not managed by Client Automation. For example, [hkey_local_machine\system\currentcontrolset\enum\root\legacy_beep\0000\control] key will be deleted when delenum=y. This prevents "new hardware found" pop up panels when the	N	1 V

	service is deleted.		
DELNTSCM	Delete SCM Keys/Values [Y/N]	Y	2 V
DEPTH	Registry Depth	&(ZCURRENT. ZCURDEPT)	20 V
ENUMPROC	Process Control Key under Enum\Root	N	1 V
LOCATION	Location		255 V
NAME	Friendly Name		80 V
OBJGUID	Instance GUID Object Global Unique Identification – <i>for future use.</i>	_UNDEF_	32 V
SIGTYPE	File Signature Type		3 V
ZCMPKEY	Compression Key		80 V
ZCMPSIZE	Compressed File Size From promotion process if the compressed file option was selected. The size of the compressed file.		11 V
ZCONTEXT	Component Context		8 V
ZCREATE	Method to Install Resource Sentient-supplied method or fully qualified, user-defined method.	RADREG32	50 M
ZDELETE	Method to De-install Resource Sentient-supplied method or fully qualified, user-defined method.	RADREG32	50 M
ZFILEUPD	Client File Update/Add Method If a file has been updated by an administrator, this method will be executed. Sentient-supplied method or fully qualified, user-defined method.		50 M
ZINIT	Resource Initialization Method A fully qualified, user-defined method.		50 M
ZOBJPRI	Priority [0000-99]	55	2 V
ZOPENERR	Client Method on File Open Error		50

	Sentient-supplied method or a fully qualified, user-defined method.		M
ZPUBVER	Publisher Version	_UNDEF_	8 V
ZRSCCFIL	Resource File Name Specific file to be distributed. A generic pattern (a wildcard character, such as *) or an explicit filename.		255 V
ZRSCCKPT	Resource Checkpoint		1 V
ZRSCCONF	Confirm File Download [Y/N] Y enables you to confirm that the resource was downloaded.	Y	1 V
ZRSCCRC	Resource CRC From the promotion process if the CRC option was selected. Eight-character calculated hexadecimal value.		8 V
ZRSCCSTA	Client File Status	999	3 V
ZRSCDATE	Resource Date Stamp Date stamp on resource file from promote process.		8 V
ZRSCMO	Mandatory/Optional [M/O]		1 V
ZRSCNAME	Resource Name Name or description of resource.	&ZRSCFIL	30 V
ZRSCPADM	Admin ID		30 V
ZRSCRASH	DOS File Attribute [R/A/S/H]		4 V
ZRSCRSTR	Restart [Y/N]	N	1 V
ZRSCSIG	File Signature		32 v
ZRSCSIZE	Resource Size – From Promote Byte count of resource file from promote process.		11 V
ZRSCSRC	Resource Source, that is Publisher		20 V

ZRSCSTYP	File Type [BINARY/TEXT/EDMFILE]	EDMFILE	8 V
ZRSCTIME	Resource Time Stamp – From Promote Time stamp on resource file from promote process.		8 V
ZRSCVERS	Resource Version - From Promote Version number of resource from promotion process.		12 V
ZRSCVRFY	Verify Resource File on Connect Verifies the file statistics of a resource file on the desktop during each agent connect process. If you are deploying versioned resources, this attribute is ignored. The rules for the E (exist) flag apply to both Client Automation- and non-Sentient distributed files. (For example, a Client Automation-managed application file or CONFIG.SYS file.) Thus, you must specify Y or N for an existing file to be distributed. Y = The file will be distributed if one or more of the following conditions is met: The file has not been distributed before. The date, time, or size of the file has changed. The file was re-promoted. N = The file will be distributed if: The file has not been distributed before. Or, The file was re-promoted. E = The file will be distributed only if it does not currently exist on the desktop. M = Reserved for Client Automation maintenance files. During the last step of the connect, the file is moved from the EDMDATA subdirectory to its proper location. Example: can be used to update EDMGETRS.EXE. S = This file is distributed if the file size has changed. T = The file is distributed if the file time has changed. D = File is distributed if the file date has changed. L = Forces the lock method to be used. The file will be automatically queued up to be deployed on restart, and the reboot flag will be set. If the L flag is not specified, and the file to be deployed is currently in use (locked) the	U	1 V

	lock method is automatically used anyway. What the L flag does is force the lock method to be always used to deploy this file, whether or not it is currently in use (locked).		
	The L flag will also skip decompression and dispatch the open error method (RADLKM). This is used for system dlls that must be replaced. Note that for the file to be verified during subsequent connects, you must also set this variable to Y. U = File is distributed only if the date and time on the CONFIGURATION SERVER is newer than the date and time on the desktop. R = Resource Stamp. For internal use only. V = (Windows only) Used in when normal date/time stamp check cannot be used. Usually when a .dll or .exe is involved and the date and time stamp are changing frequently. Radconct.exe checks the internal version number of the target file and compares it with the one that Client Automation has. If the incoming file has a higher internal version number than it will be placed on the desktop over the existing file. P = (Windows only) Set during packaging, enforces Windows System File Protection (SFP). If the verification is successful, the resource is in the desired state and no further verification is necessary. If the verification fails, Client Automation checks to see if System File Protection protects the file. If the file is protected, it will not be deployed. If the ZRSCVRFY attribute is set to P and deployment of this resource and the service are marked as failed due to SFP processing. If you do not enforce SFP, the file is marked as having passed verification.		
ZSTATUS	Client Status	999	3 V
ZUPDATE	Client Instance Update Method		50 M

Replace Application (REPLACE) Class

File:	PRIMARY

Domain:	SECURITY
Class:	REPLACE

REPLACE Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
NAME	Friendly Name		50 V
REPLACE1 Thru REPLACE9	Replace Service Name	_UNDEF_	32 V
REPOPT	Replace Action [DELETE/YIELD/COEXIST]	DELETE	8 V
ZSTOP000	Stop Expression		100 E

Scheduling (TIMER) Class

File:	PRIMARY	Description:	Use the TIMER Class to schedule service deployments. For more information, see <i>Sentient Enterprise Console Online Help</i> .
Domain:	SECURITY		
Class:	TIMER		
Icon:		Can connect to:	

TIMER Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ALWAYS	Connect to Class connections determine the path of resolution of an agent's desired state. An _ALWAYS_ class connection is always resolved.		50 C
APPSVC	Application		32 V
DOMAIN	Server Domain Name	&(ZMASTER .ZDOMNAME)	32 V
IPADDR	Server IP Address/Name	&(ZMASTER .ZIPADDR)	32 V
MGRNAME	Server Name	&(ZMASTER .ZMGRNAME)	32 V

NAME	Friendly Name		80 V
NETAVAIL	Check Network Availability [Y/N/W] Checks for network availability before executing the command		1 V
OBJGUID	Instance GUID Object Global Unique Identification – <i>for future use.</i>	_UNDEF_	32 V
PINGCNT	Number of Ping Attempts This attribute sets the number of ping attempts to be made by the Configuration Server.	3	2 V
PINGDLAY	Time between Pings (in msec) This attribute sets the amount of time between pings in msec.	2000	6 V
REQUEST	Application Request	"Apply Updates"	32 V
RUNSYNC	Synchronous Timer Execution [Y/N] This attribute sets the value of Yes or No for the synchronous timer execution. The default value is Yes.	Y	1 V
SOCKET	Server Socket Number	&(ZMASTER .ZDSTSOCK)	32 V
ZCHNNAME	Channel [DOMAIN] Name	&(ZSERVICE. ZOBJDOMN)	32 V
ZCREATE	Client Create Method Sentient-supplied method or fully qualified, user-defined method. This value is inherited from the base instance and should not be changed.	radtimeq	80 M
ZDELETE	Client Delete Method Sentient-supplied method or fully qualified, user-defined method. This value is inherited from the base instance and should not be changed.	radtimeq	80 M
ZNOPING	Stop RCS Detection [Y/N/W/S/P] Controls the automatic sensing of a network connection between the agent computer and the Configuration Server. If set to S or P , additionally, it will ping	Y	1 V

	SAP entries according to their priorities, along with the Configuration Server. An expired time will continually evaluate whether communications with the Configuration Server can be established. When communications are established, the command line associated with the time is executed. After executing the command line, the Scheduler service resumes normal evaluation of whether the timer has expired again. This attribute is especially useful for mobile users. Note: In order to use this attribute, you must add it to the TIMER Class template.		
ZOBJPRI	Priority [00-99] Sets the priority for deployment of the ZTIMEQ object. The ZTIMEQ object is deployed relative to the other elements being deployed during the agent connect. The elements with a priority number less than the value of ZOBJPRI are deployed before the ZTIMEQ object. A value of 90 is inherited from the base instance and should not be changed.	90	2 V
ZPRVNAME	Provider (Manager) Name	&(ZSERVICE .ZMGRNAME)	32 V
ZRSCCMDL	Command Line to Execute This attribute indicates the command line that is executed on the subscriber's computer when the timer expires.	radskman req=	255 V
ZSCHDEF	Time Parameter Indicates when the timer expires. The syntax varies depending on the frequency of expiration that can be DAILY, HOURLY, INTERVAL, NUMDAY, WEEKDAY, WEEKLY.		50 V
ZSCHFREQ	Frequency [PERIODIC/ONCE/RANDOM] Use this attribute to specify how often	PERIODIC	10 V

	the timer should expire. ONCE = expire one time. PERIODIC = expire repeatedly. RANDOM = expire in random intervals.		
ZSCHMODE	Timer Owner [MANAGER/DEFAULT] This attribute specifies the timer owner. It is recommended that you leave the timer owner configuration value as DEFAULT.	DEFAULT	32 V
ZSCHTYPE	Type [IMMEDIATE/DEFERRED] Used only when ZSCHFREQ = PERIODIC. Set ZSCHTYPE to DEFERRED to indicate that the first time an event is attempted to be launched, it will be deferred until the next scheduled time, no matter when the timer instance is evaluated. This was designed to handle the case of a daily 4am (non-peak) scheduled event that is sent to the agent computer during the day. If it was not deferred, it would launch during the day instead of "waiting" until the next morning. Example 1: Suppose you create and deploy a timer with the ZSCHDEF = DAILY (&ZSYSDATE,4:00:00) If ZSCHTYPE = IMMEDIATE and it is: Before 4:00:00, the command in the instance will be executed the same day at 4:00:00 After 4:00:00, the command in the instance will be executed immediately If ZSCHTYPE = DEFERRED and it is: Before 4:00:00, the command in the instance will be executed the <i>next</i> day at 4:00:00 After 4:00:00, the command in the instance will be executed the <i>next</i> day at 4:00:00 Example 2: Suppose you create and deploy a timer with the ZSCHDEF = WEEKDAY (FRIDAY,4:00:00) If ZSCHTYPE = IMMEDIATE and it is: Not Friday or Friday and before	DEFERRED	10 V

	4:00:00, the command in the instance will be executed on Friday at 4:00:00 Friday and after 4:00:00, the command in the instance will be executed immediately If ZSCHTYPE = DEFERRED and it is: Not Friday or Friday and before 4:00:00, the command in the instance will be executed a week later on Friday at 4:00:00 Friday and after 4:00:00, the command in the instance will be executed a week later on Friday at 4:00:00		
ZSTOP	Stop Expression Expressions evaluating to "true" in ZSTOP attributes cause resolution of the instance to be skipped. If left blank, the instance is not skipped, and resolution continues. This is useful for assigning a version to a specific set of users. Use the CSDB Editor to set this attribute.		255 E
ZSVCID	Service ID This attribute specifies the object ID of the Application instance that this Scheduling instance is connected to. This value is inherited from the base instance and should not be modified.	&(ZSERVICE.ZOBJID)	80 V
ZUPDATE	Client Update Method The Scheduler UPDATE method that runs on the agent computer. This value is inherited from the base instance and should not be changed.	radtimeq	80 M
ZVERIFY	Client Verify Method	radtimeq	80 M

SD Packages (SD) Class

File:	PRIMARY	Description:	The SD Class applies to the HP-UX operating system. When you publish native UNIX software with Native Packaging, the depot containing native software in compressed format is promoted to the SD Class, similar to the UNIXFILE Class. For more information, see the <i>Sentient Enterprise Console Online Help</i> .
Domain:	SECURITY		
Class:	SD		
Icon:	 SD	Can connect to:	

SD Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ALWAYS	Connect to Class connections determine the path of resolution of an agent's desired state. An _ALWAYS_ class connection is always resolved.		50 C
ADDDEPS	Auto-select Dependencies [Y/N]	N	1 V
AUTOBOOT	Allow Automatic Reboot [Y/N] Set to Y in SD Class instance in case you have wrapped SD software that contains reboot file set.	N	1 V
CMPMD5	Compressed File MD5 Signature	_UNDEF_	32 V
CONTENTS	Required Packages Included in Tar		255 V
COREQ	Corequisites Required		255 V
EXREQ	Exerequisites		255 V
FILEMD5	File MD5 Signature	_UNDEF_	32 V
INSTOPTS	Package Install Options		60 V
LOCATION	Location		255 V

NAME	Friendly Name	Defaults	80 V
OBJGUID	Instance GUID Object Global Unique Identification – <i>for future use</i> .	_UNDEF_	32 V
PREREQ	Prerequisites Required		255 V
RELYONDB	Verify Error Code? [Y/N]		1 V
ZCMPSIZE	Compressed File Size From promotion process if the compressed file option was selected. The size of the compressed file.		11 V
ZCREATE	Method to Install Resource Uses method "Harness" call. The client side script (<code>sd.tcl</code> or <code>svr4.tcl</code>) contains a native command call to install software.	hide nvdkit method	60 V
ZDELETE	Method to De-install Resource Uses method "Harness" call. The client side script (<code>sd.tcl</code> or <code>svr4.tcl</code>) contains a native command call to remove software. Note: On HP-UX, when a native software application is removed, the application files are deleted, but the directory structure will remain.	hide nvdkit method	50 M
ZFILEUPD	Client File Update/Add Method If a file has been updated by an administrator, this method will be executed. Sentient-supplied method or fully qualified, user-defined method.		50 M
ZINIT	Resource Initialization Method A fully qualified, user-defined method.		50 M
ZOPENERR	Client Method on File Open Error Sentient-supplied method or a fully qualified, user-defined method.		50 M
ZPERGID	File Group Name		32 V
ZPERUID	File Owner Name		32 V
ZREPAIR	Resource Repair Method	hide nvdkit method	60 M

	. The client side script (<code>sd.tcl</code> or <code>svr4.tcl</code>) contains a native command call to repair installed software (reinstall).		
ZRSCCFIL	SD Depot File Name Specifies path to the file that is included in this instance. The file contains HP-UX or Solaris software.	<code>_BASE_ _INSTANCE_</code>	255 V
ZRSCCKPT	Resource Checkpoint		1 V
ZRSCCONF	Confirm File Download [Y/N] Y enables you to confirm that the resource was downloaded.	Y	1 V
ZRSCCRC	Resource CRC From the promotion process if the CRC option was selected. Eight-character calculated hexadecimal value.		8 V
ZRSCCSTA	Client File Status	999	3 V
ZRSCDATE	Resource Date Stamp Date stamp on resource file from promote process.		8 V
ZRSCMO	Mandatory/Optional [M/O]	M	1 M
ZRSCNAME	SD Product Spec Specifies a string that is used by native methods to identify software contained in the published depot. This is the complete software spec on HP-UX (tag, version, architecture and vendor) and the SVR4 package name on Solaris.	<code>_BASE_ _INSTANCE_</code>	255 V
ZRSCPADM	Admin ID		30 V
ZRSCPRI	Create Sequence Priority Enables you to specify a priority level for this resource. If disk space on the client desktop is not available, resources will be downloaded based on the priority set here. 01 = Highest priority 10 = Default value 99 = Lowest priority	50	2 V
ZRSCRASH	File Permissions		4 V
ZRSCRSTR	Restart [Y/N]	Y	3

			V
ZRSCSIZE	Resource Size – From Promote Byte count of resource file from promote process.		11 V
ZRSCSRC	Resource Source, i.e., Publisher		20 V
ZRSCSTYP	Server File Type [BINARY/TEXT]	BINARY	8 V
ZRSCSVRB	Version Skip Rebuild Flag [Y/N]		1 V
ZRSCTIME	Resource Time Stamp Time stamp on resource file from promote process.		8 V
ZRSCVERS	Resource Version - From Promote Version number of resource from promotion process.		12 V
ZRSCVRFY	Verify Resource File on Connect Verifies the file statistics of a resource file on the desktop during each agent connect process. If you are deploying versioned resources, this attribute is ignored. The rules for the E(xist) flag apply to both Client Automation- and non-Sentient distributed files. (For example, a Client Automation-managed application file or CONFIG.SYS file.) Thus, you must specify Y or N for an existing file to be distributed. Y = The file will be distributed if one or more of the following conditions is met: The file has not been distributed before. The date, time, or size of the file has changed. The file was re-promoted. N = The file will be distributed if: The file has not been distributed before, Or, the file was re-promoted. E = The file will be distributed only if it does not currently exist on the desktop. M = Reserved for Client Automation maintenance files. During the last step of the connect, the file is moved from the EDMDATA subdirectory to its proper location. Example: can be used to update EDMGETRS.EXE. S = This file is distributed if the file size has changed.	Y	8 V

	T = The file is distributed if the file time has changed. D = File is distributed if the file date has changed. L = Forces the lock method to be used. The file will be automatically queued up to be deployed on restart, and the reboot flag will be set. If the L flag is not specified, and the file to be deployed is currently in use (locked) the lock method is automatically used anyway. What the L flag does is force the lock method to be always used to deploy this file, whether or not it is currently in use (locked). The L flag will also skip decompression and dispatch the open error method (RADLKM). This is used for system dlls that must be replaced. Note that for the file to be verified during subsequent connects, you must also set this variable to Y. over the existing file.		
	U = File is distributed only if the date and time on the Configuration Server is newer than the date and time on the desktop. R = Resource Stamp. For internal use only. V = (Windows only) Used in when normal date/time stamp check cannot be used. Usually when a .dll or .exe is involved and the date and time stamp are changing frequently. Radconct.exe checks the internal version number of the target file and compares it with the one that Client Automation has. If the incoming file has a higher internal version number than it will be placed on the desktop P = (Windows only) Set during packaging, enforces Windows System File Protection (SFP). If the verification is successful, the resource is in the desired state and no further verification is necessary. If the verification fails, Client Automation checks to see if System File Protection protects the file. If the file is protected, it will not be deployed. If the ZRSCVRFY attribute is set to P and deployment of this resource and the service are marked as failed due to SFP processing. If you do not enforce SFP, the file is marked as having passed verification.		
ZSTATUS	Client Status	999	3 V
ZUPDATE	Client Instance Update Method Uses method "Harness" call. The client side	hide nvdkit method	60 V

ZVERIFY	Method to Verify Resource Uses method "Harness" call. The client side script (<code>sd.tcl</code> or <code>svr4.tcl</code>) contains a native command call to verify installed software.	hide nvdkit method	40 M

Solaris Patches (SOLPATCH) Class

File:	PRIMARY	Description:	Solaris operating system only, This class is required by the Client Automation Native Packaging Feature of the Batch Publisher.
Domain:	SECURITY		
Class:	SOLPATCH		
Icon:		Can connect to:	

SOLPATCH Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ALWAYS	Connect To Class connections determine the path of resolution of an agent's desired state. An _ALWAYS_ class connection is always resolved.		50 C
BACKUP	Backup Patched Files [Y/N]	Y	1 V
CLUSTER	Is this a Patch Cluster [Y/N]		1 V
CMPMD5	Compressed File MD5 Signature	_UNDEF_	32 V
CONTENTS	Required Packages Included in Tar		255 V
FILEMD5	File MD5 Signature	_UNDEF_	32 V
INCOMPS	Incompatible Patches		255 V
LOCATION	Location		255 V
NAME	Friendly Name		80 V
OBJGUID	Instance GUID Object Global Unique Identification – <i>for future</i>	_UNDEF_	32 V

	<i>use.</i>		
OBSOLETE	Obsolete Patches		255 V
PKGS	Patched Packages		255 V
RELYONDB	Verify Error Code? [Y/N]		1 V
REQUIRES	Required Patches		255 V
UNCONINS	Unconditionally Install Patch [Y/N]	N	1 V
ZCMPSIZE	Compressed File Size From promotion process if the compressed file option was selected. The size of the compressed file.		11 V
ZCREATE	Method to Install Resource Sentient-supplied method or fully qualified, user-defined method.	hide nvdkit method	60 M
ZDELETE	Method to De-Install Resource	hide nvdkit method	50 M
ZFILEUPD	Client File Update/Add Method		50 M
ZINIT	Resource Initialization Method		50 M
ZOPENERR	Client Method on File Open Error Sentient-supplied method or a fully qualified, user-defined method.		50 M
ZPERGUID	File Group Name		32 V
ZPERUID	File Owner Name		32 V
ZREPAIR	Method to Repair Resource	hide nvdkit method	60 M
ZRSCCFIL	Patch Depot File Name: Specific file to be distributed. A generic pattern (a wildcard character, such as *) or an explicit filename.	_BASE_ INSTANCE_	255 V
ZRSCCKPT	Resource Checkpoint		1 V

ZRSCCONF	Confirm File Download [Y/N]	Y	1 V
ZRSCCRC	Resource CRC		8 V
ZRSCCSTA	Client File Status	999	3 V
ZRSCDATE	Resource Date Stamp – From Promote		8 V
ZRSCMO	Mandatory/Optional on Client [M/O]	M	1 V
ZRSCNAME	Solaris Patch ID	_BASE_ INSTANCE_	255 V
ZRSCPADM	Admin ID		30 V
ZRSCPRI	Create Sequence Priority	50	2 V
ZRSCRASH	File Permissions		4 V
ZRSCRSTR	Restart [Y/N]	Y	3 V
ZRSCSIZE	Resource Size – From Promote		11 V
ZRSCSRC	Resource Source, i.e., Publisher		20 V
ZRSCSTYP	Server File Type [BINARY/TEXT]	BINARY	8 V
ZRSCSVRB	Version Skip Rebuild Flag [Y/N]		1 V
ZRSCTIME	Resource Time Stamp – From Promote		8 V
ZRSCVERS	Resource Version – From Promote		12 V
ZRSCVRFY	Verify Resource File on Connect	Y	8 V
ZSTATUS	Client Status	999	3 V
ZUPDATE	Client Instance Update Method	hide nvdkit	60

		method	M
ZVERIFY	Method to Verify Resource	hide nvdkit method	40 M

SVR4 Packages (SVR4) Class

File:	PRIMARY	Description:	The SVR4 Class applies to the Solaris operating system. When you publish native UNIX software with Native Packaging, the depot containing native software in compressed format is promoted to the SVR4 Class, similar to the UNIXFILE Class. For more information, see the <i>Sentient Enterprise Console Online Help</i> .
Domain:	SECURITY		
Class:	SVR4		
Icon:	 SVR4	Can connect to:	

SVR4 Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ALWAYS	Connect to Class connections determine the path of resolution of an agent's desired state. An _ALWAYS_ class connection is always resolved.		50 C
ADMIN	Admin File Exists? [Y/N]		1 V
ADMINOBJ	Is this admin object? [Y/N]		1 V
AUTOBOOT	Allow automatic reboot [Y/N]		1 V
CMPD5	Compressed File MD5 Signature	_UNDEF_	32 V
CONTENTS	Required Packages Included in Tar		255 V
FILEMD5	File MD5 Signature	_UNDEF_	32 V
INCOMP	Incompatible Packages		255 V
LOCATION	Location		255 V

NAME	Friendly Name		80 V
OBJGUID	Instance GUID: Object Global Unique Identification – <i>for future use.</i>	_UNDEF_	32 V
PKGVER	Package Version		50 V
PKGREV	Package Revision		50 V
PREREQ	Prerequisites Required		255 V
RELYONDB	Verify Error Code? [Y/N]		1 V
RESPFILE	Response File Path		80 V
RESPOBJ	Is this response object? [Y/N]	N	1 V
REVERSE	Reverse Dependency Packages		255 V
RESPONSE	Response file exist? [Y/N]	N	1 V
ZCMPsize	Compressed File Size From promotion process if the compressed file option was selected. The size of the compressed file.		11 V
ZCREATE	Method to Install Resource Uses method "Harness" call. The client side script (<code>sd.tcl</code> or <code>svr4.tcl</code>) contains a native command call to install software.	hide nvdkit method	60 V
ZDELETE	Method to De-install Resource Uses method "Harness" call. The client side script (<code>sd.tcl</code> or <code>svr4.tcl</code>) contains a native command call to remove software. Note: On HP-UX, when a native software application is removed, the application files are deleted, but the directory structure will remain.	hide nvdkit method	50 M
ZFILEUPD	Client File Update/Add Method If a file has been updated by an administrator,		50 M

ZINIT	Resource Initialization Method A fully qualified, user-defined method.		50 M
ZOPENERR	Client Method on File Open Error Sentient-supplied method or a fully qualified, user-defined method.		50 M
ZPERGID	File Group Name		32 V
ZPERUID	File Owner Name		32 V
ZREPAIR	Method to Repair Resource Uses method "Harness" call. The client side script (<code>sd.tcl</code> or <code>svr4.tcl</code>) contains a native command call to repair installed software (reinstall).	hide nvdkit method	60 M
ZRSCCFIL	SVR4 Depot File Name Specifies path to the file that is included in this instance. The file contains Solaris software.	_BASE_ INSTANCE_	255 V
ZRSCCKPT	Resource Checkpoint		1 V
ZRSCCONF	Confirm File Download [Y/N] Y enables you to confirm that the resource was downloaded.	Y	1 V
ZRSCCRC	Resource CRC From the promotion process if the CRC option was selected. Eight-character calculated hexadecimal value.		8 V
ZRSCCSTA	Client File Status	999	3 V
ZRSCDATE	Resource Date Stamp Date stamp on resource file from promote process.		8 V
ZRSCMO	Mandatory/Optional [M/O]	M	1 M
ZRSCNAME	SVR4 Package Tag Specifies a string that is used by native methods to identify software contained in the published depot. This is SVR4 package name.	_BASE_ INSTANCE_	255 V
ZRSCPADM	Admin ID		30 V

ZRSCPRI	Create Sequence Priority Enables you to specify a priority level for this resource. If disk space on the client desktop is not available, resources will be downloaded based on the priority set here. 01 = Highest priority 10 = Default value 99 = Lowest priority	50	2 V
ZRSCRASH	File Permissions		4 V
ZRSCRSTR	Restart [Y/N]	Y	3 V
ZRSCSIZE	Resource Size – From Promote Byte count of resource file from promote process.		11 V
ZRSCSRC	Resource Source, i.e., Publisher		20 V
ZRSCSTYP	Server File Type [BINARY/TEXT]	BINARY	8 V
ZRSCSVRB	Version Skip Rebuild Flag [Y/N]		1 V
ZRSCTIME	Resource Time Stamp – From Promote Time stamp on resource file from promote process.		8 V
ZRSCVERS	Resource Version - From Promote Version number of resource from promotion process.		12 V
ZRSCVRFY	Verify Resource File on Connect Verifies the file statistics of a resource file on the desktop during each agent connect process. If you are deploying versioned resources, this attribute is ignored. The rules for the E(xist) flag apply to both Sentient- and non-Sentient distributed files. (For example, a Sentient-managed application file or CONFIG.SYS file.) Thus, you must specify Y or N for an existing file to be distributed. Y = The file will be distributed if one or more of the following conditions is met: The file has not been distributed before. The date, time, or size of the file has changed. The file was re-promoted.	Y	8 V

	N = The file will be distributed if: The file has not been distributed before, Or, the file was re-promoted. E = The file will be distributed only if it does not currently exist on the desktop. M = Reserved for Client Automation maintenance files. During the last step of the connect, the file is moved from the <code>EDMDATA</code> subdirectory to its proper location. Example: can be used to update <code>EDMGETRS . EXE</code>. S = This file is distributed if the file size has changed. T = The file is distributed if the file time has changed. D = File is distributed if the file date has changed. L = Forces the lock method to be used. The file will be automatically queued up to be deployed on restart, and the reboot flag will be set. If the L flag is not specified, and the file to be deployed is currently in use (locked) the lock method is automatically used anyway. What the L flag does is force the lock method to be always used to deploy this file, whether or not it is currently in use (locked). The L flag will also skip decompression and dispatch the open error method (RADLKM). This is used for system dlls that must be replaced. Note that for the file to be verified during subsequent connects, you must also set this variable to Y.		
	U = File is distributed only if the date and time on the Configuration Server is newer than the date and time on the desktop. R = Resource Stamp. For internal use only. V = (Windows only) Used in when normal date/time stamp check cannot be used. Usually when a <code>.dll</code> or <code>.exe</code> is involved and the date and time stamp are changing frequently. <code>Radconct .exe</code> checks the internal version number of the target file and compares it with the one that Client Automation has. If the incoming file has a higher internal version number than it will be placed on the desktop over the existing file. P = (Windows only) Set during packaging, enforces Windows System File Protection (SFP). If the verification is successful, the resource is in the desired state and no further verification is necessary. If the verification fails,		

	Client Automation checks to see if System File Protection protects the file. If the file is protected, it will not be deployed. If the ZRSCVRFY attribute is set to P and deployment of this resource and the service are marked as failed due to SFP processing. If you do not enforce SFP, the file is marked as having passed verification.		
ZSTATUS	Client Status	999	3 V
ZUPDATE	Client Instance Update Method Uses method "Harness" call. The agent side script (<code>sd.tcl</code> or <code>svr4.tcl</code>) contains a native command call to update software.	hide nvdkit method	60 V
ZVERIFY	Method to Verify Resource Uses method "Harness" call. The client side script (<code>sd.tcl</code> or <code>svr4.tcl</code>) contains a native command call to verify installed software.	hide nvdkit method	40 M

TPM Configuration (TPMCFG) Class

File:	PRIMARY	Description:	This class is reserved for future use.
Domain:	SECURITY		
Class:	TPMCFG		

TPMCFG Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
BOOTOPT	Boot option [NONE PROMPT REJECTPROM]		16 V
RBPATH	Backup Archive Path		255 V
STATE	TPM progress		32 V
TPMERF	Emergency Recovery File Path		255 V
TPMRES	TPM password Reset File Path		255 V
ZPWD1	Password 1 BIOS		255

			V
ZPWD2	Password 2 TPM Owner Password		255 V
ZPWD3	Password 3 Emergency Recovery Token		255 V
ZPWD4	Password 4 Password Reset token		255 V

Unix File Resources (UNIXFILE) Class

File:	PRIMARY	Description:	An instance of the UNIXFILE Class defines a file, or a group of files, to include in an application service. An instance of this class lets you specify such characteristics about the file as the file's name, size, and date stamp.
Domain:	SECURITY		
Class:	UNIXFILE		
Icon:		Can connect to:	

UNIXFILE Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ALWAYS	Connect to Class connections determine the path of resolution of an agent's desired state. An _ALWAYS_ class connection is always resolved.		50 C
DIRPERMS	Perms for newly created directories Permissions assigned to newly created directories.		4 V
LOCATION	Location		255 V
NAME	Friendly Name	Defaults	80 V
OBJGUID	Instance GUID Object Global Unique Identification – <i>for future use</i> .	_UNDEF_	32 V
SIGTYPE	File Signature Type		3 V
ZCMPKEY	Compression Key		80 V

ZCMPSIZE	Compressed File Size From promotion process if the compressed file option was selected. The size of the compressed file.		11 V
ZCONTEXT	Component Context		8 V
ZCREATE	Method to Install Resource Sentient-supplied method or fully qualified, user-defined method.		50 M
ZDELETE	Method to De-install Resource Sentient-supplied method or fully qualified, user-defined method.	RADREMF	50 M
ZFILEUPD	Client File Update/Add Method If a file has been updated by an administrator, this method will be executed. Sentient-supplied method or fully qualified, user-defined method.		50 M
ZINIT	Resource Initialization Method A fully qualified, user-defined method.		50 M
ZOBJPTCH	Perform Patching [Y/N]	Y	1 V
ZOPENERR	Client Method on File Open Error Sentient-supplied method or a fully qualified, user-defined method.		50 M
ZPERGID	File Group Name		32 V
ZPERUID	File Owner Name		32 V
ZREPAIR	Client Instance Repair Method		50 V
ZRSCCFIL	Resource File Name Specific file to be distributed. A generic pattern (a wildcard character, such as *) or an explicit filename.	_BASE_ INSTANCE_	255 V
ZRSCCKPT	Resource Checkpoint		1 V
ZRSCCONF	Confirm File Download [Y/N] Y enables you to confirm that the resource was downloaded.	Y	1 V
ZRSCCRC	Resource CRC From the promotion process if the CRC option		8 V

ZRSCCSTA	Client File Status	999	3 V
ZRSCDATE	Resource Date Stamp Date stamp on resource file from promote process.		8 V
ZRSCMO	Mandatory/Optional [M/O]	M	1 V
ZRSCNAME	Resource Name Name or description of resource.	&ZRSCCFIL	30 V
ZRSCPADM	Admin ID		30 V
ZRSCPRI	Create Sequence Priority Enables you to specify a priority level for this resource. If disk space on the client desktop is not available, resources will be downloaded based on the priority set here. 01 = Highest priority 10 = Default value 99 = Lowest priority	50	2 V
ZRSCRASH	File Permissions		4 V
ZRSCRSTR	Restart [Y/N/R] Y to restart. Set to R to bypass Windows file caching and write directly to disk (allowing checkpoint restart to operate correctly).	Y	3 V
ZRSCSIG	File Signature		32 V
ZRSCSIZE	Resource Size – From Promote Byte count of resource file from promote process.		11 V
ZRSCSRC	Resource Source, i.e., Publisher		20 V
ZRSCSTYP	Server File Type [BINARY/TEXT]	BINARY	8 V
ZRSCSVRB	Version Skip Rebuild Flag[Y/N]		1 V
ZRSCTIME	Resource Time Stamp – From Promote Time stamp on resource file from promote process.		8 V
ZRSCVERS	Resource Version - From Promote		12

			V
ZRSCVRFY	Verify Resource File on Connect Verifies the file statistics of a resource file on the desktop during each agent connect process. If you are deploying versioned resources, this attribute is ignored. The rules for the E(xist) flag apply to both Sentient- and non-Sentient distributed files. (For example, a Sentient-managed application file or <code>CONFIG.SYS</code> file.) Thus, you must specify Y or N for an existing file to be distributed. Y = The file will be distributed if one or more of the following conditions is met: The file has not been distributed before. The date, time, or size of the file has changed. The file was re-promoted. N = The file will be distributed if: The file has not been distributed before, Or, the file was re-promoted. E = The file will be distributed only if it does not currently exist on the desktop. M = Reserved for Client Automation maintenance files. During the last step of the connect, the file is moved from the <code>EDMDATA</code> subdirectory to its proper location. Example: can be used to update <code>EDMGETRS.EXE</code>. S = This file is distributed if the file size has changed. T = The file is distributed if the file time has changed. D = File is distributed if the file date has changed. L = Forces the lock method to be used. The file will be automatically queued up to be deployed on restart, and the reboot flag will be set. If the L flag is not specified, and the file to be deployed is currently in use (locked) the lock method is automatically used anyway. What the L flag does is force the lock method to be always used to deploy this file, whether or not it is currently in use (locked). The L flag will also skip decompression and dispatch the open error method (RADLKM). This is used for system dlls that must be replaced. Note that for the file to be verified during subsequent connects, you must also set this variable to Y.	Y	8 V
	U = File is distributed only if the date and time on		

	the CONFIGURATION SERVER is newer than the date and time on the desktop. R = Resource Stamp. For internal use only. V = (Windows only) Used in when normal date/time stamp check cannot be used. Usually when a .dll or .exe is involved and the date and time stamp are changing frequently. Radconct.exe checks the internal version number of the target file and compares it with the one that Client Automation has. If the incoming file has a higher internal version number than it will be placed on the desktop over the existing file. P = (Windows only) Set during packaging, enforces Windows System File Protection (SFP). If the verification is successful, the resource is in the desired state and no further verification is necessary. If the verification fails, Client Automation checks to see if System File Protection protects the file. If the file is protected, it will not be deployed. If the ZRSCVRFY attribute is set to P and deployment of this resource and the service are marked as failed due to SFP processing. If you do not enforce SFP, the file is marked as having passed verification.		
ZSTATUS	Client Status	999	3 V
ZUPDATE	Client Instance Update Method		50 M

Version Groups (VGROUP) Class

File:	PRIMARY	Description:	Each instance of the Version Group (VGROUP) class defines a set of versions for an application. This class contains connections to the Versions (VERSION) class, created using the Version Group Editor in the CSDB Editor.
Domain:	SECURITY		
Class:	VGROUP		
Icon:		Can connect to:	

VGROUP Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type

ACTDATE	Activated date/time Data maintained by the Agent in the VGROUP object on the agent computer. Do not enter a value. This is set by Agent.		32 V
CONTROL	Vgroup Control [CLIENT/MANAGER] Indicates whether the Client Automation administrator (MANAGER) or the subscriber (CLIENT) controls which version to activate on the agent computer.	MANAGER	8 V
CURVERS	Currently active version Data maintained by the Agent in the VGROUP object on the agent computer. Do not enter a value. This is set by Agent.		32 V
INITIAL	Initial version to Activate Indicates which version to activate on the agent computer.		32 V
NAME	Friendly Name The friendly name for the VGROUP instance.		255 V
NEXTVERS	Next version to activate Data maintained by the Agent in the VGROUP object on the agent computer. Do not enter a value. This is set by Agent.		32 V
REQACTDT	Requested Activation Date YYYYMMDD The earliest date on which a version in this version group will be activated on any agent computer.		8 V
REQACTTM	Requested Activation time HH:MM:SS The earliest time, on the date specified by the REQACTDT attribute, after which a version in the version group will be activated on any agent computer.		8 V
ROLLBACK	Rollback if install fails [Y/N] Indicates whether to automatically roll back to the previously activated version when deployment of a new version fails.	Y	1 V
SOURCE	Source for currently active version Data maintained by the Agent in the VGROUP object on the agent computer. Do not enter a value. This is set by the Agent.		8 V
STATUS	Status of current version – 000 Data maintained by the Agent in the VGROUP object on the agent computer. Do not enter a value. This is set by the Agent.		3 V

TIMERCON	Timer for next Version Activation If you specify a "next version to activate," the CSDB Editor automatically creates a timer and stores the connection to that timer in this attribute.		50 C
VERCON01 to VERCON05	Version Connection Connects to each version in the version group. Each VERCON0n attribute contains a connection to one instance of the VERSION Class. This is set when you assign a version to the version group in the Version Group Editor.		50 C
ZSTOP001 to ZSTOP003	Stop Resolution Expression Expressions evaluating to "true" in ZSTOP attributes cause resolution of the instance to be skipped. If left blank, the instance is not skipped, and resolution continues. This is useful for assigning a version to a specific set of users. Use the CSDB Editor to set this attribute.		255 E

Versions (VERSION) Class

File:	PRIMARY	Description:	Each instance of the VERSION Class defines one <i>version</i> of an application to be deployed and managed by Client Automation. Use the Version Group Editor to create Versions class instances and assign them to a Version Group.
Domain:	SECURITY		
Class:	VERSION		
Icon:		Can connect to:	

VERSION Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
NAME	Friendly Name The friendly name for the VERSION instance. This is set when you create the instance using the Version Group Editor.		255 V
PACKAGE	Package Connection Connects to a PACKAGE Class instance, which represents the packaged software for this version.		50 A
ZSTOP001 to ZSTOP003	Stop Resolution Expression Expressions evaluating to "true" in ZSTOP attributes cause resolution of the instance to be skipped. If left		255 E

WTS App Comp Scripts (WTSACS) Class

File:	PRIMARY	Description:	This class contains one instance for each Windows Terminal Server Application Compatibility Script required for an application. ACS are published into the database by an administrator. Use the instances in this class to configure the ACS.
Domain:	SECURITY		
Class:	WTSACS		
Icon:		Can connect to:	

WTSACS Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
BDELETE	Method Before Delete Set this attribute for the Uninstall ACS to &ZRSCCFIL so this script is run at service removal.		50 M
LOCATION	Location	&(ZMASTER.ZLIBDRV) &(ZMASTER.ZLIBDRV)	255 V
NAME	Friendly Name	Defaults	80 V
TYPE	Install, Logon, Uninstall (I/L/U) - Set to I for Install script. - Set to L for a logon script. - Set to U for an Uninstall Script.		1 V
ZCMPSIZE	Compressed File Size		11 V
ZCREATE	Method to Install Resource		50 M
ZDELETE	Method to De-install Resource	RADREMF	50 M
ZRSCCFIL	Resource File Name		255 V
ZRSCCSTA	Client File Status	999	3 V
ZRSCDATE	Resource Date Stamp – From		8

	Promote		V
ZRSCNAME	Resource Name	&ZRSCCFIL	30 V
ZRSCPRI	Create Sequence Priority	50	2 V
ZRSCRASH	DOS File Attribute [R/A/S/H]		4 V
ZRSCSIZE	Resource Size – From Promote		11 V
ZRSCSTYP	Server File Type [BINARY/TEXT]	BINARY	8 V
ZRSCSTIME	Resource Time Stamp – From Promote		8 V
ZRSCVRFY	Verify Resource Files on Connect	Y	8 V
ZSTOP000	Stop Expression The export deck may have this configured incorrectly with a period in place of the comma. Check to be sure it is shown as above.	/(EDMGETV (ZCONFIG.WTSSRVR='Y'))	50 E

WTS Control (WTCTRL) Class

File:	PRIMARY	Description:	Instances in this class describe some behaviors you may need for the application, if you require logoffs for application events, the type of install and the timeout settings.
Domain:	SECURITY		
Class:	WTCTRL		
Icon:		Can connect to:	

WTCTRL Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
LD	Logoff For Delete [Y/N]Specify Y to logoff for	Y	1 V

	application removal.		
LI	Logoff For Install [Y/N] Specify Y to logoff for application installation	N	1 V
LR	Logoff For Repair [Y/N] Specify Y to logoff for application repair.	Y	1 V
LU	Logoff For Update [Y/N] Specify Y to logoff for application update.	Y	1 V
NAME	Friendly Name	Default	80 V
RW	Rewarn of Logoff at [minutes] Specify when you want a second warning to appear that the Server will be logged off.	2	2 V
TO	Disconnect Time Out [minutes] Specify in minutes how long the Windows Terminal Server should wait before disconnecting any users from the server. A warning	15	4 V

	message will appear on the Windows Terminal Server showing how long before the log off will occur.		
WTSOPTNS	WTS Logoff Settings Composite concatenated logoff field to flow down to the client	LI=&(LI),LU=&(LU),LR=&(LR),LD=&(LD),TO=&(TO),RW=&(RW)	60 V
WTSTYPE	Install / Execute Mode [I/E] Set to I to install files natively on the Windows Terminal Server such as applications that an administrator may use specifically on the server. Set to E for an application that will be used by the Window Terminal Server's agents.	E	1 V

The SOFTWARE Domain

The SOFTWARE Domain contains information about the software being managed and the methods used to deploy the software.

Alert/Defer (ALERTDEF) Class

File:	PRIMARY	Description:	Use an instance of the ALERTDEF Class to define an application alert message or deferral.
Domain:	SOFTWARE		
Class:	ALERTDEF		
Icon:		Can connect to:	ZSERVICE

ALERTDEF Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ALERTMSG	Alert Message (! = High Priority) An exclamation point (!) preceding "Service Alert Message" denotes a high priority message.		100 V
DM	Alert Mode [I/U/B] Set to I for Install. Set to U for Update. Set to B for either Install or Update.	B	1 V
DN	Number of deferrals allowed Maximum number of deferrals allowed before the action specified in DM (Alert Mode) is taken.	0	2 V
DT	Network Threshold Network bandwidth threshold in bytes.	86000	12 V
DBT	Below Threshold Alert Size (bytes)	50000	12 V
DAT	Above Threshold Alert Size (bytes)	0	12 V
DTO	Alert Timeout (in seconds) Specify in seconds the amount of time to display the Defer Alert dialog box. After the timeout is reached, the action specified in the DA (Action on timeout) variable, will be taken.	120	4 V
DA	Action on timeout Cont/Defer [C/D] Specify C to continue with the specified action if the subscriber does not respond to the defer alert dialog box by the value specified in the DTO (Alert Timeout). Specify D to defer the specified action.	C	1 V
DI	Allow Install Deferral up to [DATE]	NA	20

	After this date is reached, the option to defer installation will no longer be available. The application will be installed. This should be in the format YYYYMMDD.		V
DU	Allow Update Deferral up to [DATE] After this date is reached, the option to defer installation will no longer be available. The application will be updated. This should be in the format YYYYMMDD.	NA	20 V
NAME	Friendly Name Friendly Name for the instance.	Default	80 V
DEPOPTNS	Defer Options (Do Not Modify) This attribute is used to resolve the values of the other attributes of this class. Do not modify.	&(DM),& (DN),& (DT), &(DBT),& (DAT),& (DTO), &(DA),& (DI),&(DU)	12 V

Application (ZSERVICE) Class

File:	PRIMARY	Description:	Use an instance of the ZSERVICE Class to define an application and its characteristics to Client Automation. An instance of the ZSERVICE Class always includes the resource files that comprise it.
Domain:	SOFTWARE		
Class:	ZSERVICE		
Icon:		Can connect to:	PACKAGE, FILE, BEHAVIOR, DIALOG, DEPT, WORKGRP, TIMER, VGROUP, STAGER, HTTP, EXECUTE, ZSERVICE, PDACONFIG

ZSERVICE Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ALWAYS	Contains Class connections determine the path of resolution of an agent's desired state. An _ALWAYS_ class connection is always resolved.		50 C
ALWAYS	Contains An _ALWAYS_ class connection is always resolved. An Includes connection is used to express inter-package relationships.		50 I

ALWAYS	Utility Resolution Method Any method you specify for this attribute is unconditionally executed by Client Automation when this instance is resolved. A valid method name, for example, ZSYSTEM.ZMETHOD.PUT_ZERROR_HIST.		51 M
ADAPTIVE	Auto Adaptability [Y/N] Indicates whether the installed package is dependent on client settings that must be monitored periodically. If the settings change, the client must reconnect to the Configuration Server to get new or different components. Useful for plug-and-play services.		1 V
APPTYPE	Application Type		12 V
AUTHOR	Author Name Name of the author of the service that appears in the properties for the service in the Service List.		50 V
BDELETE	Service Pre-Delete Method Method to run before deleting the service.		100 M
BREPAIR	Service Pre-Update Method Method to run before updating the service.		100 M
BUPDATE	Service Pre-Repair Method Method to run before repairing the service.		100 M
CACHE	App Element Caching [Y/N]	N	1 V
CACHELIM	Percent Disk Limit for Cache Cache limit, which is defined as the percentage of used drive space. Type a number between 000 and 100. Note: Used for Windows Installer applications only. If the percentage of used space is greater than the cache limit, then all of the cached files for the product are removed and the cache folder is deleted. This is checked after every file is cached on the disk.	000	3 V
CACHELOC	CACHE Location on Client For Windows Installer applications only.	_UNDEF_	254 V

	Location of the folder on the agent computer that is used to cache the compressed application files needed for the product. Sentient support for Windows Installer tags the PRODGUID value to this value to create the folder. For example, If CACHELOC=c:\progra~1\HP, and = 12345_XXXX\cache Sentient support for Windows Installer tags the PRODGUID value to this value to create the folder. For example, If CACHELOC=c:\progra~1\HP, and PRODGUID = 12345_XXXX\cache, the cache folder would be: c:\progra~1\HP\12345_XXXX\cache. Note: The folder name \cache is automatically appended to PRODGUID. If you are not deploying a Windows Installer-enabled application, the files will be cached in IDMDATA.		
CATGROUP	Catalog Group Name Type a name for the catalog to be displayed to subscribers when they click Properties in the Application Self-service Manager user interface.		30 V
COMPSIZE	Application Size – Compressed The size of the compressed application displayed to the subscribers when they click Properties in the Application Self-service Manager user interface. Cumulative value of the COMPSIZE defined in the PACKAGE Class.		12 V
DELDATE	Delete Date Indicates when the application was removed (in local time) from the agent computer. The Agent manages and maintains this attribute.		32 V
DESCRIPT	Application Description Description of the service that appears in the properties for the service in the Service List. Value is set initially in the Long Description field in the New Application Wizard (used to create a service).		100 V
ERTYPE	Event Reporting Method [O/E/X] Set on the Configuration Server, this sends an	O	1 V

	APPEVENT object to the Configuration Server. O = Object Currently supports object format only.		
EVENTS	Events to Report Indicates which events to report on. Event to report on: AI = Application Install AD = Application De-install AU = Application Update AR = Application Repair AV = Application Verify VA = Version Activation VD = Version Deactivation What (about the event) to report on: S = Success F = Failure B = Both Success and Failure N = None	AI=B AD=B AU=B AR=B AV=F VA=B VD=B	50 V
INSTDATE	Installed Date Indicates when the application was installed (in local time) on the agent computer. The Agent manages and maintains this attribute. The date is in text form, as in: Jan 28, 1998 16:10:00.		32 V
ITEMS	Number of Component Instances Cumulative numeric value of the component instances of all the attached packages. (See ITEMS from PACKAGE Class.)		7 V
LREPAIR	Local Repair [Y/N] Enables local repair of broken applications. If an application is broken because of missing files, the files (stored locally) can be used to repair the application.		1 V
MCELIBL	Service Multicast Eligible [Y/N] Indicates if the application is eligible for multi-casting.	Y	1 V
NAME	Friendly Name Friendly name for the service. This appears in the properties for the service in the Service List.		80 V
OWNER	Application Contact Package owner, as documented by the Admin Publisher.		50 V
PRICE	Price		10

	Type the price of an application to be displayed to subscribers when they click Properties in the Application Self-service Manager user interface.		V
PUBDATE	Published Date of Service Date service was published.		32 V
REBOOT	Install/Update/Delete/Version Change Used to restart the agent computer after the service is installed. Event to report on: AI = Install AD = Deinstall AU = Update AR = Repair AV = Verify Type of reboot: S = Soft Boot (Default of type Y panel.) H = Hard Boot (Default of type A panel.) N = None Type of panel: Q = No panel. A = OK button only. Y = OK and Cancel button. Type of connect: None specified: Reboot on Machine connect (context = m). U = reboot on user connect only (context = u). MU = reboot when both machine and user parts of the service have been installed. Example: AI=S performs a soft boot on application installation.		64 V
RECONFIG	Reconfiguration Enabled [Y/N] Indicates whether an application can be relocated after it has been installed.		1 V
REMOVAL	Un-Managed Behavior [A/D/U] Controls removal of the service. If ZVCMO is set to M, set REMOVAL to: A = Abandon (deletes the objects, but not the components) D = Delete (deletes the objects and components) If ZVCMO is set to O, set REMOVAL to: A = Abandon (deletes the objects, but not the components) D = Delete (deletes the objects and components) U = Unmanage (does not delete the objects or	D	1 V

	components)		
RSTRSIZE	Download Restart Threshold (in bytes) Use the RSTRSIZE attribute in the appropriate ZSERVICE Class instance to control which files are enabled for check point restart based on the amount of data being downloaded (in bytes). This is for Windows Installer-Enabled Applications (MSI)	0	11 V
RUNDLG	Dialog Processing [Y/N] Specifies whether to enable processing of dialog boxes during the installation of the service.	N	1 V
RUNLEVEL	Runlevel Support for UNIX.		1 V
SCHEDOK	Update Schedule Locally [Y/N] Specifies whether the subscribers are allowed to change the update schedule for the service locally. Y = Subscriber is allowed to change the schedule. N = Configuration Server controls the update schedule.		1 V
SIZE	Application Size – Uncompressed The size of the uncompressed application displayed to the subscribers when they click Properties in the Application Self-service Manager user interface.		12 V
THROTYPE	[ADAPTIVE/RESERVED/NONE/] Type of Bandwidth throttling to use. Set to ADAPTIVE to yield to other services that are using the network. Set to RESERVED to allow for a specific reservation of the bandwidth. It is the maximum percentage of network bandwidth to use. Set to NONE for no bandwidth throttling, and use the maximum available bandwidth. NONE is the default.		8 V
UIOPTION	Progress Indicator [NONE/FULL/INFO] Controls whether the status window appears. NONE = No interface appears. FULL = Interface appears and Cancel button is available. INFO = Interface appears with no option to cancel.		4 V

UIOPTMSI	MSI Progress Indicator [NONE/FULL/INFO] Controls whether the MSI status window appears. NONE = No interface appears. FULL = Displays what is happening on the agent computer and provides the subscriber with controls to make changes. INFO = Displays what is happening on the agent computer, but disables all the controls so that the subscriber cannot make any changes. See the Setup log files and Event Viewer for warnings, errors, successes, failures, and other details.		
UPDDATE	Upgrade Date (Programmatic) Stores the date when the file was updated on the Configuration Server.		32 V
UPGDATE	When Application was Upgraded Stores the date when the application was last upgraded (in local time) on the agent computer. This value is set by the Sentient user after the dialogs have been installed. This date can be used for reporting purposes. The date is in text form, as in: Jan 28, 1998 16:10:00.		32 V
URL	WEB URL Name Address of a Web page where the subscriber can find additional information about the service.		100 V
VENDOR	Vendor Name Name of the vendor of the service.		24 V
VERDATE	Verified Date of Service Indicates when the application was last verified (in local time) on the agent computer. The Agent manages and maintains this attribute.		32 V
VERSION	Version Description Version of the software. This appears in the properties for the service in the Service List.		50 V
ZAVIS	Available, Verified, Installed, Sync F The Agent manages and maintains this attribute to show the different states of the application in the Service List. Y = Yes	YXNX	4 V

	N = No X = Unknown The four states are: Available indicates whether a service is available from the Configuration Server. Verified indicates whether a service has been verified. Installed indicates whether the service has been installed. Synchronized indicates whether the installed service has all of the latest changes from the Configuration Server.		
ZBITARCH	Target Bit Architecture [/32/64] Set to 32 to deploy to 32-bit architectures. Set to 64 to deploy to 64-bit architectures. Set to blank to deploy to all architectures.		8 V
ZBWAPCT	Adaptive Percentage Adaptive minimum percentage of bandwidth used when using bandwidth throttling.		3 V
ZBWPCT	Reserved Percentage Reserved maximum percentage bandwidth used when using bandwidth throttling.		3 V
ZCONTEXT	Component Context [M/U]		8 V
ZCREATE	Service Installation Method Sentient-supplied method or fully qualified, user-defined method.		100 M
ZDELETE	Service Delete Method Sentient-supplied method or fully qualified, user-defined method.		100 M
ZDISCONN	Disconnect on Install [Y/N] Allows the client to disconnect from the Configuration Server if there is an open session with the Configuration Server. Y = Disconnects the client from the Configuration Server. N = Does not disconnect from the client from the Configuration Server.	Y	1 V
ZINIT	Service Initialization Method A fully qualified, user-defined method.		100 M
ZREPAIR	Service Repair Method Name of the method to repair the service.		100 M

ZSTOP000 to ZSTOP002	Stop Resolution Expression Expressions evaluating to "true" in ZSTOP attributes cause resolution of the instance to be skipped. If left blank, the instance is not skipped, and resolution continues. This is useful for assigning a version to a specific set of users. Use the CSDB Editor to set this attribute.		100 E
ZSTOP999	Stop Unless RCA Connect	"\\(UPPER (EDMGETV (ZMASTER, ZDOMNAME)) ='SOFTWARE'	50 E
ZSVCCAT	Service Visible in Catalog? [Y/N]		1 V
ZSVCCSTA	Service Status on Client (999) Status code for the service that leads you to the reason why files for a service may not be deployed correctly. 000 = All actions for service completed. 008 = At least one resource had a writer error. 014 = No disk space for at least one resource. 015 = ZDLIMIT exceeded for at least one resource. 024 = A method timed out. 036 = A create method set ZBRC>7. 900 = Not all files were verified. No work done for service.	999	3 V
ZSVCMO	Mandatory/Optional Service [M/O/MO/OM] Indicates whether the service is mandatory or optional on the agent. If you are using Application Manager <i>and</i> Application Self-service Manager, you could also specify mandatory and then optional, MO , or optional then mandatory, OM . The first character indicates how the application will be handled before installation. The second character indicates how the application will be handled after installation.	O	1 V
ZSVCMODE	Application Context [M/U/MU] Specifies whether the service is to be installed in the Machine or User context. M = Machine U = User		2 V
ZSVCNAME	Service Name/Description	Unknown Service	24 V

ZSVCPRI	Service Create Ordering [01-99] Enables you to specify a priority level for this service. If disk space on the agent desktop is not available, services will be downloaded based on the priority level set here. 01 = Highest priority 10 = Default value 99 = Lowest priority		2 V
ZSVCTTYP	Application Target Type [A/S] Indicates which agent will receive this application. A = Application Manager S = Application Self-service Manager		2 V
ZSYSACCT	Install Under System Account [Y/N/M] Specifies whether to install the service under the system account or the user's account. Y = The application is installed using the system rights. N = The application is installed using the rights of the user who is logged on. M = The application is installed using the system rights if the current connect is in the machine context (context =m).	Y	1 V
ZUPDATE	Service Update Method Sentient-supplied method or fully qualified, user-defined method.		130 M
ZVERIFY	Service Verify Method		100 M

Application Packages (PACKAGE) Class

File:	PRIMARY	Description:	The PACKAGE Class contains an instance with associated references used to identify an application component model. An administrator uses the Admin Publisher to create a package instance.
Domain:	SOFTWARE		
Class:	PACKAGE		
Icon:		Can connect to:	DESKTOP, FILE, PATH, MSFEATES, MSI, PACKAGE, PDACONFIG, REGISTRY

PACKAGE Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ALWAYS	Connect to		50

	Class connections determine the path of resolution of an agent's desired state. An _ALWAYS_ class connection is always resolved.		C
ALWAYS	Package Includes Class connections determine the path of resolution of an agent's desired state. An _ALWAYS_ class connection is always resolved.		50 C
BEHAVIOR	Behavior Includes connection to BEHAVIOR Class instances		255 I
COMPSize	Package Compressed Resource Size Compressed size of package (in bytes).		12 V
DESCRIPT	Application Description Description of the service that appears in the properties for the service in the Service List. Value is set initially in the Long Description field in the New Application Wizard (used to create a service).		255 V
DESKTOP	Desktop Includes connection to DESKTOP Class instances.		255 I
FILE	File Includes connection to FILE Class instances.		255 I
INCLUDES	Package Includes Generic Includes connection.		50 I
ITEMS	Number of Component instances Number of component instances in a package at time of publishing.		7 V
NAME	Friendly Name		80 V
OWNER	Application Contact Package owner, as documented by the Admin Publisher.		80 V
PATH	Path Includes connection to PATH Class instances.		255 I
PKGNAME	Package Name		80 V
REGISTRY	Registry Includes connection to REGISTRY Class instances.		255 I
RELEASE	Package Release Release level of the package, as documented by the Admin Publisher.		80 V

REQUIRES	Required Package Level Generic Requires connection.		80 R
SIZE	Package Resource Size– Uncompressed Size of package (in bytes).		12 V
UNIXFILE	UNIX File Includes connection to UNIXFILE Class instances.		255 I
ZBITARCH	Target Bit Architecture [/32/64] Set to 32 to deploy to 32-bit architectures. Set to 64 to deploy to 64-bit architectures. Set to blank to deploy to all architectures.	_UNDEF_	8 V
ZSTOP000 to ZSTOP003	Stop Resolution Expression Expressions evaluating to "true" in ZSTOP attributes cause resolution of the instance to be skipped. If left blank, the instance is not skipped, and resolution continues. This is useful for assigning a version to a specific set of users. Use the CSDB Editor to set this attribute.		255 E

Auto Run (EXECUTE) Class

File:	PRIMARY	Description:	An instance of the EXECUTE Class defines a file, or a group of files, that will be launched after an application has been installed on the agent desktop.
Domain:	SOFTWARE		
Class:	EXECUTE		
Icon:		Can connect to:	

EXECUTE Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ARGUMENT	Arguments passed to the program Stores the arguments that have to be passed to the program to be launched.		255 V
EXECPATH	Fully qualified executable name Stores the fully qualified name of the executable to be launched after an application install.		255 V
WORKDIR	Starting Directory Stores the name of the directory/folder from which the executable is started.		255 V
ZCREATE	Create Method	EXECUTE	50

			M
ZDELETE	Delete Method Sentient-supplied method or fully qualified, user-defined method.	EXECUTE	50 M
ZVERIFY	Verify Method	EXECUTE	50 M

Behavior Services (BEHAVIOR) Class

File:	PRIMARY	Description:	Defining an instance of the BEHAVIOR Class lets you define a file, or a group of files, to include in an application service. An instance of this class lets you specify such characteristics about the file as the file's name, size, and date stamp.
Domain:	SOFTWARE		
Class:	BEHAVIOR		
Icon:		Can connect to:	PACKAGE

BEHAVIOR Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ALWAYS	Connect to Class connections determine the path of resolution of an agent's desired state. An _ALWAYS_ class connection is always resolved.		50 C
CMPMD5	Compressed File MD5 Signature	_UNDEF_	32 V
FILEMD5	File MD5 Signature	_UNDEF_	32 V
LOCATION	Location		255 V
NAME	Friendly Name Name that appears in the CSDB Editor for this class.		80 V
OBJGUID	Instance GUID Object Global Unique Identification – <i>for future use</i> .	_UNDEF_	32 V
ZCMPKEY	Compression Key Not used.		80 V
ZCMPSIZE	Compressed File Size		11

	From promotion process if the compressed file option was selected. The size of the compressed file.		V
ZCREATE	Method to Install Behavior Name of method to install a resource file. Sentient-supplied method or fully qualified, user-defined method.		50 M
ZDELETE	Method to De-install Behavior Sentient-supplied method or a fully qualified, user-defined method.	EDMREMF	50 M
ZFILEUPD	Client File Update/Add Method If a file has been updated by an administrator, this method will be executed. Sentient-supplied method or fully qualified, user-defined method.		50 V
ZINIT	Resource Initialization Method Sentient-supplied method or a fully qualified, user-defined method.		50 M
ZOPENERR	Client Method on File Open Error Sentient-supplied method or a fully qualified, user-defined method.	EDMLKM	50
ZRSCCFIL	Behavior File Name Specific file to be distributed. A generic pattern (a wildcard character, such as *) or an explicit file name.		255 V
ZRSCCKPT	Resource Checkpoint: Not used. Replaced by ZRSCRSTR.		1 V
ZRSCCONF	Confirm File Download [Y/N] Y enables you to confirm that the resource was downloaded.	N	1 V
ZRSCCRC	Resource CRC From the promotion process if the CRC option was selected. Eight-character calculated hexadecimal value.		8 V
ZRSCCSTA	Client File Status Resource status code. 000 = Successfully downloaded 001 = Data can be found in EDMDATA directory 004 = Resource with no data 008 = Could not create the file 009 = No disk space for file (ZRSCMO=O) 010 = ZDLIMIT was exceeded (ZRSCMO=O) 011 = No data in Sentient for resource 014 = No disk space (ZRSCMO=M)	999	3 V

	was exceeded (ZRSCMO=M) 024 = Time-out waiting for method 036 = A method was set ZBRC>7 999 = No work done for resource		
ZRSCDATE	Behavior Date Stamp Date stamp on resource file from promote process. A value up to eight digits long.		8 V
ZRSCMDAT	Manager Stores Data [Y/N]	Y	1 V
ZRSCMMEM	PDS Member Name PDS where resource actually resides. PDS name where resource resides in Manager (MVS only).		8 V
ZRSCMO	Mandatory/Optional [M/O]	M	1 V
ZRSCNAME	Behavior Name Name or description of resource. Valid name of resource to distribute to the Agent.	&ZRSCCFIL	30
ZRSCOBJN	Persistent Object Name The name of the in-storage object. Valid only when the ZLOCMGR location is STORAGE.		8 V
ZRSCPRI	Create Sequence Priority Enables you to specify a priority level for this resource. If disk space on the client desktop is not available, resources will be downloaded based on the priority set here. 01 = Highest priority 10 = Default value 99 = Lowest priority	10	2 V
ZRSCRASH	DOS File Attribute [R/A/S/H] Enables you to specify file attributes for a resource. Read-only Archive System Hidden.		4 V
ZRSCRSTR	Restart [Y/N] You can either restart a download process at the point it was interrupted (Yes), or resend the entire object (No).	N	3 V
ZRSCSIZE	Behavior Size Byte count of resource file from promote		11 V

ZRSCSRC	Resource Source, i.e., Publisher		20 V
ZRSCSRC	Resource Source		1 V
ZRSCSTYP	Server File Type [BINARY/TEXT] File format for file download: Binary, Text, or Object.	BINARY	8 V
ZRSCSVRB	Version Skip Rebuild Flag [Y/N] Enables you to skip the reactivation of version resources if the version is already active. Y = Skip the rebuilding of version resources N = Rebuild version resources	Y	1 V
ZRSCTIME	Behavior Time Stamp Time stamp on resource file from promote process.		8 V
ZRSCVERS	Resource Version - From Promote Version number of resource from promotion process.		12 V
ZRSCVRFY	Verify Resource File on Connect Verifies the file statistics of a resource file on the desktop during each agent connect process. If you are deploying versioned resources, this attribute is ignored. The rules for the E(xist) flag apply to both Client Automation- and non-Sentient distributed files. (For example, a Client Automation-managed application file or CONFIG.SYS file.) Thus, you must specify Y or N for an existing file to be distributed. Y = The file will be distributed if one or more of the following conditions is met: The file has not been distributed before. The date, time, or size of the file has changed. The file was re-promoted. N = The file will be distributed if: The file has not been distributed before, Or, the file was re-promoted. E = The file will be distributed only if it does not currently exist on the desktop. M = Reserved for Client Automation maintenance files. During the last step of the connect, the file is moved from the EDMDATA subdirectory to its proper location. Example: can be used to update EDMGETRS.EXE.	Y	1 V

	S = This file is distributed if the file size has changed. T = The file is distributed if the file time has changed. D = File is distributed if the file date has changed. L = Forces the lock method to be used. The file will be automatically queued up to be deployed on restart, and the reboot flag will be set. If the L flag is not specified, and the file to be deployed is currently in use (locked) the lock method is automatically used anyway. What the L flag does is force the lock method to be always used to deploy this file, whether or not it is currently in use (locked). The L flag will also skip decompression and dispatch the open error method (RADLKM).		
	This is used for system dlls that must be replaced. Note that for the file to be verified during subsequent connects, you must also set this variable to Y. U = File is distributed only if the date and time on the Configuration Server is newer than the date and time on the desktop. R = Resource Stamp. For internal use only. V = (Windows only) Used in when normal date/time stamp check cannot be used. Usually when a .dll or .exe is involved and the date and time stamp are changing frequently. Radconct.exe checks the internal version number of the target file and compares it with the one that Client Automation has. If the incoming file has a higher internal version number than it will be placed on the desktop over the existing file. P = (Windows only) Set during packaging, enforces Windows System File Protection (SFP). If the verification is successful, the resource is in the desired state and no further verification is necessary. If the verification fails, Client Automation checks to see if System File Protection protects the file. If the file is protected, it will not be deployed. If the ZRSCVRFY attribute is set to P and deployment of this resource and the service are marked as failed due to SFP processing. If you do not enforce SFP, the file is marked as having passed verification.		

ZUPDATE	Client Instance Update Method If an object (resource instance) has been updated by an administrator, this method will be executed.		50 M
ZUSERID	Admin ID		30 V

Class Defaults (METACLAS) Class

File:	PRIMARY	Description:	An instance of the METACLAS Class defines prerequisites.
Domain:	SOFTWARE		
Class:	METACLAS		
Icon:		Can connect to:	PACKAGE

METACLAS Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ALWAYS	Connect to Class connections determine the path of resolution of an agent's desired state. An _ALWAYS_ class connection is always resolved.		50 C
CCREATE	Client Install Method		128 V
CDATAUPD	Client File Update/Add Method		128 V
CDELETE	Client De-install Method		128 V
CERROR	Client Error Method		128 V
CINIT	Client Initialization Method		128 V
COPENERR	Client Method on File Open Error		128 V
CUPDATE	Client Update Method		128 V
CVERIFY	Client Verify Method		128

			V
NAME	Friendly Name		80 V

Desktop (DESKTOP) Class

File:	PRIMARY	Description:	Instances of this class specify those resources which are icons appearing in the Windows environment. The attributes of this class also specify the properties of the Client Automation resources. Some of the icons could store just-in-time executables, which would be updated by connecting to the Configuration Server before they are run.
Domain:	SOFTWARE		
Class:	DESKTOP		
Icon:		Can connect to:	CATALOG, ZSERVICE

DESKTOP Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ALWAYS	Connect to Class connections determine the path of resolution of an agent's desired state. An _ALWAYS_ class connection is always resolved.		50 C
JITARGS	Application Manager Arguments Stores the arguments to be passed to the just-in-time executable.		255 V
JITLNK	Application Manager Installation [Y/N]		1 V
JITPATH	Application Manager Path Stores the fully qualified name of the just-in-time executable. Should be a fully qualified path name with the drive and directory. If the directory name contains embedded spaces then the whole value should be in double quotes, e.g., "c:\program files\HP\test.exe".		255 V
MACHUSER	Machine User		32 V
NAME	Friendly Name		80 V

VERIFY	Verify [Y/E/N]		1 V
ZCONTEXT	Component Context		8 V
ZCREATE	Method to Create Link Sentient-supplied method or fully qualified, user-defined method.	RADCLNK	50 M
ZDELETE	Method to Delete Link Sentient-supplied method or fully qualified, user-defined method.	RADDLNK	50 M
ZINSTALL	Method CLNK Processing [Y/N] Stores if the links/icons have to be installed.		1 V
ZOBJPRI	Priority [09-99]	60	2 V
ZPUBVER	Publisher Version	_UNDEF_	8 V
ZRSCLFIL	Link Path Stores the fully qualified name for the link. The links are usually created in the Windows folder, which might appear in different drives for different users. The Windows system directory is usually stored in the ZMASTER object in ZWINDRV & ZWINDIR variables.. A special variable, \$NTUSER\$, can be used to distinguish between different user directories. This variable is interpreted by the EDMCLNK create method to store the value of the current user.		255 V
ZRSCMO	Mandatory/Optional [M/O]	o	1 V
ZSTATUS	Client Status	999	3 V
ZW95ARGS	Command Line Parameters Stores the arguments to be passed to the executable specified in the ZW95PATH field.		255 V
ZW95DESC	Icon Description A brief description for the icon/link.		60 V
ZW95HOTK	Hot Key Assigned by MS Windows		12 V
ZW95ICON	Icon File		128 V

ZW95INDX	Index of Icon		60 V
ZW95PATH	Command Line Stores the name of the executable associated with this icon/link.		128 V
ZW95SHOC	Visibility Flag [00000000/00000001]	00000001	8 V
ZW95WDIR	Working Directory Stores the starting directory for the program associated with the icon or link. Should be a fully qualified path name with the drive and directory. If the directory name contains embedded spaces then the whole value should be in double quotes, e.g., "c:\program files\HP".		50 V

Dialog Services (DIALOG) Class

File:	PRIMARY	Description:	Defining an instance of the DIALOG Class lets you define an install process for an application and its characteristics to Client Automation. An instance of the DIALOG Class always includes the resource files that comprise it, which could be either the panels or scripts. The panels and scripts are used to initiate an interactive session, to gather information pertinent to installing a particular application.
Domain:	SOFTWARE		
Class:	DIALOG		
Icon:		Can connect to:	BEHAVIOR, INSTALL, PACKAGE, PANEL

DIALOG Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ALWAYS	Connect to Class connections determine the path of resolution of an agent's desired state. An _ALWAYS_ class connection is always resolved.		50 C
ALWAYS	Utility Resolution Method Any method you specify for this attribute is unconditionally executed by Sentient when this object is resolved. A valid method name, for example, ZSYSTEM.ZMETHOD.PUT_ZERROR_HIST.		51 M

AUTHOR	Author Name Name of the author of the dialogs.		50 V
CATALOG	Catalog Name The catalog name for this dialog.	&(CATALOG. CATNAME)	32 V
COMPSize	Application Size- Compressed Stores the compressed size of all the dialog components (informational). This value is filled in by the Admin Publisher when the dialogs are stored in the CSDB.		12 V
DESCRIPT	Application Description Description of the dialog that appears in the properties for the service in the Service List. Value is set initially in the Long Description field in the New Application Wizard (used to create a service).		255 V
INSTDATE	Installed Date Stores the date and time when the dialog was installed on the client desktop. This value is set by the Sentient user after a dialog has been installed. It is useful for reporting purposes. (informational). The date is in text form, as in: Jan 28, 1998 16:10:00.		32 V
NAME	Friendly Name Stores the dialog name.		80 V
PRICE	Price		10 V
PUBDATE	Published Date of Service Date service was published.		32 V
SCHEDOK	Update Schedule Locally [Y/N] Specifies whether the subscribers are allowed to change the update schedule for the service locally. Y = Subscriber is allowed to change the schedule. N = Configuration Server controls the update schedule.		1 V
SIZE	Application Size - Uncompressed		12 V
UPDDATE	Upgrade Date (Programmatic) Stores the date when the file was updated on the Configuration Server.		12 V
UPGDATE	When Application was Upgraded		12

	Sentient user after the dialogs have been installed. This date can be used for reporting purposes. The date is in text form, as in: Jan 28, 1998 16:10:00.		V
URL	WEB URL Name Stores the address of a Web page, where additional information is available about this application. http://www.company_name.com		255 V
VENDOR	Vendor Name		24 V
VERDATE	Verified Date of Service		32 V
ZAVIS	Available, Verified, Installed, Sync F The Agent manages and maintains this attribute to show the different states of the application in the Service List. Y = Yes N = No X = Unknown The four states are: Available indicates whether a service is available from the Configuration Server. Verified indicates whether a service has been verified. Installed indicates whether the service has been installed. Synchronized indicates whether the installed service has all of the latest changes from the Configuration Server.	YXNX	4 V
ZCREATE	Service Installation Method Sentient-supplied method or fully qualified, user-defined method.	50 M	
ZDELETE	Service Delete Method Sentient-supplied method or fully qualified, user-defined method.		50 M
ZINIT	Service Initialization Method A fully qualified, user-defined method.		50 M
ZSVCCONF	Confirm Service Install [Y/N] Enables you to confirm the service was downloaded.	N	1 V
ZSVCCSTA	Service Status on Client (999) Service status code. 000 = All actions for service completed	999	3 V

	014 = No disk space for at least one resource 015 = ZDLIMIT exceeded for at least one resource 024 = A method timed out 036 = A create method set ZBRC>7 900 = Not all files were verified. No work done for service		
ZSVCINST	Service Installation Date/Time		32 V
ZSVCMO	Mandatory/Optional Service [M/O/MO/OM] Indicates whether the service is mandatory or optional on the Agent. If you are using Application Manager <i>and</i> Application Self-service Manager, you could also specify mandatory and then optional, MO , or optional then mandatory, OM . The first character indicates how the application should be handled before installation. The second character indicates how the application should be handled after installation.	M	1 V
ZSVCNAME	Service Name/Description Name of the service, e.g., VB Calculator.		24 V
ZSVCPRI	Service Create Ordering [01-99] Enables you to specify a priority level for this service. If disk space on the agent desktop is not available, services will be downloaded based on the priority level set here. 01 = Highest priority 10 = Default value 99 = Lowest priority		2 V
ZSVCSEL	Update Force Bypass Remove [U/F/B/R] If ZSVCMO is optional, this attribute can be Update, Force, Remove, or Bypass. Update Force Remove Bypass		1 V
ZSVCUPD	Service Update Date/Time		32 V
ZSVCVERS	Version Activation Level Version number of resources or service from the promotion process.		12 V
ZUPDATE	Service Update Method Sentient-supplied method or fully qualified, user-defined method.		50 M

File Resources (FILE) Class

File:	PRIMARY	Description:	An instance of the FILE Class defines a file, or a group of files, to include in an application service. An instance of this class lets you specify such characteristics about the file as the file's name, size, and date stamp.
Domain:	SOFTWARE		
Class:	FILE		
Icon:		Can connect to:	

FILE Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ALWAYS	Connect to Class connections determine the path of resolution of an agent's desired state. An _ALWAYS_ class connection is always resolved.		50 C
FVERSION	File Version	_UNDEF_	30 V
LOCATION	Location Stores the location of the file. The location of the drive and directory is encapsulated in the PATH object. The value is filled by the publisher and contains value of the form &P (D001ABCDEF01_12345678) , which is an indirect reference to the PATH object. The &P triggers a special kind of variable substitution which looks for the drive and directory from the PATH object.		255 V
NAME	Friendly Name		80 V
OBJGUID	Instance GUID Object Global Unique Identification – <i>for future use</i> .	_UNDEF_	32 V
PVERSION	Promote Version	_UNDEF_	30 V
SIGTYPE	File Signature Type		3 V
ZCMPKEY	Compression Key Not used.	_UNDEF_	80 V

ZCMPSIZE	Compressed File Size From promotion process if the compressed file option was selected. The size of the compressed file.		11 V
ZCONTEXT	Component Context		8 V
ZCREATE	Method to Install Resource Sentient-supplied method or fully qualified, user-defined method.		50 M
ZDELETE	Method to De-install Resource Sentient-supplied method or fully qualified, user-defined method.	RADREMF	50 M
ZFILEUPD	Client File Update/Add Method Method that is executed if a file has been updated by an administrator. Sentient-supplied method or fully qualified, user-defined method		50 M
ZINIT	Resource Initialization Method Sentient-supplied method or fully qualified, user-defined method.		50 M
ZOBJPATCH	Perform Patching [Y/N]	Y	1 V
ZOPENERR	Client Method on File Open Error Sentient-supplied method or fully qualified, user-defined method.	RADLKM	50 M
ZPUBVER	Publisher Version	_UNDEF_	8 V
ZREPAIR	Client Instance Repair Method		50 M
ZRSCCFIL	Resource File Name Specific file to be distributed. A generic pattern (a wildcard character, such as *) or an explicit filename.		255 V
ZRSCCKPT	Resource Checkpoint Not used. Replaced by ZRSCRSTR.		1 V
ZRSCCONF	Confirm File Download [Y/N] Y enables you to confirm that the resource was downloaded.	Y	1 V
ZRSCCRC	Resource CRC From the promotion process if the CRC option was selected. Eight-character calculated hexadecimal value.		8 V

ZRSCCSTA	Client File Status Resource status code. 000 = Successfully downloaded 001 = Data can be found in EDMDATA directory 004 = Resource with no data 008 = Could not create the file 009 = No disk space for file (ZRSCMO=O) 010 = ZDLIMIT was exceeded (ZRSCMO=O) 011 = No data in Sentient for resource 014 = No disk space (ZRSCMO=M) 015 = ZDLIMIT was exceeded (ZRSCMO=M) 024 = Time-out waiting for method 036 = A method was set ZBRC>7 999 = No work done for resource	999	3 V
ZRSCDATE	Resource Date Stamp Date stamp on resource file from promote process.		8 V
ZRSCMMEM	PDS Member Name PDS where resource actually resides. PDS name where resource resides in Manager (MVS only).		8 V
ZRSCMO	Mandatory/Optional [M/O]	M	1 V
ZRSCNAME	Resource Name Name or description of resource. Valid name of resource to distribute to client.	&ZRSCCFIL	30 V
ZRSCOBJN	Persistent Object Name The name of the in-storage object. Valid only when the ZLOCMGR location is STORAGE. A valid in-storage object name.		8 V
ZRSCPADM	Admin ID		30 V
ZRSCPRI	Create Sequence Priority Enables you to specify a priority level for this resource. If disk space on the client desktop is not available, resources will be downloaded based on the priority set here. 01 = Highest priority 10 = Default value 99 = Lowest priority	50	2 V
ZRSCRASH	DOS File Attribute [R/A/S/H] Enables you to specify file attributes for a resource. Read-only		4 V

	Archive System and/or Hidden		
ZRSCRSTR	Restart [Y/N/R] Restart a download process at the point it was interrupted (Y), or resend the entire object (No). Set to (R) to bypass Windows file caching and write directly to disk (allowing checkpoint restart to operate correctly).	N	3 V
ZRSCSIG	File Signature		32 V
ZRSCSIZE	Resource Size Byte count of resource file from promote process.		11 V
ZRSCSRC	Resource Source, i.e., Publisher Stores the source of the resource: If resource was retrieved from Configuration Server, the value would be M. If resource existed on the desktop, the value would be E. If resource was retrieved from Stager, value would be S. These values are filled in by the Sentient user during an application installation. When the user uninstalls an application, only the files that were retrieved from the Configuration Server or the Staging Server would be deleted. The files that existed on the desktop prior to an application install would not be deleted.		20 V
ZRSCSTYP	Server File Type [BINARY/TEXT] File format for file download: Binary, Text, or Object.	BINARY	8 V
ZRSCSVRB	Version Skip Rebuild Flag [Y/N] Enables you to skip the reactivation of version resources if the version is already active. Y = Skip the rebuilding of version resources N = Rebuild version resources		1 V
ZRSCTIME	Resource Time Stamp Time stamp on resource file from promote process.		8 V
ZRSCVERS	Resource Version - From Promote Version number of resource from promotion process.		12 V

ZRSCVRFY	Verify Resource File on Connect Verifies the file statistics of a resource file on the desktop during each agent connect process. If you are deploying versioned resources, this attribute is ignored. The rules for the E(xist) flag apply to both Client Automation- and non-Sentient distributed files. (For example, a Client Automation-managed application file or CONFIG.SYS file.) Thus, you must specify Y or N for an existing file to be distributed. Y = The file will be distributed if one or more of the following conditions is met: - The file has not been distributed before. - The date, time, or size of the file has changed. - The file was re-promoted. N = The file will be distributed if: - The file has not been distributed before. - Or, The file was re-promoted. E = The file will be distributed only if it does not currently exist on the desktop. M = Reserved for Client Automation maintenance files. During the last step of the agent connect, the file is moved from the EDMDATA subdirectory to its proper location. Example: can be used to update EDMGETRS.EXE. S = This file is distributed if the file size has changed. T = The file is distributed if the file time has changed. D = File is distributed if the file date has changed. L = Forces the lock method to be used. The file will be automatically queued up to be deployed on restart, and the reboot flag will be set. If the L flag is not specified, and the file to be deployed is currently in use (locked) the lock method is automatically used anyway. What the L flag does is force the lock method to be always used to deploy this file, whether or not it is currently in use (locked). The L flag will also skip decompression and dispatch the open error method (RADLKM).	Y	1 V
	This is used for system dlls that must be replaced. Note that for the file to be verified during subsequent connects, you must also set this variable to Y. U = File is distributed only if the date and time		

	on the CONFIGURATION SERVER is newer than the date and time on the desktop. R = Resource Stamp. For internal use only. V = (Windows only) Used in when normal date/time stamp check cannot be used. Usually when a .dll or .exe is involved and the date and time stamp are changing frequently. Radconct.exe checks the internal version number of the target file and compares it with the one that Client Automation has. If the incoming file has a higher internal version number then it will be placed on the desktop over the existing file. P = (Windows only) Set during packaging, enforces Windows System File Protection (SFP). If the verification is successful, the resource is in the desired state and no further verification is necessary. If the verification fails, Client Automation checks to see if System File Protection protects the file. If the file is protected, it will not be deployed. If the ZRSCVRFY attribute is set to P and deployment of this resource and the service are marked as failed due to SFP processing. If you do not enforce SFP, the file is marked as having passed verification.		
ZUPDATE	Client Instance Update Method If an object (resource instance) has been updated by an administrator, this method will be executed.		50 M

HTTP Proxy (HTTP) Class

File:	PRIMARY	Description:	
Domain:	SOFTWARE		
Class:	HTTP		
Icon:		Can connect to:	

HTTP Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
HTTPADDR	WEB Server Host		255 V

HTTPLOG	HTTP Transaction Log [Y/N]	N	1 V
HTTPNVDS	Using Novadigm HTTP Server	N	1 V
HTTPPORT	Web Server Port	80	4 V
HTTPPROXO	Use Web server if proxy fails	N	1 V
HTTPPROXP	Proxy Server Port	80	4 V
HTTPPROXY	Proxy Server Host		255 V
HTTPVDIR	Web Server Virtual Directory	/RADIADB	255 V
ZPCPROTL	Protocol [HTTP]		4 V
ZSTGOPT	Enable Staging	1	1 V

IBM AIX Packages (AIX) Class

File:	PRIMARY	Description:	AIX operating systems only. This class is required by the Native Packaging Feature of the Batch Publisher.
Domain:	SOFTWARE		
Class:	AIX		
Icon:		Can connect to:	

AIX Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ALWAYS	Connect To Class connections determine the path of resolution of an agent's desired state. An _ALWAYS_ class connection is always resolved.		50 A
CMPMD5	Compressed File MD5 Signature	_UNDEF_	32 V
COMMIT	Commit Update [Y/N]	N	1 V

COREQ	AIX Package Corequisites		255 V
FILEMD5	File MD5 Signature	_UNDEF_	32 V
FORCE	Force (overwrite) installation [Y/N]	N	1 V
IFREQ	AIX Package If-Requisites		255 V
INSTREQ	AIX Package Installed Requisites		255 V
LOCATION	Location		255 V
NAME	Friendly Name	Defaults	80 V
OBJGUID	Instance GUID Object Global Unique Identification – <i>for future use</i> .	_UNDEF_	32 V
PKGDESC	Package Description		80 V
PREREQ	AIX Package Prerequisites		255 V
RELYONDB	Verify Error Code? [Y/N]		1 V
ZCMPsize	Compressed File Size From promotion process if the compressed file option was selected. The size of the compressed file.		11 V
ZCREATE	Method to Install Resource Sentient-supplied method or fully qualified, user-defined method.	hide nvdkit method	60 M
ZDELETE	Method to De-Install resource	hide nvdkit method	50 M
ZFILEUPD	Client File Update/Add Method		50 M
ZINIT	Resource Initialization Method		50 M
ZOPENERR	Client Method on File Open Error Sentient-supplied method or a fully qualified, user-defined method.		50 M

ZPERGUID	File Group Name		32 V
ZPERUID	File Owner Name		32 V
ZREPAIR	Method to Repair Resource	hide nvdkit method	60 M
ZRSCCFIL	AIX Package File Name Specific file to be distributed. A generic pattern (a wildcard character, such as *) or an explicit filename.	_BASE_ INSTANCE_	255 V
ZRSCCKPT	Resource Checkpoint		1 V
ZRSCCONF	Confirm File Download [Y/N]	Y	1 V
ZRSCCRC	Resource CRC		8 V
ZRSCCSTA	Client File Status	999	3 V
ZRSCDATE	Resource Date Stamp – From Promote		8 V
ZRSCFLEV	Filesets + Levels		255 V
ZRSCFSET	Filesets in the Package		255 V
ZRSCMO	Mandatory/Optional on client [M/O]	M	1 V
ZRSCNAME	AIX Package Name	_BASE_ INSTANCE_	255 V
ZRSCPADM	Admin ID		30 V
ZRSCPRI	Create Sequence Priority	50	2 V
ZRSCRASH	File Permissions		4 V
ZRSCRSTR	Restart [Y/N]	Y	3 V
ZRSCSIZE	Resource Size – From Promote		11 V

ZRSCSRC	Resource Source, i.e. Publisher		20 V
ZRSCSTYP	Server File Type [BINARY/TEXT]	BINARY	8 V
ZRSCSVRB	Version Skip Rebuild Flag [Y/N]		1 V
ZRSCTIME	Resource Time Stamp – From Promote		8 V
ZRSCVERS	Resource Version – From Promote		12 V
ZRSCVRFY	Verify Resource File on Connect	Y	8 V
ZSTATUS	Client Status	999	3 V
ZUPDATE	Client Instance Update Method	hide nvdkit method	60 M
ZVERIFY	Method to Verify Resource	hide nvdkit method	40 M

Install Options (INSTALL) Class

File:	PRIMARY	Description:	Defining an instance of the INSTALL Class lets you activate and configure the presentation of dialog panels which are used for the purposes of personalizing the delivery of an application along with its associated components.
Domain:	SOFTWARE		
Class:	INSTALL		
Icon:		Can connect to:	

INSTALL Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
APPNAME	Application Name		30 V
AUDIT	Search for Application File [Y/N]	N	1 V
COMP1SEL	Component 1 Selected [Y/N]	Y	1 V

COMP1TXT	Component 1 Description		20 V
COMP2SEL	Component 2 Selected [Y/N]	Y	1 V
COMP2TXT	Component 2 Description		20 V
COMP3SEL	Component 3 Selected [Y/N]	Y	1 V
COMP3TXT	Component 3 Description		20 V
COMP4SEL	Component 4 Selected [Y/N]	Y	1 V
COMP4TXT	Component 4 Description		20 V
COMPPIC	Components Picture (bmp)	&(WELCPIC)	12 V
COPYRITE	Copyright Text	@Copyright 1998	30 V
EULAPIC	EULA Picture (bmp)	&(WELCPIC)	12 V
EULATEXT	End User License Agreement Text	This software is protected by copyright laws. By accepting, you are agreeing with all terms included in the original program documentation.	200 V
IDIRNAME	Install Directory Name		100 V
IDIRPIC	Install Directory Picture (bmp)	&(WELCPIC)	12
KEYFILE	Search for File Name		12 V
NAME	Friendly Name		80 V
OPTS1SEL	Option 1 Selected [Y/N]	Y	1 V
OPTS1TXT	Option 1 Description		20 V

OPTS2SEL	Option 2 Selected [Y/N]	Y	1 V
OPTS2TXT	Option 2 Description		20 V
OPTS3SEL	Option 3 Selected [Y/N]	Y	1 V
OPTS3TXT	Option 3 Description		20 V
OPTS4SEL	Option 4 Selected [Y/N]	Y	1 V
OPTS4TXT	Option 4 Description		20 V
OPTSPIC	Options Picture (bmp)	&(WELCPIC)	12 V
SCANDRVS	Search Drives		32 V
SHOWCOMP	Show Components [Y/N]	N	1 V
SHOWEULA	Show End User License Agreement [Y/N]	Y	1 V
SHOWIDIR	Show Install Directory [Y/N]	N	1 V
SHOWOPTS	Show Options [Y/N]	N	1 V
SHOWTYPE	Show Install Type [Y/N]	N	1 V
SHOWUINF	Show User Info [Y/N]	N	1 V
SHOWWELC	Show Welcome (N disables all) [Y/N]	Y	1 V
TYPE1BTN	Type 1 Button Caption	Typical	10 V
TYPE1TXT	Type 1	Program will be installed with the most	100

		common options. Recommended for most users.	V
TYPE2BTN	Type 2 Button Caption	Custom	10 V
TYPE2TXT	Type 2 Description	You may choose the options you want to install. Recommended for advanced users.	100 V
TYPE3BTN	Type 3 Button Caption		10 V
TYPE3TXT	Type 3 Description		100 V
TYPEPIC	Install Type Picture (bmp)	&(WELCPIC)	12 V
TYPESEL	Type Selection	Typical	10 V
UINF1LBL	User Info 1 Label		12 V
UINF1TXT	User Info - Text 1		30 V
UINF2LBL	User Info 2 Label		12 V
UINF2TXT	User Info - Text 2		30 V
UINF3LBL	User Info 3 Label		12 V
UINF3TXT	User Info - Text 3		30 V
UINFNAME	User Info - Name		30 V
UINFPIC	User Info Picture (bmp)	&(WELCPIC)	12 V
WELCPIC	Welcome Picture (bmp)	instpic.bmp	12 V
ZOBJDATA	Search Operation is Scan [SCN]	SCN	3 V
ZRSCCDIR	Search Directory	*	255 V

ZRSCCDRV	Search Drive	&(ZMASTER.ZWINDRV)	19 V
ZRSCSTYP	Search File Type [BINARY]	BINARY	6 V

Linux RPM Packages (RPM) Class

File:	PRIMARY	Description:	Linux operating systems only. This class is required by the Native Packaging Feature of the Batch Publisher.
Domain:	SOFTWARE		
Class:	RPM		
Icon:		Can connect to:	

RPM Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ALWAYS	Connect To Class connections determine the path of resolution of an agent's desired state. An _ALWAYS_ class connection is always resolved.		50 A
CMPMD5	Compressed File MD5 Signature	_UNDEF_	32 V
CONTENTS	Required Packages Included in Tar		255 V
FILEMD5	File MD5 Signature	_UNDEF_	32 V
INSTOPTS	Package Install Options		60 V
LOCATION	Location		255 V
NAME	Friendly Name		80 V
OBJGUID	Instance GUID Object Global Unique Identification – <i>for future use.</i>	_UNDEF_	32 V
PKGARCH	RPM Package Architecture		32 V
PKGEOCH	RPM Package EPOCH		32

			V
PKGNAME	RPM Package Name		32 V
PKGREL	RPM Package Release		32 V
PKGSUMM	RPM Package Summary		100 V
PKGVER	RPM Package Version		32 V
RELYONDB	Verify Error Code? [Y/N]		1 V
REQCMDS	Requires Commands		255 V
REQLIBS	Requires Shared Libraries		255 V
REQPKGS	Requires Package		255 V
VRFYOPTS	Package Verify Options		60 V
ZCMPSIZE	Compressed File Size From promotion process if the compressed file option was selected. The size of the compressed file.		11 V
ZCREATE	Method to Install Resource Sentient-supplied method or fully qualified, user-defined method.	hide nvdkit method	60 M
ZDELETE	Method to De-Install resource	hide nvdkit method	50 M
ZFILEUPD	Client File Update/Add Method		50 M
ZINIT	Resource Initialization Method		50 M
ZOPENERR	Client Method on File Open Error Sentient-supplied method or a fully qualified, user-defined method.	RADLKM	50 M
ZPERGUID	File Group Name		32 V
ZPERUID	File Owner Name		32 V

ZREPAIR	Method to Repair Resource	hide nvdkit method	60 M
ZRSCCFIL	RPM Package File Name Specific file to be distributed. A generic pattern (a wildcard character, such as *) or an explicit filename.	_BASE_ INSTANCE_	255 V
ZRSCCKPT	Resource Checkpoint		1 V
ZRSCCONF	Confirm File Download [Y/N]	Y	1 V
ZRSCCRC	Resource CRC		8 V
ZRSCCSTA	Client File Status	999	3 V
ZRSCDATE	Resource Date Stamp – From Promote		8 V
ZRSCMO	Mandatory/Optional on client [M/O]	M	1 V
ZRSCNAME	RPM Package	_BASE_ INSTANCE_	255 V
ZRSCPADM	Admin ID		30 V
ZRSCPRI	Create Sequence Priority	50	2 V
ZRSCRASH	File Permissions		4 V
ZRSCRSTR	Restart [Y/N]	Y	3 V
ZRSCSIZE	Resource Size – From Promote		11 V
ZRSCSRC	Resource Source, i.e. Publisher		20 V
ZRSCSTYP	Server File Type [BINARY/TEXT]	BINARY	8 V
ZRSCSVRB	Version Skip Rebuild Flag [Y/N]		1 V
ZRSCTIME	Resource Time Stamp – From Promote		8 V

ZRSCVERS	Resource Version – From Promote		12 V
ZRSCVRFY	Verify Resource File on Connect	Y	8 V
ZSTATUS	Client Status	999	3 V
ZUPDATE	Client Instance Update Method	hide nvdkit method	60 M
ZVERIFY	Method to Verify Resource	hide nvdkit method	40 M

Mac Alias (MACALIAS) Class

File:	PRIMARY	Description:	Mac operating systems only. Instances of this class define Macintosh Aliases.
Domain:	SOFTWARE		
Class:	MACALIAS		
Icon:		Can connect to:	

MACALIAS Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ALWAYS	Connect to Class connections determine the path of resolution of an agent's desired state. An _ALWAYS_ class connection is always resolved.		50 C
JITARGS	Application Manager Arguments Stores the arguments to be passed to the just-in-time executable.		255 V
JITLNK	Application Manager Installation [Y/N]		2 V
JITPATH	Application Manager Path Stores the fully qualified name of the just-in-time executable. Should be a fully qualified path name with the drive and directory. If the directory name contains embedded spaces then the whole value should be in double quotes, e.g., "c:\program files\HP\test.exe".		255 V

MACHUSER	Machine User		32 V
NAME	Friendly Name		80 V
ZCREATE	Method to Create Link Sentient-supplied method or fully qualified, user-defined method.	RADCALIS	50 M
ZDELETE	Method to Delete Link Sentient-supplied method or fully qualified, user-defined method.	RADDALIS	50 M
ZINSTALL	Method CLNK Processing [Y/N]		1 V
ZMACCRTR	Macintosh File Creator of Alias		6 V
ZMACSRVR	Server Name of Alias Target		32 V
ZMACTRGT	Alias Target File Path		255 V
ZMACTYPE	Macintosh File Type of Alias		6 V
ZMACZONE	Network Zone of Alias Target		32 V
ZOBJPRI	Priority [00-99]	60	2 V
ZRSCAFIL	Alias Path		255 V
ZRSCMO	Mandatory/Optional [M/O]	O	1 V

Mac File Resources (MACFILE) Class

File:	PRIMARY	Description:	Mac operating systems only. An instance of the MACFILE Class defines a file, or a group of files, to include in an application service. An instance of this class lets you specify such characteristics about the file as the file's name, size, and date stamp.
Domain:	SOFTWARE		
Class:	MACFILE		
Icon:		Can connect to:	

MACFILE Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ALWAYS	Connect to Class connections determine the path of resolution of an agent's desired state. An _ALWAYS_ class connection is always resolved.		50 C
LINKDEST	Symbolic Link Destination		80 V
LOCATION	Location		255 V
NAME	Friendly Name	Defaults	80 V
OBJGUID	Instance GUID Object Global Unique Identification – <i>for future use</i> .	_UNDEF_	32 V
SIGTYPE	File Signature Type		3 V
ZCMPKEY	Compression Key		80 V
ZCMPSIZE	Compressed File Size From promotion process if the compressed file option was selected. The size of the compressed file.		11 V
ZCONTEXT	Component Context		8 V
ZCREATE	Method to Install Resource Sentient-supplied method or fully qualified, user-defined method.		50 M
ZDELETE	Method to De-install Resource Sentient-supplied method or fully qualified, user-defined method.	RADREMF	50 M
ZFILEUPD	Client File Update/Add Method If a file has been updated by an administrator, this method will be executed. Sentient-supplied method or fully qualified, user-defined method.		50 M
ZINIT	Resource Initialization Method A fully qualified, user-defined method.		50 M
ZOBJPTCH	Perform Patching [Y/N]	Y	1 V

ZOPENERR	Client Method on File Open Error Sentient-supplied method or a fully qualified, user-defined method.		50 M
ZPERGID	File Group Name		32 V
ZPERUID	File Owner Name		32 V
ZREPAIR	Client Instance Repair Method		50 V
ZRSCCFIL	Resource File Name Specific file to be distributed. A generic pattern (a wildcard character, such as *) or an explicit filename.	_BASE_ INSTANCE_	255 V
ZRSCCKPT	Resource Checkpoint		1 V
ZRSCCONF	Confirm File Download [Y/N] Y enables you to confirm that the resource was downloaded.	Y	1 V
ZRSCCRC	Resource CRC From the promotion process if the CRC option was selected. Eight-character calculated hexadecimal value.		8 V
ZRSCCSTA	Client File Status	999	3 V
ZRSCDATE	Resource Date Stamp Date stamp on resource file from promote process.		8 V
ZRSCMO	Mandatory/Optional [M/O]	M	1 V
ZRSCNAME	Resource Name Name or description of resource.	&ZRSCCFIL	30 V
ZRSCPADM	Admin ID		30 V
ZRSCPRI	Create Sequence Priority Enables you to specify a priority level for this resource. If disk space on the client desktop is not available, resources will be downloaded based on the priority set here. 01 = Highest priority 10 = Default value 99 = Lowest priority	50	2 V

ZRSCRASH	File Permissions		4 V
ZRSCRSTR	Restart [Y/N/R] Y to restart. Set to R to bypass Windows file caching and write directly to disk (allowing checkpoint restart to operate correctly).	Y	3 V
ZRSCSIG	File Signature		32 V
ZRSCSIZE	Resource Size Byte count of resource file from promote process.		11 V
ZRSCSRC	Resource Source, that is Publisher		20 V
ZRSCSTYP	Server File Type [BINARY/TEXT]	BINARY	8 V
ZRSCSVRB	Version Skip Rebuild Flag[Y/N]		1 V
ZRSCTIME	Resource Time Stamp Time stamp on resource file from promote process.		8 V
ZRSCVERS	Resource Version - From Promote Version number of resource from promotion process.		12 V
ZRSCVRFY	Verify Resource File on Connect Verifies the file statistics of a resource file on the desktop during each agent connect process. If you are deploying versioned resources, this attribute is ignored. The rules for the E(xist) flag apply to both Sentient- and non-Sentient distributed files. (For example, a Sentient-managed application file or CONFIG.SYS file.) Thus, you must specify Y or N for an existing file to be distributed. Y = The file will be distributed if one or more of the following conditions is met: The file has not been distributed before. The date, time, or size of the file has changed. The file was re-promoted. N = The file will be distributed if: The file has not been distributed before. Or, The file was re-promoted.	Y	8 V

	E = The file will be distributed only if it does not currently exist on the desktop. M = Reserved for Client Automation maintenance files. During the last step of the connect, the file is moved from the <code>EDMDATA</code> subdirectory to its proper location. Example: can be used to update <code>EDMGETRS.EXE</code>. S = This file is distributed if the file size has changed. T = The file is distributed if the file time has changed. D = File is distributed if the file date has changed. L = Forces the lock method to be used. The file will be automatically queued up to be deployed on restart, and the reboot flag will be set. If the L flag is not specified, and the file to be deployed is currently in use (locked) the lock method is automatically used anyway. What the L flag does is force the lock method to be always used to deploy this file, whether or not it is currently in use (locked). The L flag will also skip decompression and dispatch the open error method (RADLKM). This is used for system dlls that must be replaced.		
	Note that for the file to be verified during subsequent connects, you must also set this variable to Y. U = File is distributed only if the date and time on the CONFIGURATION SERVER is newer than the date and time on the desktop. R = Resource Stamp. For internal use only. V = (Windows only) Used in when normal date/time stamp check cannot be used. Usually when a <code>.dll</code> or <code>.exe</code> is involved and the date and time stamp are changing frequently. <code>Radconct.exe</code> checks the internal version number of the target file and compares it with the one that Client Automation has. If the incoming file has a higher internal version number than it will be placed on the desktop over the existing file. P = (Windows only) Set during packaging, enforces Windows System File Protection (SFP). If the verification is successful, the resource is in the desired state and no further verification is necessary. If the verification fails, Client Automation checks to see if System File Protection protects the file. If the file is protected,		

	it will not be deployed. If the ZRSCVRFY attribute is set to P and deployment of this resource and the service are marked as failed due to SFP processing. If you do not enforce SFP, the file is marked as having passed verification.		
ZSTATUS	Client Status	999	3 V
ZUPDATE	Client Instance Update Method		50 M

Mobile File Resources (RMMFILE) Class

File:	PRIMARY	Description:	Instances of this class are created when publishing files for Mobile Management using the Mobile Management Publisher.
Domain:	SOFTWARE		
Class:	RMMFILE		
Icon:		Can connect to:	

RMMFILE Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ALWAYS	Connect To Class connections determine the path of resolution of an agent's desired state. An _ALWAYS_ class connection is always resolved.		50 C
FVERSION	File Version	_UNDEF_	30 V
LOCATION	Location		255 V
NAME	Friendly Name	Defaults	80 V
OBJGUID	Instance GUID Object Global Unique Identification – <i>for future use.</i>	_UNDEF_	32 V
PVERSION	Promote Version	_UNDEF_	30 V
SIGTYPE	File Signature Type		3

			V
UPROIMPT	User Prompt [Y/N]	N	1 V
ZAPPNAME	PDA Application Name		50 V
ZCMPKEY	Compression Key	_UNDEF_	80 V
ZCMPSIZE	Compressed File Size From promotion process if the compressed file option was selected. The size of the compressed file.		11 V
ZCONTEXT	Component Context		8 V
ZCREATE	Method to Install Resource Sentient-supplied method or fully qualified, user-defined method.		50 M
ZDELETE	Method to De-install Resource	RADREMF	50 M
ZFILEUPD	Client File Update/Add Method		50 M
ZINIT	Resource Initialization Method		50 M
ZOBJPTCH	Perform Patching [Y/N]	Y	1 V
ZOPENERR	Client Method on File Open Error Sentient-supplied method or a fully qualified, user-defined method.	RADLKM	50
ZPDAVERS	PDA Resource Internal Version		20 V
ZPUBVER	Publisher Version	_UNDEF_	8 V
ZREPAIR	Client Instance Repair Method		50 M
ZRSCCFIL	Resource File Name Specific file to be distributed. A generic pattern (a wildcard character, such as *) or an explicit filename.		255 V
ZRSCCKPT	Resource Checkpoint		1

			V
ZRSCCONF	Confirm File Download [Y/N]	Y	1 V
ZRSCCPADM	Admin ID		30 V
ZRSCCRC	Resource CRC		8 V
ZRSCCSTA	Client file Status	999	3 V
ZRSCDATE	Resource Date Stamp – From Promote		8 V
ZRSCMMEM	PDS Member Name		8 V
ZRSCMO	Mandatory/Optional on Client [M/O]	M	1 V
ZRSCNAME	Resource Name	&ZRSCCFIL	30 V
ZRSCOBJN	Persistent Object Name		8 V
ZRSCPRI	Create Sequence Priority	50	2 V
ZRSCRASH	DOS File Attribute [R/A/S/H]		4 V
ZRSCRSCR	Restart [Y/N]	N	3 V
ZRSCSIG	File Signature		32 V
ZRSCSIZE	Resource Size – From Promote		11 V
ZRSCSRC	Resource Source, i.e., Publisher		20 V
ZRSCSTYP	Server File Type [BINARY/TEXT]	BINARY	8 V
ZRSCSVRB	Version Skip Rebuild Flag [Y/N]		1 V
ZRSCTIME	Resource Time Stamp – From Promote		8 V

ZRSCVERS	Resource Version – From Promote		12 V
ZRSCVRFY	Verify Resource File on Connect	Y	8 V
ZUPDATE	Client Instance Update Method		50 M

MSI Basic Resources (MSIBASIC) Class

File:	PRIMARY	Description:	Packages created with the Extensions for Windows Installer have a connection to the MSIBASIC class in the CSDB. When the New Application Wizard is used to create a service from an application with one of these connections, you can define the MSI basic behavior methods.
Domain:	SOFTWARE		
Class:	MSIBASIC		
Icon:		Can connect to:	

MSIBASIC Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ALWAYS	Connect To Class connections determine the path of resolution of an agent's desired state. An _ALWAYS_ class connection is always resolved.		50 C
BCREATE	Method Before Install		50 M
BDELETE	Method Before De-Install		50 M
BREPAIR	Method Before Repair		50 M
BUPDATE	Method Before Update		50 M
BVERIFY	Method Before Verify		50 M
FVERSION	File Version	_UNDEF_	30 V
LOCATION	Location		255

			V
NAME	Friendly Name		80 V
PRODGUID	MSI Product GUID		50 V
PVERSION	Promote Version	_UNDEF_	30 V
SIGTYPE	File Signature Type		3 V
ZCMPSIZE	Compressed File Size From promotion process if the compressed file option was selected. The size of the compressed file.		11 V
ZCONTEXT	Component Context		8 V
ZCREATE	Method to Install Resource Sentient-supplied method or fully qualified, user-defined method.		50 M
ZDELETE	Method to De-install Resource	RADREMF	50 M
ZINIT	Resource Initialization Method		50 M
ZOBJPTCH	Perform Patching [Y/N]	Y	1 V
ZOPENERR	Client Method on File Open Error Sentient-supplied method or a fully qualified, user-defined method.		50 M
ZREPAIR	Client Instance Repair Method		50 M
ZRSCCFIL	Resource File Name Specific file to be distributed. A generic pattern (a wildcard character, such as *) or an explicit file name.		255 V
ZRSCCRC	Resource CRC		8 V
ZRSCCSTA	Client file Status	999	3 V
ZRSCDATE	Resource Date Stamp – From Promote		8

			V
ZRSCMO	Mandatory/Optional on Client [M/O]	M	1 V
ZRSCPRI	Create Sequence Priority	50	2 V
ZRSCRASH	DOS File Attribute [R/A/S/H]		4 V
ZRSCRSCR	Restart [Y/N]	N	3 V
ZRSCSIG	File Signature		32 V
ZRSCSIZE	Resource Size – From Promote		11 V
ZRSCSRC	Resource Source, i.e., Publisher		20 V
ZRSCSTYP	Server File Type [BINARY/TEXT]	BINARY	8 V
ZRSCTIME	Resource Time Stamp – From Promote		8 V
ZRSCVRFY	Verify Resource File on Connect	Y	8 V
ZUPDATE	Client Instance Update Method		50 M
ZVERIFY	Client Instance Verify Method		50 M

MSI Features (MSIFEATS) Class

File:	PRIMARY	Description:	An instance of the MSIFEATS Class defines MSI feature states.
Domain:	SOFTWARE		
Class:	MSIFEATS		
Icon:		Can connect to:	

MSIFEATS Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type

ASTATES	Allowed States		100 V
DISABLED	Disabled [Y/N]		1 V
DSTATE	Default State		10 V
FEATID	Feature ID		38 V
FEATNAME	Feature Name		50 V
FEATSIZE	Feature Size		12 V
HIDE	Hide Features in CP A/R Dialog		1 V
ISTATE	Install State		10 V
OBJGUID	Instance GUID Object Global Unique Identification – <i>for future use.</i>	_UNDEF_	32 V
PARENTID	Parent Feature ID		38 V
SEQNO	Sequence Number		4 V

MSI Properties (MSIPROPS) Class

File:	PRIMARY	Description:	The MSIPROPS Class stores MSI public properties and their values.
Domain:	SOFTWARE		
Class:	MSIPROPS		
Icon:		Can connect to:	

MSIPROPS Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
DEFAULT	Default Property Value		255 V

DISABLED	Disabled [Y/N]		1 V
PROPNAME	Property Name		255 V
VALUE	Property Value		255 V
VERIFY	Verify Property Value [Y/N]	N	1 V

MSI Resources (MSI) Class

File:	PRIMARY	Description:	Instances in this class are used to define MSI resources.
Domain:	SOFTWARE		
Class:	MSI		
Icon:		Can connect to:	

MSI Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ALWAYS	Connect to Class connections determine the path of resolution of an agent's desired state. An _ALWAYS_ class connection is always resolved.		50 C
DATAURL	Administrative Control Point		255 V
LOCATION	Location		255 V
MSIDRIVR	Use Local AIP [Y/N]		1 V
NAME	Friendly Name	Defaults	80 V
OBJGUID	Instance GUID Object Global Unique Identification – <i>for future use</i> .	_UNDEF_	32 V
ORGNAME	Organization Name		100 V
PRODGUID	MSI Product GUID	_UNDEF_	50

			V
PRODVER	Product Version	_UNDEF_	10 V
SALVAGE	Salvage Cache Files[U/M/B/N] Salvage existing Windows Installer-enabled content. U – Unmanaged application salvage. Set to U if you want to salvage content that Sentient is not currently managing. M – Managed application salvage. Set to M to salvage content that Sentient is currently managing. B – Both. Set to B to salvage both managed and unmanaged applications. This is useful when you have clients with resources both managed and unmanaged by Sentient. N – No content is salvaged.	N	1 V
SAVETEMP	Preserve Internet Cache [Y/N]	N	1 V
SEQNO	Feature Sequence Number		3 V
SHOWINCP	Show in CP Add/Remove	Y	1 V
SIGTYPE	File Signature Type		3 V
USERCNTL	User Controls MSI Features [Y/N]		1 V
ZCMPKEY	Compression Key		80 V
ZCMPSIZE	Compressed File Size From promotion process if the compressed file option was selected. The size of the compressed file.		11 V
ZCONTEXT	Component Context [M/U]		8 V
ZCREATE	Service Installation Method Sentient-supplied method or fully qualified, user-defined method.		50 M
ZDELETE	Service Delete Method Sentient-supplied method or fully qualified, user-defined method.	_NONE_	50 M

ZFILEUPD	Client File Update/Add Method If a file has been updated by an administrator, this method will be executed. Sentient-supplied method or fully qualified, user-defined method.		50 M
ZINIT	Service Initialization Method A fully qualified, user-defined method.		50 M
ZOBJPTCH	Perform Patching [Y/N]	Y	1 V
ZOPENERR	Client Method on File Open Error Sentient-supplied method or a fully qualified, user-defined method.	_UNDEF_	50 M
ZPUBVER	Publisher Version		8 V
ZREPAIR	Service Repair Method		50 M
ZRSCCFIL	Resource File Name Specific file to be distributed. A generic pattern (a wildcard character, such as *) or an explicit filename.		255 V
ZRSCCKPT	Resource Checkpoint		1 V
ZRSCCONF	Confirm File Download [Y/N] Y enables you to confirm that the resource was downloaded.	Y	1 V
ZRSCCRC	Resource CRC From the promotion process if the CRC option was selected. Eight-character calculated hexadecimal value.		8 V
ZRSCCSTA	Client File Status	999	3 V
ZRSCDATE	Resource Date Stamp Date stamp on resource file from promote process.		8 V
ZRSCMMEM	PDS Member Name		8 V
ZRSCMO	Mandatory/Optional [M/O]	M	1 V
ZRSCNAME	Resource Name Name or description of resource.	&ZRSCCFIL	30 V

ZRSCOBJN	Persistent Object Name		8 V
ZRSCPADM	Admin ID		30 V
ZRSCPRI	Create Sequence Priority Enables you to specify a priority level for this resource. If disk space on the client desktop is not available, resources will be downloaded based on the priority set here. 01 = Highest priority 10 = Default value 99 = Lowest priority	50	2 V
ZRSCRASH	DOS File Attribute [R/A/S/H]	N	4 V
ZRSCRSTR	Restart [Y/N]		3 V
ZRSCSIG	File Signature		32 V
ZRSCSIZE	Resource Size Byte count of resource file from promote process.		11 V
ZRSCSRC	Resource Source, that is Publisher		20 V
ZRSCSTYP	Server File Type [BINARY/TEXT]	BINARY	8 V
ZRSCSVRB	Version Skip Rebuild Flag [Y/N]		1 V
ZRSCTIME	Resource Time Stamp Time stamp on resource file from promote process.		8 V
ZRSCVERS	Resource Version - From Promote Version number of resource from promotion process.		12 V
ZRSCVRFY	Verify Resource File on Connect Verifies the file statistics of a resource file on the desktop during each agent connect process. If you are deploying versioned resources, this attribute is ignored. The rules for the E (exist) flag apply to both Sentient - and non-Sentient		8 V

	distributed files. (For example, a Sentient n-managed application file or <code>CONFIG.SYS</code> file.) Thus, you must specify Y or N for an existing file to be distributed. Y = The file will be distributed if one or more of the following conditions is met: - The file has not been distributed before. - The date, time, or size of the file has changed. - The file was re-promoted. N = The file will be distributed if: The file has not been distributed before. Or, The file was re-promoted. E = The file will be distributed only if it does not currently exist on the desktop. M = Reserved for Client Automation maintenance files. During the last step of the connect, the file is moved from the <code>EDMDATA</code> subdirectory to its proper location. Example: can be used to update <code>EDMGETRS.EXE</code>. S = This file is distributed if the file size has changed. T = The file is distributed if the file time has changed. D = File is distributed if the file date has changed. L = Forces the lock method to be used. The file will be automatically queued up to be deployed on restart, and the reboot flag will be set. If the L flag is not specified, and the file to be deployed is currently in use (locked) the lock method is automatically used anyway. What the L flag does is force the lock method to be always used to deploy this file, whether or not it is currently in use (locked).		
	The L flag will also skip decompression and dispatch the open error method (RADLKM). This is used for system DLLs that must be replaced. Note that for the file to be verified during subsequent connects, you must also set this variable to Y. U = File is distributed only if the date and time on the Configuration Server is newer than the date and time on the desktop. R = Resource Stamp. For internal use only. V = (Windows only) Used in when normal date/time stamp check cannot be used. Usually when a <code>.dll</code> or <code>.exe</code> is involved and the date and time stamp are changing frequently.		

	Radconct.exe checks the internal version number of the target file and compares it with the one that Client Automation has. If the incoming file has a higher internal version number than it will be placed on the desktop over the existing file. P = (Windows only) Set during packaging, enforces Windows System File Protection (SFP). If the verification is successful, the resource is in the desired state and no further verification is necessary. If the verification fails, Client Automation checks to see if System File Protection protects the file. If the file is protected, it will not be deployed. If the ZRSCVRFY attribute is set to P and deployment of this resource and the service are marked as failed due to SFP processing. If you do not enforce SFP, the file is marked as having passed verification.		
ZUPDATE	Service Update Method		50 M
ZVERIFY	Service Verify Method		50 M

Panel Services (PANEL) Class

File:	PRIMARY	Description:	The PANEL Class contains the Client Automation panel objects used during Client Automation installation dialog processing. This class contains shipping dialog panel examples. Publish all custom dialog panels to this class.
Domain:	SOFTWARE		
Class:	PANEL		
Icon:		Can connect to:	

PANEL Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ALWAYS	Connect to Class connections determine the path of resolution of an agent's desired state. An _ALWAYS_ class connection is always resolved.		50 C
CMPMD5	Compressed File MD5 Signature	_UNDEF_	32 V
FILEMD5	File MD5 Signature	_UNDEF_	32

			V
NAME	Friendly Name	Defaults	80 V
OBJGUID	Instance GUID Object Global Unique Identification – <i>for future use.</i>	_UNDEF_	32 V
ZCMPKEY	Compression Key Not used.		80 V
ZCMPSIZE	Compressed File Size From promotion process if the compressed file option was selected. The size of the compressed file.		11 V
ZCREATE	Install Behavior Sentient-supplied method or fully qualified, user-defined method.		50 M
ZDELETE	Method to De-install Behavior Sentient-supplied method or fully qualified, user-defined method.	RADREMF	50 M
ZFILEUPD	Client File Update/Add Method If a file has been updated by an administrator, this method will be executed. Sentient-supplied method or fully qualified, user-defined method.		50 M
ZINIT	Resource Initialization Method Sentient-supplied method or fully qualified, user-defined method.		50 M
ZOPENERR	Client Method on File Open Error Sentient-supplied method or fully qualified, user-defined method.	RADLKM	50 M
ZRSCCFIL	Panel Name Specific file to be distributed. A generic pattern (a wildcard character, such as *) or an explicit filename.		255 V
ZRSCCKPT	Resource Checkpoint Not used. Replaced by ZRSCRSTR.		1 V
ZRSCCONF	Confirm File Download [Y/N] Y enables you to confirm that the resource was downloaded.	N	1 V
ZRSCCRC	Resource CRC From the promotion process if the CRC option was selected. Eight-character calculated hexadecimal value.		8 V

ZRSCCSTA	Client File Status Resource status code. 000 = Successfully downloaded 001 = Data can be found in EDMDATA directory 004 = Resource with no data 008 = Could not create the file 009 = No disk space for file (ZRSCMO=O) 010 = ZDLIMIT was exceeded (ZRSCMO=O) 011 = No data in Sentient for resource 014 = No disk space (ZRSCMO=M) 015 = Time-out waiting for method 036 = A method was set ZBRC>7 999 = No work done for resource	999	3 V
ZRSCDATE	Panel Date Stamp Date stamp on resource file from promote process.		8 V
ZRSCMDAT	Manager Stores Data [Y/N]	Y	1 V
ZRSCMMEM	PDS Member Name PDS where resource actually resides. PDS name where resource resides in Manager (MVS only).		8 V
ZRSCMO	Mandatory/Optional [M/O]	M	1 V
ZRSCNAME	Panel Name Name or description of resource. Valid name of resource to distribute to Agent.	&ZRSCCFIL	30 V
ZRSCOBJN	Persistent Object Name The name of the in-storage object. Valid only when the ZLOCMGR location is STORAGE.		8 V
ZRSCPRI	Create Sequence Priority Enables you to specify a priority level for this resource. If disk space on the client desktop is not available, resources will be downloaded based on the priority set here. 01 = Highest priority 10 = Default value 99 = Lowest priority		2 V
ZRSCRASH	DOS File Attribute Enables you to specify file attributes for a resource. Read-only Archive System and/or		4 V

	Hidden		
ZRSCRSTR	Restart [Y/N] You can either restart a download process at the point it was interrupted (Yes), or resend the entire object (No).	N	3 V
ZRSCSIZE	Panel Size Byte count of resource file from promote process.		11 V
ZRSCSRC	Resource Source, that is Publisher		20 V
ZRSCSRC	Resource Source		1 V
ZRSCSTYP	Server File Type [BINARY/TEXT] File format for file download: Binary, Text, or Object.	BINARY	8 V
ZRSCSVRB	Version Skip Rebuild Flag [Y/N] Enables you to skip the reactivation of version resources if the version is already active. Y = Skip the rebuilding of version resources. N = Rebuild version resources.	Y	1 V
ZRSCTIME	Panel Time Stamp Time stamp on resource file from promote process.		8 V
ZRSCVERS	Resource Version - From Promote Version number of resource from promotion process.		12 V
ZRSCVRFY	Verify Resource File on Connect Verifies the file statistics of a resource file on the desktop during each agent connect process. If you are deploying versioned resources, this attribute is ignored. The rules for the E (exist) flag apply to both Sentient- and non-Sentient distributed files. (For example, a Sentient-managed application file or CONFIG.SYS file.) Thus, you must specify Y or N for an existing file to be distributed. Y = The file will be distributed if one or more of the following conditions is met: The file has not been distributed before. The date, time, or size of the file has changed. The file was re-promoted. N = The file will be distributed if: The file was re-	Y	1 V

	promoted. E = The file will be distributed only if it does not currently exist on the desktop. M = Reserved for Client Automation maintenance files. During the last step of the agent connect, the file is moved from the <code>EDMDATA</code> subdirectory to its proper location. Example: can be used to update <code>EDMGETERS.EXE</code>. S = This file is distributed if the file size has changed. T = The file is distributed if the file time has changed. D = File is distributed if the file date has changed. L = Forces the lock method to be used. The file will be automatically queued up to be deployed on restart, and the reboot flag will be set. If the L flag is not specified, and the file to be deployed is currently in use (locked) the lock method is automatically used anyway. What the L flag does is force the lock method to be always used to deploy this file, whether or not it is currently in use (locked). The L flag will also skip decompression and dispatch the open error method (<code>RADLKM</code>).		
	This is used for system dlls that must be replaced. Note that for the file to be verified during subsequent connects, you must also set this variable to Y. U = File is distributed only if the date and time on the Configuration Server is newer than the date and time on the desktop. R = Resource Stamp. For internal use only. V = (Windows only) Used in when normal date/time stamp check cannot be used. Usually when a <code>.dll</code> or <code>.exe</code> is involved and the date and time stamp are changing frequently. <code>Radconct.exe</code> checks the internal version number of the target file and compares it with the one that Client Automation has. If the incoming file has a higher internal version number than it will be placed on the desktop over the existing file. P = (Windows only) Set during packaging, enforces Windows System File Protection (SFP). If the verification is successful, the resource is in the desired state and no further verification is necessary. If the verification fails,		

	Client Automation checks to see if System File Protection protects the file. If the file is protected, it will not be deployed. If the ZRSCVRFY attribute is set to P and deployment of this resource and the service are marked as failed due to SFP processing. If you do not enforce SFP, the file is marked as having passed verification.		
ZUPDATE	Client Instance Update Method If an object (resource instance) has been updated by an administrator, this method will be executed.		50 M
ZUSERID	Admin ID		30 V

Path (PATH) Class

File:	PRIMARY	Description:	This class stores the drive and directory required to install a resource. Packages can be relocated by updating instances of this class.
Domain:	SOFTWARE		
Class:	PATH		
Icon:		Can connect to:	CATALOG, ZSERVICE

PATH Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
DIR	Directory Stores the directory name where the application is installed. The directory name could be selected by the administrator, or it could be selected by the client during the dialog sessions that precede an application install. The value can be substituted from an agent object, or it can be a fully qualified name.		255 V
DRIVE	Drive Stores the drive where the application would be installed. The drive name could be selected by the administrator or it could be selected by the client during the dialog sessions that precede an application install. The value can be substituted from an agent object, or it can be a valid drive name on the Agent machine.		20 V

NAME	Friendly Name		255 V
OBJGUID	Instance GUID Object Global Unique Identification – <i>for future use.</i>	_UNDEF_	32 V
ZRSCMO	Mandatory/Optional [M/O]	M	1 V
ZRSCPRI	Priority Stores the Create priority. The path object should be one of the first objects built on the Agent during an application installation so that all other files, desktop instances, can refer to it. The lower priority instances, or classes, are installed first. Numeric value between 1 and 99.	5	2 V

Registry Resources (REGISTRY) Class

File:	PRIMARY	Description:	An instance of the REGISTRY Class defines a registry resource that is part of a PACKAGE.
Domain:	SOFTWARE		
Class:	REGISTRY		
Icon:		Can connect to:	

REGISTRY Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ALWAYS	Connect to Class connections determine the path of resolution of an agent's desired state. An _ALWAYS_ class connection is always resolved.		50 C
DELENUM	Delete Device Key under Enum\Root When set to Y, will delete the device key under ENUM which otherwise will not be deleted due to the appearance of two new values after reboot; values that are not managed by Client Automation. For example, [hkey_local_machine\system\currentcontrolset\enum\root\legacy_beep\0000\control] key will be deleted when delenum=y. This prevents "new hardware found" pop up panels when the service is deleted.	N	1 V

DELNTSCM	Delete SCM Keys/Values [Y/N]	Y	2 V
DEPTH	Registry Depth	&(ZCURRENT. ZCURDEPT)	20 V
ENUMPROC	Process Control Key under Enum\Root	N	1 V
LOCATION	Location		255 V
NAME	Friendly Name		80 V
OBJGUID	Instance GUID Object Global Unique Identification – <i>for future use.</i>	_UNDEF_	32 V
SIGTYPE	File Signature Type		3 V
ZCMPKEY	Compression Key		80 V
ZCMPSIZE	Compressed File Size From promotion process if the compressed file option was selected. The size of the compressed file.		11 V
ZCONTEXT	Component Context		8 V
ZCREATE	Method to Install Resource Sentient-supplied method or fully qualified, user-defined method.	RADREG32	50 M
ZDELETE	Method to De-install Resource Sentient-supplied method or fully qualified, user-defined method.	RADREG32	50 M
ZFILEUPD	Client File Update/Add Method If a file has been updated by an administrator, this method will be executed. Sentient-supplied method or fully qualified, user-defined method.		50 M
ZINIT	Resource Initialization Method A fully qualified, user-defined method.		50 M
ZOBJPRI	Priority [0000-99]	55	2 V
ZOPENERR	Client Method on File Open Error		50

	Sentient-supplied method or a fully qualified, user-defined method.		M
ZPUBVER	Publisher Version	_UNDEF_	8 V
ZRSCCFIL	Resource File Name Specific file to be distributed. A generic pattern (a wildcard character, such as *) or an explicit filename.		255 V
ZRSCCKPT	Resource Checkpoint		1 V
ZRSCCONF	Confirm File Download [Y/N] Y enables you to confirm that the resource was downloaded.	Y	1 V
ZRSCCRC	Resource CRC From the promotion process if the CRC option was selected. Eight-character calculated hexadecimal value.		8 V
ZRSCCSTA	Client File Status	999	3 V
ZRSCDATE	Resource Date Stamp Date stamp on resource file from promote process.		8 V
ZRSCMO	Mandatory/Optional [M/O]		1 V
ZRSCNAME	Resource Name Name or description of resource.	&ZRSCFIL	30 V
ZRSCPADM	Admin ID		30 V
ZRSCRASH	DOS File Attribute [R/A/S/H]		4 V
ZRSCRSTR	Restart [Y/N]	N	1 V
ZRSCSIG	File Signature		32 v
ZRSCSIZE	Resource Size – From Promote Byte count of resource file from promote process.		11 V
ZRSCSRC	Resource Source, i.e., Publisher		20 V

ZRSCSTYP	File Type [BINARY/TEXT/EDMFILE]	EDMFILE	8 V
ZRSCTIME	Resource Time Stamp – From Promote Time stamp on resource file from promote process.		8 V
ZRSCVERS	Resource Version - From Promote Version number of resource from promotion process.		12 V
ZRSCVRFY	Verify Resource File on Connect Verifies the file statistics of a resource file on the desktop during each agent connect process. If you are deploying versioned resources, this attribute is ignored. The rules for the E (exist) flag apply to both Sentient- and non-Sentient distributed files. (For example, a Sentient - managed application file or <code>CONFIG.SYS</code> file.) Thus, you must specify Y or N for an existing file to be distributed. Y = The file will be distributed if one or more of the following conditions is met: The file has not been distributed before. The date, time, or size of the file has changed. The file was re-promoted. N = The file will be distributed if: The file has not been distributed before. Or, The file was re-promoted. E = The file will be distributed only if it does not currently exist on the desktop. M = Reserved for Client Automation maintenance files. During the last step of the connect, the file is moved from the <code>EDMDATA</code> subdirectory to its proper location. Example: can be used to update <code>EDMGETRS.EXE</code>. S = This file is distributed if the file size has changed. T = The file is distributed if the file time has changed. D = File is distributed if the file date has changed. L = Forces the lock method to be used. The file will be automatically queued up to be deployed on restart, and the reboot flag will be set. If the L flag is not specified, and the file to be deployed is currently in use (locked) the lock method is automatically used anyway. What the L flag does is force the lock method	U	1 V

	to be always used to deploy this file, whether or not it is currently in use (locked). The L flag will also skip decompression and dispatch the open error method (RADLKM).		
	This is used for system dlls that must be replaced. Note that for the file to be verified during subsequent connects, you must also set this variable to Y. U = File is distributed only if the date and time on the Configuration Server is newer than the date and time on the desktop. R = Resource Stamp. For internal use only. V = (Windows only) Used in when normal date/time stamp check cannot be used. Usually when a .dll or .exe is involved and the date and time stamp are changing frequently. Radconct.exe checks the internal version number of the target file and compares it with the one that Client Automation has. If the incoming file has a higher internal version number than it will be placed on the desktop over the existing file. P = (Windows only) Set during packaging, enforces Windows System File Protection (SFP). If the verification is successful, the resource is in the desired state and no further verification is necessary. If the verification fails, Client Automation checks to see if System File Protection protects the file. If the file is protected, it will not be deployed. If the ZRSCVRFY attribute is set to P and deployment of this resource and the service are marked as failed due to SFP processing. If you do not enforce SFP, the file is marked as having passed verification.		
ZSTATUS	Client Status	999	3 V
ZUPDATE	Client Instance Update Method		50 M

Replace Application (REPLACE) Class

File:	PRIMARY
Domain:	SOFTWARE
Class:	REPLACE

REPLACE Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
NAME	Friendly Name		50 V
REPLACE1 Thru REPLACE9	Replace Service Name	_UNDEF_	32 V
REPOPT	Replace Action [DELETE/YIELD/COEXIST]	DELETE	8 V
ZSTOP000	Stop Expression		100 E

Scheduling (TIMER) Class

File:	PRIMARY	Description:	Use the TIMER Class to schedule service deployments. For more information, see <i>Sentient Enterprise Console Online Help</i> .
Domain:	SOFTWARE		
Class:	TIMER		
Icon:		Can connect to:	

TIMER Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ALWAYS	Connect to Class connections determine the path of resolution of an agent's desired state. An _ALWAYS_ class connection is always resolved.		50 C
APPSVC	Application		32 V
DOMAIN	Server Domain Name	&(ZMASTER .ZDOMNAME)	32 V
IPADDR	Server IP Address/Name	&(ZMASTER .ZIPADDR)	32 V
MGRNAME	Server Name	&(ZMASTER .ZMGRNAME)	32 V
NAME	Friendly Name		80 V

NETAVAIL	Check Network Availability [Y/N/W] Checks for network availability before executing the command		1 V
OBJGUID	Instance GUID Object Global Unique Identification – <i>for future use.</i>	_UNDEF_	32 V
PINGCNT	Number of Ping Attempts This attribute sets the number of ping attempts to be made by the Configuration Server.	3	2 V
PINGDLAY	Time between Pings (in msec) This attribute sets the amount of time between pings in msec.	2000	6 V
REQUEST	Application Request	"Apply Updates"	32 V
RUNSYNC	Synchronous Timer Execution [Y/N] This attribute sets the value of Yes or No for the synchronous timer execution. The default value is Yes.	Y	1 V
SOCKET	Server Socket Number	&(ZMASTER .ZDSTSOCK)	32 V
ZCHNNAME	Channel [DOMAIN] Name	&(ZSERVICE. ZOBJDOMN)	32 V
ZCREATE	Client Create Method Sentient-supplied method or fully qualified, user-defined method. This value is inherited from the base instance and should not be changed.	radtimeq	80 M
ZDELETE	Client Delete Method Sentient-supplied method or fully qualified, user-defined method. This value is inherited from the base instance and should not be changed.	radtimeq	80 M
ZNOPING	Stop RCS Detection [Y/N/W/S/P] Controls the automatic sensing of a network connection between the agent computer and the Configuration Server. If set to S or P , additionally, it will ping SAP entries according to their priorities, along with the configuration server.	Y	1 V

	An expired time will continually evaluate whether communications with the Configuration Server can be established. When communications are established, the command line associated with the time is executed. After executing the command line, the Scheduler service resumes normal evaluation of whether the timer has expired again. This attribute is especially useful for mobile users. Note: In order to use this attribute, you must add it to the TIMER Class template.		
ZOBJPRI	Priority [00-99] Sets the priority for deployment of the ZTIMEQ object. The ZTIMEQ object is deployed relative to the other elements being deployed during the agent connect. The elements with a priority number less than the value of ZOBJPRI are deployed before the ZTIMEQ object. A value of 90 is inherited from the base instance and should not be changed.	90	2 V
ZPRVNAME	Provider (Manager) Name	&(ZSERVICE .ZMGRNAME)	32 V
ZRSCCMDL	Command Line to Execute This attribute indicates the command line that is executed on the subscriber's computer when the timer expires.	radskman req=	255 V
ZSCHDEF	Time Parameter Indicates when the timer expires. The syntax varies depending on the frequency of expiration that can be DAILY, HOURLY, INTERVAL, NUMDAY, WEEKDAY, WEEKLY.		50 V
ZSCHFREQ	Frequency [PERIODIC/ONCE/RANDOM] Use this attribute to specify how often the timer should expire. ONCE = expire one time	PERIODIC	10 V

	PERIODIC = expire repeatedly RANDOM = expire in random intervals		
ZSCHMODE	Timer Owner [MANAGER/DEFAULT] This attribute specifies the timer owner. It is recommended that you leave the timer owner configuration value as DEFAULT.	DEFAULT	32 V
ZSCHTYPE	Type [IMMEDIATE/DEFERRED] <i>Used only when ZSCHFREQ = PERIODIC.</i> Set ZSCHTYPE to DEFERRED to indicate that the first time an event is attempted to be launched, it will be deferred until the <i>next</i> scheduled time, no matter when the timer instance is evaluated. This was designed to handle the case of a daily 4 AM (non-peak) scheduled event that is sent to the agent computer during the day. If it was not deferred, it would launch during the day instead of "waiting" until the next morning. Example 1: Suppose you create and deploy a timer with the ZSCHDEF = DAILY (&ZSYSDATE,4:00:00) If ZSCHTYPE = IMMEDIATE and it is: Before 4:00:00, the command in the instance will be executed the same day at 4:00:00 After 4:00:00, the command in the instance will be executed immediately If ZSCHTYPE = DEFERRED and it is: Before 4:00:00, the command in the instance will be executed the <i>next</i> day at 4:00:00 After 4:00:00, the command in the instance will be executed the <i>next</i> day at 4:00:00 Example 2: Suppose you create and deploy a timer with the ZSCHDEF = WEEKDAY (FRIDAY,4:00:00) If ZSCHTYPE = IMMEDIATE and it is: Not Friday or Friday and before 4:00:00, the command in the instance will be executed on Friday at 4:00:00 Friday and after 4:00:00, the command	DEFERRED	10 V

	in the instance will be executed immediately If ZSCHTYPE = DEFERRED and it is: Not Friday or Friday and before 4:00:00, the command in the instance will be executed a week later on Friday at 4:00:00 Friday and after 4:00:00, the command in the instance will be executed a week later on Friday at 4:00:00		
ZSTOP	Stop Expression Expressions evaluating to "true" in ZSTOP attributes cause resolution of the instance to be skipped. If left blank, the instance is not skipped, and resolution continues. This is useful for assigning a version to a specific set of users. Use the CSDB Editor to set this attribute.		255 E
ZSVCOID	Service ID This attribute specifies the object ID of the Application instance that this Scheduling instance is connected to. This value is inherited from the base instance and should not be modified.	&(ZSERVICE.ZOBJID)	80 V
ZUPDATE	Client Update Method The Scheduler UPDATE method that runs on the agent computer. This value is inherited from the base instance and should not be changed.	radtimeq	80 M
ZVERIFY	Client Verify Method	radtimeq	80 M

SD Packages (SD) Class

File:	PRIMARY	Description:	The SD Class applies to the HP-UX operating system. When you publish native UNIX software with Client Automation Native Packaging, the depot containing native software in compressed format is promoted to the SD Class, similar to the UNIXFILE Class. For more information, see the <i>Sentient Enterprise Console Online Help</i> .
Domain:	SOFTWARE		
Class:	SD		
Icon:		Can connect to:	

SD Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ALWAYS	Connect to Class connections determine the path of resolution of an agent's desired state. An _ALWAYS_ class connection is always resolved.		50 C
ADDDEPS	Auto-select Dependencies [Y/N]	N	1 V
AUTOBOOT	Allow Automatic Reboot [Y/N] Set to Y in SD Class instance in case you have wrapped SD software that contains reboot file set.	N	1 V
CMPMD5	Compressed File MD5 Signature	_UNDEF_	32 V
CONTENTS	Required Packages Included in Tar		255 V
COREQ	Corequisites Required		255 V
EXREQ	Exerequisites		255 V
FILEMD5	File MD5 Signature	_UNDEF_	32 V
INSTOPTS	Package Install Options		60 V
LOCATION	Location		255 V
NAME	Friendly Name	Defaults	80 V
OBJGUID	Instance GUID Object Global Unique Identification – <i>for future use</i> .	_UNDEF_	32 V
PREREQ	Prerequisites Required		255 V
RELYONDB	Verify Error Code? [Y/N]		1 V
ZCMPSIZE	Compressed File Size From promotion process if the compressed file option was selected. The size of the compressed		11 V

ZCREATE	Method to Install Resource Uses method "Harness" call. The client side script (<code>sd.tcl</code> or <code>svr4.tcl</code>) contains a native command call to install software.	hide nvdkit method	60 V
ZDELETE	Method to De-install Resource Uses method "Harness" call. The client side script (<code>sd.tcl</code> or <code>svr4.tcl</code>) contains a native command call to remove software. Note: On HP-UX, when a native software application is removed, the application files are deleted, but the directory structure will remain.	hide nvdkit method	50 M
ZFILEUPD	Client File Update/Add Method If a file has been updated by an administrator, this method will be executed. Sentient-supplied method or fully qualified, user-defined method.		50 M
ZINIT	Resource Initialization Method A fully qualified, user-defined method.		50 M
ZOPENERR	Client Method on File Open Error Sentient-supplied method or a fully qualified, user-defined method.		50 M
ZPERGID	File Group Name		32 V
ZPERUID	File Owner Name		32 V
ZREPAIR	Resource Repair Method Uses method "Harness" call. The client side script (<code>sd.tcl</code> or <code>svr4.tcl</code>) contains a native command call to repair installed software (reinstall).	hide nvdkit method	60 M
ZRSCCFIL	SD Depot File Name Specifies path to the file that is included in this instance. The file contains HP-UX or Solaris software.	_BASE_ INSTANCE_	255 V
ZRSCCKPT	Resource Checkpoint		1 V
ZRSCCONF	Confirm File Download [Y/N] Y enables you to confirm that the resource was downloaded.	Y	1 V

ZRSCCRC	Resource CRC From the promotion process if the CRC option was selected. Eight-character calculated hexadecimal value.		8 V
ZRSCCSTA	Client File Status	999	3 V
ZRSCDATE	Resource Date Stamp Date stamp on resource file from promote process.		8 V
ZRSCMO	Mandatory/Optional [M/O]	M	1 M
ZRSCNAME	SD Product Spec Specifies a string that is used by native methods to identify software contained in the published depot. This is the complete software spec on HP-UX (tag, version, architecture, and vendor) and the SVR4 package name on Solaris.	_BASE_ INSTANCE_	255 V
ZRSCPADM	Admin ID		30 V
ZRSCPRI	Create Sequence Priority Enables you to specify a priority level for this resource. If disk space on the client desktop is not available, resources will be downloaded based on the priority set here. 01 = Highest priority 10 = Default value 99 = Lowest priority	50	2 V
ZRSCRASH	File Permissions		4 V
ZRSCRSTR	Restart [Y/N]	Y	3 V
ZRSCSIZE	Resource Size – From Promote Byte count of resource file from promote process.		11 V
ZRSCSRC	Resource Source, i.e., Publisher		20 V
ZRSCSTYP	Server File Type [BINARY/TEXT]	BINARY	8 V
ZRSCSVRB	Version Skip Rebuild Flag [Y/N]		1 V

ZRSCTIME	Resource Time Stamp Time stamp on resource file from promote process.		8 V
ZRSCVERS	Resource Version - From Promote Version number of resource from promotion process.		12 V
ZRSCVRFY	Verify Resource File on Connect Verifies the file statistics of a resource file on the desktop during each agent connect process. If you are deploying versioned resources, this attribute is ignored. The rules for the E(xist) flag apply to both Sentient- and non-Sentient distributed files. (For example, a Sentient-managed application file or <code>CONFIG.SYS</code> file.) Thus, you must specify Y or N for an existing file to be distributed. Y = The file will be distributed if one or more of the following conditions is met: The file has not been distributed before. The date, time, or size of the file has changed. The file was re-promoted. N = The file will be distributed if: The file has not been distributed before. Or, The file was re-promoted. E = The file will be distributed only if it does not currently exist on the desktop. M = Reserved for Client Automation maintenance files. During the last step of the connect, the file is moved from the <code>EDMDATA</code> subdirectory to its proper location. Example: can be used to update <code>EDMGETRS.EXE</code> . S = This file is distributed if the file size has changed. T = The file is distributed if the file time has changed. D = File is distributed if the file date has changed. L = Forces the lock method to be used. The file will be automatically queued up to be deployed on restart, and the reboot flag will be set. If the L flag is not specified, and the file to be deployed is currently in use (locked) the lock method is automatically used anyway. What the L flag does is force the lock method to be always used to deploy this file, whether or not it is currently in use (locked). The L flag will also skip decompression and dispatch the open error	Y	8 V

	method (RADLKM).		
	This is used for system dlls that must be replaced. Note that for the file to be verified during subsequent connects, you must also set this variable to Y. U = File is distributed only if the date and time on the Configuration Server is newer than the date and time on the desktop. R = Resource Stamp. For internal use only. V = (Windows only) Used in when normal date/time stamp check cannot be used. Usually when a .dll or .exe is involved and the date and time stamp are changing frequently. Radconct.exe checks the internal version number of the target file and compares it with the one that Client Automation has. If the incoming file has a higher internal version number than it will be placed on the desktop over the existing file. P = (Windows only) Set during packaging, enforces Windows System File Protection (SFP). If the verification is successful, the resource is in the desired state and no further verification is necessary. If the verification fails, Client Automation checks to see if System File Protection protects the file. If the file is protected, it will not be deployed. If the ZRSCVRFY attribute is set to P and deployment of this resource and the service are marked as failed due to SFP processing. If you do not enforce SFP, the file is marked as having passed verification.		
ZSTATUS	Client Status	999	3 V
ZUPDATE	Client Instance Update Method Uses method "Harness" call. The client side script (sd.tcl or svr4.tcl) contains a native command call to update software.	hide nvdkit method	60 V
ZVERIFY	Method to Verify Resource Uses method "Harness" call. The client side script (sd.tcl or svr4.tcl) contains a native command call to verify installed software.	hide nvdkit method	40 M

Solaris Patches (SOLPATCH) Class

File:	PRIMARY	Description:	Solaris operating system only, This class is required by the Native Packaging Feature of the Batch Publisher.
Domain:	SOFTWARE		
Class:	SOLPATCH		
Icon:		Can connect to:	

SOLPATCH Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ALWAYS	Connect To Class connections determine the path of resolution of an agent's desired state. An _ALWAYS_ class connection is always resolved.		50 C
BACKUP	Backup Patched Files [Y/N]	Y	1 V
CLUSTER	Is this a Patch Cluster [Y/N]		1 V
CMPMD5	Compressed File MD5 Signature	_UNDEF_	32 V
CONTENTS	Required Packages Included in Tar		255 V
FILEMD5	File MD5 Signature	_UNDEF_	32 V
INCOMPS	Incompatible Patches		255 V
LOCATION	Location		255 V
NAME	Friendly Name		80 V
OBJGUID	Instance GUID Object Global Unique Identification – <i>for future use.</i>	_UNDEF_	32 V
OBSOLETE	Obsolete Patches		255 V

PKGS	Patched Packages		255 V
RELYONDB	Verify Error Code? [Y/N]		1 V
REQUIRES	Required Patches		255 V
UNCONINS	Unconditionally Install Patch [Y/N]	N	1 V
ZCMPSIZE	Compressed File Size From promotion process if the compressed file option was selected. The size of the compressed file.		11 V
ZCREATE	Method to Install Resource Sentient-supplied method or fully qualified, user-defined method.	hide nvdkit method	60 M
ZDELETE	Method to De-Install Resource	hide nvdkit method	50 M
ZFILEUPD	Client File Update/Add Method		50 M
ZINIT	Resource Initialization Method		50 M
ZOPENERR	Client Method on File Open Error Sentient-supplied method or a fully qualified, user-defined method.		50 M
ZPERGUID	File Group Name		32 V
ZPERUID	File Owner Name		32 V
ZREPAIR	Method to Repair Resource	hide nvdkit method	60 M
ZRSCCFIL	Patch Depot File Name: Specific file to be distributed. A generic pattern (a wildcard character, such as *) or an explicit filename.	_BASE_ INSTANCE_	255 V
ZRSCCKPT	Resource Checkpoint		1 V
ZRSCCONF	Confirm File Download [Y/N]	Y	1 V

ZRSCCRC	Resource CRC		8 V
ZRSCCSTA	Client File Status	999	3 V
ZRSCDATE	Resource Date Stamp – From Promote		8 V
ZRSCMO	Mandatory/Optional on Client [M/O]	M	1 V
ZRSCNAME	Solaris Patch ID	_BASE_ INSTANCE_	255 V
ZRSCPADM	Admin ID		30 V
ZRSCPRI	Create Sequence Priority	50	2 V
ZRSCRASH	File Permissions		4 V
ZRSCRSTR	Restart [Y/N]	Y	3 V
ZRSCSIZE	Resource Size – From Promote		11 V
ZRSCSRC	Resource Source, i.e., Publisher		20 V
ZRSCSTYP	Server File Type [BINARY/TEXT]	BINARY	8 V
ZRSCSVRB	Version Skip Rebuild Flag [Y/N]		1 V
ZRSCTIME	Resource Time Stamp – From Promote		8 V
ZRSCVERS	Resource Version – From Promote		12 V
ZRSCVRFY	Verify Resource File on Connect	Y	8 V
ZSTATUS	Client Status	999	3 V
ZUPDATE	Client Instance Update Method	hide nvdkit method	60 M
ZVERIFY	Method to Verify Resource	hide nvdkit method	40 M

SVR4 Packages (SVR4) Class

File:	PRIMARY	Description:	The SVR4 Class applies to the Solaris operating system. When you publish native UNIX software with Native Packaging, the depot containing native software in compressed format is promoted to the SVR4 Class, similar to the UNIXFILE Class. For more information, see the <i>Sentient Enterprise Console Online Help</i> .
Domain:	SOFTWARE		
Class:	SVR4		
Icon:	 SVR4	Can connect to:	

SVR4 Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ALWAYS	Connect to Class connections determine the path of resolution of an agent's desired state. An _ALWAYS_ class connection is always resolved.		50 C
ADMIN	Admin File Exists? [Y/N]		1 V
ADMINOBJ	Is this admin object? [Y/N]		1 V
AUTOBOOT	Allow automatic reboot [Y/N]		1 V
CMPD5	Compressed File MD5 Signature	_UNDEF_	32 V
CONTENTS	Required Packages Included in Tar		255 V
FILEMD5	File MD5 Signature	_UNDEF_	32 V
INCOMP	Incompatible Packages		255 V
LOCATION	Location		255 V
NAME	Friendly Name		80 V
OBJGUID	Instance GUID:	_UNDEF_	32

			V
PKGVER	Package Version		50 V
PKGREV	Package Revision		50 V
PREREQ	Prerequisites Required		255 V
RELYONDB	Verify Error Code? [Y/N]		1 V
RESPFILE	Response File Path		80 V
RESPOBJ	Is this response object? [Y/N]	N	1 V
REVERSE	Reverse Dependency Packages		255 V
RESPONSE	Response file exist? [Y/N]	N	1 V
ZCMPSIZE	Compressed File Size From promotion process if the compressed file option was selected. The size of the compressed file.		11 V
ZCREATE	Method to Install Resource Uses method "Harness" call. The client side script (<code>sd.tcl</code> or <code>svr4.tcl</code>) contains a native command call to install software.	hide nvdkit method	60 V
ZDELETE	Method to De-install Resource Uses method "Harness" call. The client side script (<code>sd.tcl</code> or <code>svr4.tcl</code>) contains a native command call to remove software. Note: On HP-UX, when a native software application is removed, the application files are deleted, but the directory structure will remain.	hide nvdkit method	50 M
ZFILEUPD	Client File Update/Add Method If a file has been updated by an administrator, this method will be executed. Sentient-supplied method or fully qualified, user-defined method.		50 M
ZINIT	Resource Initialization Method A fully qualified, user-defined method.		50 M

ZOPENERR	Client Method on File Open Error Sentient-supplied method or a fully qualified, user-defined method.		50 M
ZPERGID	File Group Name		32 V
ZPERUID	File Owner Name		32 V
ZREPAIR	Method to Repair Resource Uses method "Harness" call. The client side script (<code>sd.tcl</code> or <code>svr4.tcl</code>) contains a native command call to repair installed software (reinstall).	hide nvdkit method	60 M
ZRSCCFIL	SVR4 Depot File Name Specifies path to the file that is included in this instance. The file contains Solaris software.	_BASE_ INSTANCE_	255 V
ZRSCCKPT	Resource Checkpoint		1 V
ZRSCCONF	Confirm File Download [Y/N] Y enables you to confirm that the resource was downloaded.	Y	1 V
ZRSCCRC	Resource CRC From the promotion process if the CRC option was selected. Eight-character calculated hexadecimal value.		8 V
ZRSCCSTA	Client File Status	999	3 V
ZRSCDATE	Resource Date Stamp Date stamp on resource file from promote process.		8 V
ZRSCMO	Mandatory/Optional [M/O]	M	1 M
ZRSCNAME	SVR4 Package Tag Specifies a string that is used by native methods to identify software contained in the published depot. This is SVR4 package name.	_BASE_ INSTANCE_	255 V
ZRSCPADM	Admin ID		30 V
ZRSCPRI	Create Sequence Priority Enables you to specify a priority level for this resource. If disk space on the client desktop is	50	2 V

	01 = Highest priority 10 = Default value 99 = Lowest priority		
ZRSCRASH	File Permissions		4 V
ZRSCRSTR	Restart [Y/N]	Y	3 V
ZRSCSIZE	Resource Size – From Promote Byte count of resource file from promote process.		11 V
ZRSCSRC	Resource Source, i.e., Publisher		20 V
ZRSCSTYP	Server File Type [BINARY/TEXT]	BINARY	8 V
ZRSCSVRB	Version Skip Rebuild Flag [Y/N]		1 V
ZRSCTIME	Resource Time Stamp – From Promote Time stamp on resource file from promote process.		8 V
ZRSCVERS	Resource Version - From Promote Version number of resource from promotion process.		12 V
ZRSCVRFY	Verify Resource File on Connect Verifies the file statistics of a resource file on the desktop during each agent connect process. If you are deploying versioned resources, this attribute is ignored. The rules for the E(xist) flag apply to both Sentient- and non-Sentient distributed files. (For example, a Sentient-managed application file or CONFIG.SYS file.) Thus, you must specify Y or N for an existing file to be distributed. Y = The file will be distributed if one or more of the following conditions is met: The file has not been distributed before. The date, time, or size of the file has changed. The file was re-promoted. N = The file will be distributed if: The file has not been distributed before. Or, The file was re-promoted. E = The file will be distributed only if it does not	Y	8 V

	currently exist on the desktop. M = Reserved for Client Automation maintenance files. During the last step of the connect, the file is moved from the <code>EDMDATA</code> subdirectory to its proper location. Example: can be used to update <code>EDMGETRS.EXE</code>. S = This file is distributed if the file size has changed. T = The file is distributed if the file time has changed. D = File is distributed if the file date has changed. L = Forces the lock method to be used. The file will be automatically queued up to be deployed on restart, and the reboot flag will be set. If the L flag is not specified, and the file to be deployed is currently in use (locked) the lock method is automatically used anyway. What the L flag does is force the lock method to be always used to deploy this file, whether or not it is currently in use (locked). The L flag will also skip decompression and dispatch the open error method (RADLKM). This is used for system dlls that must be replaced.		
	Note that for the file to be verified during subsequent connects, you must also set this variable to Y. U = File is distributed only if the date and time on the Configuration Server is newer than the date and time on the desktop. R = Resource Stamp. For internal use only. V = (Windows only) Used in when normal date/time stamp check cannot be used. Usually when a <code>.dll</code> or <code>.exe</code> is involved and the date and time stamp are changing frequently. <code>Radconct.exe</code> checks the internal version number of the target file and compares it with the one that Client Automation has. If the incoming file has a higher internal version number than it will be placed on the desktop over the existing file. P = (Windows only) Set during packaging, enforces Windows System File Protection (SFP). If the verification is successful, the resource is in the desired state and no further verification is necessary. If the verification fails, Client Automation checks to see if System File Protection protects the file. If the file is protected, it will not be deployed. If the <code>ZRSCVRFY</code>		

	attribute is set to P and deployment of this resource and the service are marked as failed due to SFP processing. If you do not enforce SFP, the file is marked as having passed verification.		
ZSTATUS	Client Status	999	3 V
ZUPDATE	Client Instance Update Method Uses method "Harness" call. The agent side script (<code>sd.tcl</code> or <code>svr4.tcl</code>) contains a native command call to update software.	hide nvdkit method	60 V
ZVERIFY	Method to Verify Resource Uses method "Harness" call. The client side script (<code>sd.tcl</code> or <code>svr4.tcl</code>) contains a native command call to verify installed software.	hide nvdkit method	40 M

TPM Configuration (TPMCFG) Class

File:	PRIMARY	Description:	This class is reserved for future use.
Domain:	SOFTWARE		
Class:	TPMCFG		

TPMCFG Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
BOOTOPT	Boot option [NONE PROMPT REJECTPROM]		16 V
RBPATH	Backup Archive Path		255 V
STATE	TPM progress		32 V
TPMERF	Emergency Recovery File Path		255 V
TPMRES	TPM password Reset File Path		255 V
ZPWD1	Password 1 BIOS		255 V
ZPWD2	Password 2 TPM Owner Password		255 V

ZPWD3	Password 3 Emergency Recovery Token		255 V
ZPWD4	Password 4 Password Reset token		255 V

Unix File Resources (UNIXFILE) Class

File:	PRIMARY	Description:	An instance of the UNIXFILE Class defines a file, or a group of files, to include in an application service. An instance of this class lets you specify such characteristics about the file as the file's name, size, and date stamp.
Domain:	SOFTWARE		
Class:	UNIXFILE		
Icon:		Can connect to:	

UNIXFILE Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ALWAYS	Connect to Class connections determine the path of resolution of an agent's desired state. An _ALWAYS_ class connection is always resolved.		50 C
DIRPERMS	Perms for newly created directories Permissions assigned to newly created directories.		4 V
LOCATION	Location		255 V
NAME	Friendly Name	Defaults	80 V
OBJGUID	Instance GUID Object Global Unique Identification – <i>for future use</i> .	_UNDEF_	32 V
SIGTYPE	File Signature Type		3 V
ZCMPKEY	Compression Key		80 V
ZCMPSIZE	Compressed File Size From promotion process if the compressed file option was selected. The size of the compressed file.		11 V

ZCONTEXT	Component Context		8 V
ZCREATE	Method to Install Resource Sentient-supplied method or fully qualified, user-defined method.		50 M
ZDELETE	Method to De-install Resource Sentient-supplied method or fully qualified, user-defined method.	RADREMF	50 M
ZFILEUPD	Client File Update/Add Method If a file has been updated by an administrator, this method will be executed. Sentient-supplied method or fully qualified, user-defined method.		50 M
ZINIT	Resource Initialization Method A fully qualified, user-defined method.		50 M
ZOBJPTCH	Perform Patching [Y/N]	Y	1 V
ZOPENERR	Client Method on File Open Error Sentient-supplied method or a fully qualified, user-defined method.		50 M
ZPERGID	File Group Name		32 V
ZPERUID	File Owner Name		32 V
ZREPAIR	Client Instance Repair Method		50 V
ZRSCCFIL	Resource File Name Specific file to be distributed. A generic pattern (a wildcard character, such as *) or an explicit filename.	_BASE_ INSTANCE_	255 V
ZRSCCKPT	Resource Checkpoint		1 V
ZRSCCONF	Confirm File Download [Y/N] Y enables you to confirm that the resource was downloaded.	Y	1 V
ZRSCCRC	Resource CRC From the promotion process if the CRC option was selected. Eight-character calculated hexadecimal value.		8 V
ZRSCCSTA	Client File Status	999	3 V

ZRSCDATE	Resource Date Stamp Date stamp on resource file from promote process.		8 V
ZRSCMO	Mandatory/Optional [M/O]	M	1 V
ZRSCNAME	Resource Name Name or description of resource.	&ZRSCCFIL	30 V
ZRSCPADM	Admin ID		30 V
ZRSCPRI	Create Sequence Priority Enables you to specify a priority level for this resource. If disk space on the client desktop is not available, resources will be downloaded based on the priority set here. 01 = Highest priority 10 = Default value 99 = Lowest priority	50	2 V
ZRSCRASH	File Permissions		4 V
ZRSCRSTR	Restart [Y/N/R] Y to restart. Set to R to bypass Windows file caching and write directly to disk (allowing checkpoint restart to operate correctly).	Y	3 V
ZRSCSIG	File Signature		32 V
ZRSCSIZE	Resource Size – From Promote Byte count of resource file from promote process.		11 V
ZRSCSRC	Resource Source, i.e., Publisher		20 V
ZRSCSTYP	Server File Type [BINARY/TEXT]	BINARY	8 V
ZRSCSVRB	Version Skip Rebuild Flag[Y/N]		1 V
ZRSCTIME	Resource Time Stamp – From Promote Time stamp on resource file from promote process.		8 V
ZRSCVERS	Resource Version - From Promote Version number of resource from promotion process.		12 V

ZRSCVRFY	Verify Resource File on Connect Verifies the file statistics of a resource file on the desktop during each agent connect process. If you are deploying versioned resources, this attribute is ignored. The rules for the E(xist) flag apply to both Sentient- and non-Sentient distributed files. (For example, a Sentient-managed application file or <code>CONFIG.SYS</code> file.) Thus, you must specify Y or N for an existing file to be distributed. Y = The file will be distributed if one or more of the following conditions is met: - The file has not been distributed before. - The date, time, or size of the file has changed. - The file was re-promoted. N = The file will be distributed if: The file has not been distributed before. Or, The file was re-promoted. E = The file will be distributed only if it does not currently exist on the desktop. M = Reserved for Client Automation maintenance files. During the last step of the connect, the file is moved from the <code>EDMDATA</code> subdirectory to its proper location. Example: can be used to update <code>EDMGETRS.EXE</code>. S = This file is distributed if the file size has changed. T = The file is distributed if the file time has changed. D = File is distributed if the file date has changed. L = Forces the lock method to be used. The file will be automatically queued up to be deployed on restart, and the reboot flag will be set. If the L flag is not specified, and the file to be deployed is currently in use (locked) the lock method is automatically used anyway. What the L flag does is force the lock method to be always used to deploy this file, whether or not it is currently in use (locked). The L flag will also skip decompression and dispatch the open error method (RADLKM). This is used for system dlls that must be replaced.	Y	8 V
	Note that for the file to be verified during subsequent connects, you must also set this variable to Y. U = File is distributed only if the date and time on the Configuration Server is newer than the		

	date and time on the desktop. R = Resource Stamp. For internal use only. V = (Windows only) Used in when normal date/time stamp check cannot be used. Usually when a .dll or .exe is involved and the date and time stamp are changing frequently. Radconct.exe checks the internal version number of the target file and compares it with the one that Client Automation has. If the incoming file has a higher internal version number than it will be placed on the desktop over the existing file. P = (Windows only) Set during packaging, enforces Windows System File Protection (SFP). If the verification is successful, the resource is in the desired state and no further verification is necessary. If the verification fails, Client Automation checks to see if System File Protection protects the file. If the file is protected, it will not be deployed. If the ZRSCVRFY attribute is set to P and deployment of this resource and the service are marked as failed due to SFP processing. If you do not enforce SFP, the file is marked as having passed verification.		
ZSTATUS	Client Status	999	3 V
ZUPDATE	Client Instance Update Method		50 M

Version Groups (VGROUP) Class

File:	PRIMARY	Description:	Each instance of the Version Group (VGROUP) class defines a set of versions for an application. This class contains connections to the Versions (VERSION) class, created using the Version Group Editor in the CSDB Editor.
Domain:	SOFTWARE		
Class:	VGROUP		
Icon:		Can connect to:	

VGROUP Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ACTDATE	Activated date/time		32

	Data maintained by the Agent in the VGROUPOBJECT on the agent computer. Do not enter a value. This is set by Agent.		V
CONTROL	Vgroup Control [CLIENT/MANAGER] Indicates whether the Client Automation administrator (MANAGER) or the subscriber (CLIENT) controls which version to activate on the agent computer.	MANAGER	8 V
CURVERS	Currently active version Data maintained by the Agent in the VGROUPOBJECT on the agent computer. Do not enter a value. This is set by Agent.		32 V
INITIAL	Initial version to Activate Indicates which version to activate on the agent computer.		32 V
NAME	Friendly Name The friendly name for the VGROUPOBJECT instance.		255 V
NEXTVERS	Next version to activate Data maintained by the Agent in the VGROUPOBJECT on the agent computer. Do not enter a value. This is set by Agent.		32 V
REQACTDT	Requested Activation Date YYYYMMDD The earliest date on which a version in this version group will be activated on any agent computer.		8 V
REQACTTM	Requested Activation time HH:MM:SS The earliest time, on the date specified by the REQACTDT attribute, after which a version in the version group will be activated on any agent computer.		8 V
ROLLBACK	Rollback if install fails [Y/N] Indicates whether to automatically roll back to the previously activated version when deployment of a new version fails.	Y	1 V
SOURCE	Source for currently active version Data maintained by the Agent in the VGROUPOBJECT on the agent computer. Do not enter a value. This is set by the Agent.		8 V
STATUS	Status of current version – 000 Data maintained by the Agent in the VGROUPOBJECT on the agent computer. Do not enter a value. This is set by the Agent.		3 V

TIMERCON	Timer for next Version Activation If you specify a "next version to activate," the CSDB Editor automatically creates a timer and stores the connection to that timer in this attribute.		50 C
VERCON01 to VERCON05	Version Connection Connects to each version in the version group. Each VERCON0n attribute contains a connection to one instance of the VERSION Class. This is set when you assign a version to the version group in the Version Group Editor.		50 C
ZSTOP001 to ZSTOP003	Stop Resolution Expression Expressions evaluating to "true" in ZSTOP attributes cause resolution of the instance to be skipped. If left blank, the instance is not skipped, and resolution continues. This is useful for assigning a version to a specific set of users. Use the CSDB Editor to set this attribute.		255 E

Versions (VERSION) Class

File:	PRIMARY	Description:	Each instance of the VERSION Class defines one <i>version</i> of an application to be deployed and managed by Client Automation. Use the Version Group Editor to create Versions class instances and assign them to a Version Group.
Domain:	SOFTWARE		
Class:	VERSION		
Icon:		Can connect to:	

VERSION Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
NAME	Friendly Name The friendly name for the VERSION instance. This is set when you create the instance using the Version Group Editor.		255 V
PACKAGE	Package Connection Connects to a PACKAGE Class instance, which represents the packaged software for this version.		50 A
ZSTOP001 to ZSTOP003	Stop Resolution Expression Expressions evaluating to "true" in ZSTOP attributes cause resolution of the instance to be skipped. If left blank, the instance is not skipped, and resolution		255 E

WTS App Comp Scripts (WTSACS) Class

File:	PRIMARY	Description:	This class contains one instance for each Windows Terminal Server Application Compatibility Script required for an application. ACS are published into the database by an administrator. Use the instances in this class to configure the ACS.
Domain:	SOFTWARE		
Class:	WTSACS		
Icon:		Can connect to:	

WTSACS Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
BDELETE	Method Before Delete		50 M
LOCATION	Location	&(ZMASTER.ZLIBDRV)& (ZMASTER.ZLIBDRV)	255 V
NAME	Friendly Name	Defaults	80 V
TYPE	Install, Logon, Uninstall (I/L/U)		1 V
ZCMPSIZE	Compressed File Size		11 V
ZCREATE	Method to Install Resource		50 M
ZDELETE	Method to De-install Resource	RADREMF	50 M
ZRSCCFIL	Resource File Name		255 V
ZRSCCSTA	Client File Status	999	3 V
ZRSCDATE	Resource Date Stamp – From Promote		8 V
ZRSCNAME	Resource Name	&ZRSCCFIL	30 V
ZRSCPRI	Create Sequence Priority	50	2 V

ZRSCRASH	DOS File Attribute [R/A/S/H]		4 V
ZRSCSIZE	Resource Size – From Promote		11 V
ZRSTYP	Server File Type [BINARY/TEXT]	BINARY	8 V
ZRSTIME	Resource Time Stamp – From Promote		8 V
ZRSCVRFY	Verify Resource Files on Connect	Y	8 V
ZSTOP000	Stop Expression	/(EDMGETV (ZCONFIG.WTSSRVR='Y'))	50 E

WTS Control (WTCTRL) Class

File:	PRIMARY	Description:	Instances in this class describe some behaviors you may need for the application, if you require logoffs for application events, the type of install and the timeout settings.
Domain:	SOFTWARE		
Class:	WTCTRL		
Icon:		Can connect to:	

WTCTRL Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
LD	Logoff For Delete [Y/N]	Y	1 V
LI	Logoff For Install [Y/N]	N	1 V
LR	Logoff For Repair [Y/N]	Y	1 V
LU	Logoff For Update [Y/N]	Y	1 V
NAME	Friendly Name	Default	80 V
RW	Rewarn of Logoff at [minutes]	2	2 V

TO	Disconnect Time Out [minutes]	15	4 V
WTSOPTNS	WTS Logoff Settings	LI=& (LI),LU=& (LU),LR=& (LR), LD=& (LD),TO=& (TO),RW=& (RW)	60 V
WTSTYPE	Install / Execute Mode [I/E] Set to I to install files natively on the Windows Terminal Server such as applications that an administrator may use specifically on the server. Set to E for an application that will be used by the Window Terminal Server's agents.	E	1 V

The SYSTEM Domain

The SYSTEM Domain contains administrative and process control definitions. These commands, methods, processes, tables, and system files are used for deploying applications.

Application Manager (ZCOMMAND) Class

File:	PRIMARY	Description:	Create an instance of the ZCOMMAND Class to define a customized command.
Domain:	SYSTEM		
Class:	ZCOMMAND		
Icon:		Can connect to:	

ZCOMMAND Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
NAME	Friendly Name		32 V
ZCMDDESC1	Command Description 1		255 V
ZCMDDESC2	Command Description 2		255 V
ZCMDNAME	Command Name		8

			V
ZCMDOPTN	Command Option		8 V
ZCMDPATH	Fully Qualified Path for CMD Location of the command.		255 V
ZCMDPRMS	Parameters Passed to the Command		255 V
ZCMDSEP	Separator to Use for the Parameters Separator used for parameters in user-defined commands.		1 V
ZCMDSYNC	Synchronize Flag [Y/N]	N	1 V
ZCMDTYPE	Command Type [REXX/EXE]		8 V
ZCMDUCLS	USER Class Name USER Class name. This is the name of the class in which to look for users connected to the dropee. For example, if the value is set to COMPUTERS and the dropee is WORKGRP.ACCOUNTING, instances of the COMPUTERS Class that are members of WORKGRP.ACCOUNTING will be the selected audience for the notification. If ZCMDUCLS is not specified, then (using the above example) the audience will be created by instances of the COMPUTERS Class that are members of WORKGRP.ACCOUNTING. The default for ZCMDUCLS is USER.	USER	8 V

Consoles (ZCONSOLE) Class

File:	PRIMARY	Description:	Reserved for future use.
Domain:	SYSTEM		
Class:	ZCONSOLE		
Icon:		Can connect to:	

ZCONSOLE Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ZCOMMAND	Console Operation		8 V

DB Version (DBVER) Class

File:	PRIMARY	Description:	Defines the version of the CSDB.
Domain:	SYSTEM		
Class:	DBVER		
Icon:		Can connect to:	

DBVER Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
DBVER	RCA Template DB Version	7.50.000	50 V

LICENSE (LICENSE) Class

File:	PRIMARY	Description:	This class is reserved for future use.
Domain:	SYSTEM		
Class:	LICENSE		

LICENSE Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
KEY	License Key		128 V

Methods (ZMETHOD) Class

File:	PRIMARY	Description:	Defines how to invoke an external method or program.
Domain:	SYSTEM		
Class:	ZMETHOD		
Icon:		Can connect to:	

ZMETHOD Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type

DESCRIPT	Method Description Stores a brief description of the method (informational). Value is set initially in the Long Description field in the New Application Wizard (used to create a service).	Manager Method &ZMTHNAME	70 V
ZMTHDSC1	Method Description 1		255 V
ZMTHDSC2	Method Description 2		255 V
ZMTHMODE	Mode [INTERNAL/EXTERNAL]	EXTERNAL	8 V
ZMTHNAME	Member Name of Method		8 V
ZMTHPRMS	Parameters Passed to Method		80 V
ZMTHSYNC	Synchronization Flag [Y/N]	Y	1 V
ZMTHTYPE	Method Type [REXX/ASM/EXE]		8 V
ZMUSTRUN	Return Code critical to Resolution?	Y	1 V
ZSTOP001	Stop Method		255 E

RCA Intent Class (ZINTENT) Class

File:	PRIMARY	Description:	Allows resolution to take different paths depending on the SESSION.INTENT (self- maintenance, catalog resolution, service resolution, etc...).
Domain:	SYSTEM		
Class:	ZINTENT		
Icon:		Can connect to:	

ZINTENT Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ALWAYS	Method		80 M
ALWAYS	Connect To		80

			C
ALWAYS	Connect To	SYSTEM.ZPRODUCT. _NULL_INSTANCE_	80 C
DESCRIPT	Process Description		32 V
ZMAXOKRC	Max acceptable method Return Code	008	3 V

RCA Processes (PROCESS) Class

File:	PRIMARY	Description:	Defines what actions to take when a client object is received.
Domain:	SYSTEM		
Class:	PROCESS		
Icon:		Can connect to:	

PROCESS Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ALWAYS	Method Any method you specify for this attribute is unconditionally executed by Client Automation when this instance is resolved.		50 M
ALWAYS	Connect To Class connections determine the path of resolution of an agent's desired state. An _ALWAYS_ class connection is always resolved.		50 C
DESCRIPT	Process Description Stores a brief description of the process (informational). Value is set initially in the Long Description field in the New Application Wizard (used to create a service).	Processing Client request for &ZCUROBJ	80 V
ZMAXOKRC	Maximum Acceptable Method Return Code	008	3 V

RCA Processes (ZPRODUCT) Class

File:	PRIMARY	Description:	ZPRODUCT further refines resolution by product based on PREFACE.ZDOMNAME (software, OS, PATCH, etc...).
Domain:	SYSTEM		
Class:	ZPRODUCT		
Icon:		Can connect to:	

ZPRODUCT Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ALWAYS	Method		80 M
ALWAYS	Connect To		80 C
ALWAYS	Connect To	PRIMARY.POLICY.USER/& (ZMASTER.ZUSER)	80 C
DESCRIPT	Process Description	Processing Client Request for &ZCUROBJ	32 V
ZMAXOKRC	Max acceptable method Return Code	008	3 V

SQLTABLE (SQLTABLE) Class

File:	PRIMARY	Description:	An instance of the SQLTABLE Class defines the parameters necessary to populate a table in an SQL Database.
Domain:	SYSTEM		
Class:	SQLTABLE		
Icon:		Can connect to:	

SQLTABLE Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ALWAYS	RDM Method Any method you specify for this attribute is unconditionally executed by Client Automation when this instance is	SYSTEM.ZMETHOD .PUT_SQL_OBJECT	50 C

	resolved.		
DESTOBJ	Destination Object		32 V
PUTTYPE	Insert/Replace [I/R] Indicator for type of operation to be performed on the back-end database.		1 V
SQLDSN	DSN Name The ODBC data source name (DSN) to be used to connect to the SQL database.		50 V
SQLPASSW	Password The password to use in the database connect process.		50 V
SQLTABLE	Table Name The file name of the SQL table in the SQL database into which EDMMSQLP will store the data.		50 V
SQLTOUT	Time Out (in seconds) Timeout value for the SQL Connect operation.	30	3 V
SQLUSER	User Name		50 V
SRCOBJ	RDM object containing information The name of the Client Automation source object containing the data to be recorded. When recording Client Automation events, this must be APPEVENT.	&(ZCURPCLS)	32 V
VC000 to VC009	Columns 1 to 10 Defines the correspondence between a variable in the Client Automation source object and the column in the back-end database table where it will be stored. One VCnnn value must be specified for each variable-column pair participating in the operation. Each VCnnn variable is coded as follows: VARNAME[, COLUMN_NAME [,U]] VARNAME is the name of the variable in the Sentient object whose value will be transferred to the back-end database. COLUMN_NAME is the name of the		50 V

	column in the back-end database table that will receive the data from the VARNAME variable. If COLUMN_NAME is omitted, VARNAME will be used; this assumes that the back-end database table's receiving column is named the same as the Sentient object variable supplying its data.		

System Files (ZFILE) Class

File:	PRIMARY	Description:	Reserved for future use.
Domain:	SYSTEM		
Class:	ZFILE		
Icon:		Can connect to:	

ZFILE Class Attribute Reference

Attribute Name	Explanation	Value of Base Instance	Length Type
ZMETHOD	Method		50 M
ZUSERID	User ID		8 V

Chapter 11

Application Management Profiles

Environment Variables

In addition to the standard platform-specific environment variables, such as `windir=C:\WINDOWS` for Windows operating systems, there are AMP specific environment variables which are listed below. The APPMGMT environment variables, like other environment variables, are usually used in environment references for pre- and post- conditions. You may also want to set a local variable to the value of an environment variable.

List of Environment Variables

- APPMGMT_BYTEORDER Byte Order
- APPMGMT_MACHINE Processor
- APPMGMT_OS Operating System
- APPMGMT_OSCOMMONNAME Common Name of OS
- APPMGMT_OSSERVICEPACK Windows Service Pack
- APPMGMT_OSVERSION Operating System Version
- APPMGMT_PLATFORM Platform
- APPMGMT_HOSTIP Host IP address
- APPMGMT_HOSTNAME Fully qualified name of the machine
- APPMGMT_PID Process Id of Application Manager agent
- APPMGMT_STANDALONE [Note: A Value of **Y** only applies to pre-Version 7.20 releases; as of Version 7.20, support for AMPs is embedded in the Application Manager Agent.] If the AMP Agent is running under the Application Manager agent the value is **N**, otherwise it is running standalone and the value will be **Y**.
- APPMGMT_PHASE_NAME Phase name that the Application Manager is currently executing.

The following are examples of Windows XP values:

- APPMGMT_BYTEORDER = littleEndian
- APPMGMT_MACHINE = intel
- APPMGMT_OS = Windows NT
- APPMGMT_OSCOMMONNAME = Windows XP
- APPMGMT_OSSERVICEPACK = Service Pack 1
- APPMGMT_OSVERSION = 5.1

- APPMGMT_PLATFORM = windows
- APPMGMT_HOSTIP = 192.168.0.11
- APPMGMT_HOSTNAME = LBLACK.americas.foods.com
- APPMGMT_PID = 1800
- APPMGMT_STANDALONE = Y
- APPMGMT_PHASE_NAME = install

The following are examples of Red Hat Linux values:

- APPMGMT_BYTEORDER = littleEndian
- APPMGMT_MACHINE = i686
- APPMGMT_OS = Linux
- APPMGMT_OSCOMMONNAME = Linux 2.2.14-5.0smp
- APPMGMT_OSSSERVICEPACK = N/A
- APPMGMT_OSVERSION = 2.2.14-5.0smp
- APPMGMT_PLATFORM = unix
- APPMGMT_HOSTIP = 192.168.0.11
- APPMGMT_HOSTNAME = LBLACK.americas.foods.com
- APPMGMT_PID = 1900
- APPMGMT_STANDALONE = Y
- APPMGMT_PHASE_NAME = install

The following are examples of HP-UX values:

- APPMGMT_BYTEORDER = bigEndian
- APPMGMT_MACHINE = 9000/800
- APPMGMT_OS = HP-UX
- APPMGMT_OSCOMMONNAME = HP-UX B.11.00
- APPMGMT_OSSSERVICEPACK = N/A
- APPMGMT_OSVERSION = B.11.00
- APPMGMT_PLATFORM = unix
- APPMGMT_HOSTIP = 192.168.0.11
- APPMGMT_HOSTNAME = LBLACK.americas.foods.com
- APPMGMT_PID = 1900
- APPMGMT_STANDALONE = Y
- APPMGMT_PHASE_NAME = install

The following are examples of Solaris values

- APPMGMT_BYTEORDER = bigEndian
- APPMGMT_MACHINE = sun4u
- APPMGMT_OS = SunOS
- APPMGMT_OSCOMMONNAME = SunOS 5.9
- APPMGMT_OSSERVICEPACK = N/A
- APPMGMT_OSVERSION = 5.9
- APPMGMT_PLATFORM = unix
- APPMGMT_HOSTIP = 192.168.0.11
- APPMGMT_HOSTNAME = LBLACK.americas.foods.com
- APPMGMT_PID = 1900
- APPMGMT_STANDALONE = Y
- APPMGMT_PHASE_NAME = install

Application Management Profiles Standalone Agent

A command line utility, standalone agent, is available on the installation media to test a Master Application Profile (MAP) or Custom Application Profile (CAP). Use the standalone agent to execute profiles without Sentient Client Automation infrastructure or to test the profile. The standalone execution runs as the logged on user.

The standalone agent, includes the following arguments:

- The `-xml` parameter is used to specify the profile you want to test.
- The `-mode` parameter is used to specify the phase you want to run in the test. The possible values for mode are **create** (application installation), **delete** (application removal), **verify** (verification of application components), **repair** (repairing broken applications), and **update** (for updating an application). If mode is omitted, it will default to **create**.
- The `-idmlib` parameter specifies the location to build Sentient objects. If this parameter is not specified, the objects are built in the location of the `appcntrl.tkd`. If `idmlib` is omitted it will default to either `IDMROOT`, if Sentient agent or the Administrator is installed, or the current directory, if the agent or Administrator is not installed. Of course, this also depends on what directory option is chosen in the command itself.
- The `-logfile` parameter allows you to set an alternate log file. The default is `appcntrl.log`.

Caution: Be aware that this will not work for all AMPs. A profile that is designed to run locally may succeed. However, if the profile has dependencies between phases, then you will need to run all of those phases to get a reliable result. In other words, this is not guaranteed to work for all profiles. Always run the profile in a test environment before moving to production.

To test a profile:

1. Load any media that the profile requires to the appropriate directory.
2. Copy the profile to the same directory as the Application Manager agent on the test device.
3. Open a command prompt, and navigate to the directory where the Application Manager agent and the profile have been loaded.
4. Type the following command:

```
nvdkit appcntrl.tkd -xml filename.xml -mode modetotest
```

The profile runs, and executes any commands that the profile indicates. All actions are logged to `appcntrl.log` in the same directory as `appcntrl.tkd`. Messages are appended to the end of the log with the most recent entries at the end.

For testing the verify phase, you should following these steps:

- Break the application.
- Run this utility with `-mode verify`.
- Confirm that the application was found to be broken by checking `appcntrl.log`. Also check that the log is reporting the correct error.

As an option, run the utility with `-mode repair`.

To ensure the old timestamped XML files are not purged during Application Management Profiles connect, add the following command line option for ZCREATE and ZINIT service under CSDB:

```
-purgeoldxml N
```

For example: `hide nvdkit &(ZMASTER.ZSYSDRV)&(ZMASTER.ZSYSDIR) appcntrl.tkd
-instance &(ZCURRENT.ZCURINS) -purgeoldxml N`

By default, the value of `purgeoldxml` is `Y`.

Chapter 12

Policy Management

The Policy Server Service is used by the Sentient Console for administration purposes such as mapping services to users in the directory tree. It provides integration and extended enterprise functionality with your directory services. Policy method connections in the Sentient Configuration Server Database (CSDB) are used to determine what services should be distributed and managed for the user that is currently logged on by querying the Policy Server.

The Policy Server leverages your investment in directory services while using Sentient for software management. This greatly reduces the total cost of ownership of your environment. In other words, directory services handle policy management and Sentient manages services. This saves you time because you do not have to define or maintain lists of users in the Sentient Configuration Server Database.

The Policy Server integrates with existing Lightweight Directory Access Protocol (LDAP) directory servers and SQL databases in a customer's enterprise to enable single source points of control for user authentication, access policies, and subscriber entitlement. When using an external policy store, the default behavior in the Core CSDB is:

- For Sentient agent connects, if the user is not defined by a USER Instance, resolution defaults to using the machine domain name and looks for policy defined in an external LDAP directory that has been configured for access using the policy settings on the Core and Satellite Consoles.
- The resolution by machine name from an external directory is defined in the `_NULL_INSTANCE_` of `PRIMARY.POLICY.USER`. This instance includes an `_ALWAYS_` (Utility Method) connection with its attribute set to `SYSTEM.ZMETHOD.LDAP_RESOLVE`.

Policy Server Processing

The Policy Server acts as a bridge between the Configuration Server and a directory server. It is a separate component from the Configuration Server. The Policy Server is installed as part of your Core and Satellite installation. A Policy Server can communicate with several LDAP directory services. The Policy Server allows you to perform more customized tasks that cannot be performed directly through the Sentient Console.

The following provides an overview of Policy Server Processing:

1. The Sentient agent connects to the Configuration Server to resolve its **desired state**. The desired state embodies the content that Sentient manages for a device. The desired state for each device is dynamically created by the Configuration Server based on information in the Configuration Server Database.
2. The Configuration Server contacts the Policy Server to perform policy resolution, and builds the agent's desired state using the policy information. The Sentient administrator *must* entitle an agent device to minimum one service. If the Policy Server does not find any entitlements defined for an agent device, it checks the value for the parameter `POLICY_DEFAULT` in the `MGR_POLICY` section of the `edmprof.dat` file. Set the `POLICY_DEFAULT` parameter to a

valid ZSERVICE, such that if entitlements are not defined for an agent device, the ZSERVICE is resolved by default.

3. The policy method, LDAP_RESOLVE, handles the resolution requests, converting the requests into HTTP queries to the Policy Server, and treats the results as a set of objects and attributes to be incorporated into the desired state of the connected agent.
4. The Configuration Server completes resolution of the desired state and returns the information to the Sentient agent.

The Policy Server can maintain a persistent connection to multiple LDAP directory servers and responds to policy requests by performing a policy resolution against the policy database and returns the set of objects resolved as the result set of the HTTP query. Internally, the LDAP Policy Extension starts at the specified DN, and walks the entire tree of relationships that the object has with other objects, accumulating policy attributes. Then it evaluates all conditional policies, and finally resolves any conflicting policies, using a straightforward should/may, grant/deny model.

Adding Client Automation Policy Attributes

The Policy Server requires that the LDAP schema of an existing directory implementation be modified before it can be used to manage policy if the directory is not Active Directory (AD).

These attributes are used to manage policy scope, relationships, and assignments. Consult your directory service documentation and your enterprise's directory service administrator to make these changes. Be sure to back up your directory schema before any modifications.

Note: Changes to the LDAP schema can be risky because modifications to many directory services are not reversible. Be sure you type correctly. Check and double check the values you are entering before saving the changes to each value entered into the directory schema. Consult your directory services administrator and documentation.

The LDAP Policy Extension understands the standard relationships that exist in a directory between different objects (parent-child, memberOf). In addition to these standard relationships, three additional attributes may be used:

- `<prefix>Flags`
controls various subtle aspects of the policy resolution.
- `<prefix>Link`
allows you to specify additional, potentially dynamic or conditional relationships. Two directory objects are said to be **related** if one can be reached from the other, directly or indirectly. Examples of relationship include parent-child, and group membership (a user is **related** to the group he is a member of). Relationships are unidirectional.
- `<prefix>Policy`
allows you to define resultant strings that will be netted out during policy resolution. Policy is a string that is used to **represent a desired** outcome. The Policy Server does not impose any particular interpretation upon policy string. When used in conjunction with the LDAP Adapter, the adapter will interpret the policy string as the name of an application defined within Sentient.

By default the prefix used is `edm`, but alternatives may be used to allow your directory to support multiple concurrent policy frameworks for different purposes. For example:

- Add `edmFlags` as a single-valued, integer attribute with an object ID of 1.3.6.1.4.1.2133.2.1.1. The `edmFlags` controls scope of a policy and various subtle aspects of the policy resolution.. This is added as an optional attribute of the `nvdObject` class.
- Add `edmLink` as a multi-valued, case-sensitive string with an object ID of 1.3.6.1.4.1.2133.2.1.2. This attribute allows you to specify additional, potentially dynamic or conditional relationships. This attribute also allows you to create a connection to a group that is not part of the user's LDAP group membership. This is added as an optional attribute of the `nvdObject` class.
- Add `edmPolicy` as a multi-valued, case-sensitive string with an object ID of 1.3.6.1.4.1.2133.2.1.3. Use `edmPolicy` to assign services to users and groups. The `edmPolicy` allows you to define resultant strings that will be netted out during policy resolution. This is added as an optional attribute of the `nvdObject` class.

The following attributes are not mandatory, but you may want to add them.

- Add `edmPolicyDefault` as a multi-valued, case-exact string with an object ID of 1.3.6.1.4.1.2133.2.1.4. Use `edmPolicyDefault` to assign policy defaults. This is added as an optional attribute of the `nvdObject` class.
- Add `edmPolicyOverride` as a multi-valued, case-exact string with an object ID of 1.3.6.1.4.1.2133.2.1.5. Use `edmPolicyOverride` to define policy overrides. This is added as an optional attribute of the `nvdObject` class.

Adding the `nvdObject` Class

Some directory services, such as Microsoft Active Directory, do not allow adding of attributes to the *top* class. This is the highest level in the schema. If you cannot add attributes to the *top* class, create a class that will hold the required `edmLink`, `edmFlags`, and `edmPolicy` attributes, and inherit the values included in the *top* class. `EdmPolicyOverride` and `EdmPolicyDefault` are not required, but may be added for additional functionality. By creating this class, including its inherited values, the areas needed to apply Sentient policies to specific areas of the directory tree can be modified. If you can add the attributes to the *top* class, policies can be placed anywhere in the tree.

If you need to create a class, name the class `nvdObject`. Create it as an auxiliary class with *top* as its parent class. Set the object ID to 1.3.6.1.4.1.2133.2.1. After creating the `nvdObject` class, you must add the `edmFlags`, `edmLink`, and `edmPolicy`. To proceed, you must reload your directory schema. Consult your directory service's documentation for instructions on how to do this.

Modifying classes with `nvdObject`

Once the schema has been re-loaded, the values entered above will show up as a selection, and you can add the `nvdObject` class to areas of your directory affected by the Policy Server.

To complete the modification for Microsoft Active Directory, `nvdObject` must be added as an Auxiliary class on the **Relationships** tab to all of the Active Directory classes listed below.

- Person
- Container
- DomainDNS

- Organizational Unit
- Group

You have now completed the necessary modifications to your directory schema. See ["Connection to LDAP" below](#) for instructions on how to connect the Policy Server to your directory services.

Connection to LDAP

The Policy Server obtains information about the LDAP directory services configured in Sentient by communicating with the Portal periodically. The Policy Server communicates with the Portal by using information found in the `pm.cfg` file as discussed in ["The Policy Server Configuration File" below](#) section.

Based on the information obtained from the portal, the Policy Server then generates a configuration file (`.cfg` file) for each LDAP directory service that is configured. These files contain the necessary options for the Policy Server to be able to communicate with the directory services and resolve policies as discussed in ["LDAP Configuration Files" on the next page](#) section.

The Policy Server Configuration File

The Policy Server configuration file, `pm.cfg`, is located in the `<InstallDir>\PolicyServer\etc` folder. It contains necessary information for Policy Server to Portal communication. It also contains some global configuration settings as discussed in ["Support for Multiple LDAP Connections" below](#).

You may want to edit this file (with a text editor such as Notepad) to enable the directory services (`ENABLE_DS`) option and to configure the directory services synchronization interval (`DSSYNC_INTERVAL`). When the directory services option is enabled, the Policy Server will communicate with the Portal every `DSSYNC_INTERVAL` number of seconds to obtain a list of directory services. The default synchronization interval is 300 seconds. You can adjust this setting depending on how dynamic your specific environment is.

It is highly recommended that you do not change the `RMP_*` default settings the Policy Server configuration file without consulting with Persistent Systems support.

Note: If you make manual changes to `pm.cfg`, you will need to restart the Sentient Policy Server service that is hosting the Policy Server.

Support for Multiple LDAP Connections

The Policy Server supports multiple concurrent LDAP queries to a single LDAP directory at one time. You can configure the number of concurrent LDAP queries in the Policy Server's configuration file, `pm.cfg`. The table below describes which parameters apply. Note as stated previously, when you make changes to `pm.cfg`, you will need to restart the Sentient Policy Server service.

Configurable Values for Multiple LDAP Queries

Value	Default	Description
N_workers	4	Specifies number of parallel LDAP directory connections to be created.
PolicyUrl	/policy/ldap	Registers the URL of the Policy Server's LDAP. This is required to use the N_WORKERS parameter. The parameter name is case sensitive. If you do not have this line in your <code>pm.cfg</code> , then you will need to add it.

LDAP Configuration Files

As previously noted, Policy Server generates an LDAP configuration file for each LDAP directory service configured in Sentient. These files are located in the

`<InstallDir>\PolicyServer\etc\ldap` folder.

Each configuration file contains two distinct sections, the `DsConfig` section and the `DsConfigOverride` section. The override section contains settings that can be overridden, such as adding a list of hostnames for failover, policy prefix changes (edm instead of nvd), and other settings that may require an override. When a change is detected in the Portal relating to the directory service, the `.cfg` file will automatically be updated. However, the overrides section will remain intact.

Options for DsConfig Section

The following table shows the options that are specified in the `DsConfig` section of the LDAP configuration file:

DsConfig Options

Option	Description
BASE_DN	Specifies the base domain. Example: dc=asdfs, dc=com.
BIND_DN	Specifies the fully qualified name of the account that has Active Directory Schema Permissions on the Directory. Example: cn=Administrator , cn=Users, dc=asdfs, dc=com or administrator@asdfs.com
BIND_PW	Specifies password associated with BASE_DN
TYPE	Specifies type of LDAP connection. It is either 'ldap' or 'ldaps'. 'ldap' is used for non SSL connection with LDAP server and 'ldaps' is used for SSL connection.
HOST	Specifies the host name of the LDAP server
PORT	Specifies the LDAP server port number

Option	Description
SSL_ CACERTDIR	Specifies the directory that hosts Certificate Authority certificate file
SSL_ CACERTFILE	Specifies full path for Certificate Authority certificate file
LDAP_DEBUG	Specifies if LDAP debugging should be enabled. Values are either 1 or 0.
LDAP_LOGFILE	Specifies the log file in which debugging messages are logged if LDAP_DEBUG is enabled
DISABLED	Specifies if this LDAP configuration can be used for Policy Resolution. Values are either 0 or 1 based on if this configuration file can be used or cannot be used for Policy Resolution
FORWARD	Applicable to the Satellite servers in your environment. Specifies if the agent connect resolution request can be forwarded to the upstream server. Values are either 0 or 1. The default value is zero. Zero signifies that the policy server on the Satellite server processes the AD policy resolution request. The OpenLDAP or internal policy resolution requests are forwarded to the upstream server.
IGNORE_ REFERRALS_ IN_RSOP	When Policy Server flags resolved entitlement catalog as partial while policy resolution, managed device do not process partial catalogs except when soft policy is requested, which is a rare scenario in Sentient implementations. When IGNORE_REFERRALS_IN_RSOP is set to 1, Policy Server will flag resolved entitlement catalog as complete.

Options for DsConfigOverride Section

The following table shows the options that are specified in the `DsConfigOverride` section of the LDAP configuration file:

DsConfigOverride Options

Field	Default	Description
HOST	Not Set	Specifies coma separated host names for failover support; e.g. host1,host2,host3
LDAP_ DEBUG	0	Specifies if LDAP debugging should be enabled. Values are either 1 or 0.
LDAP_ LOGFILE	Not Set	Specifies the log file in which debugging messages are logged if LDAP_DEBUG is enabled
DISABLED	Not Set	Specifies if this LDAP configuration can be used for Policy Resolution. Values are either 0 or 1 based on if this configuration file can or cannot be used for Policy Resolution.

Field	Default	Description
VERSION	3	Specifies LDAP Protocol version to use (2 or 3)
CACHE	1	Specifies if caching is enabled. Values are either 0 or 1.
FLUSH_FREQ	"60*60"	Specifies the delay in seconds between each flush of the cache
RETRY	1	Specifies the number of attempts to issue the LDAP request before marking the directory as unavailable. If this occurs, a reconnection attempt will be made when the next ping is performed. This is also the number of attempts to reconnect to the directory if the ping detects the directory to be offline. If HOST is specified as a comma separated list of host names, RETRY should have a value of at least the number of host names.
TIMEOUT	120	Specifies Timeout (in seconds) for LDAP request
PREFIX	"edm"	Specifies prefix of Sentient attributes and object class specified in LDAP schema
VIEW	Not Set	See the "Controlling Policy Scope Globally" on page 1304 section.

Policy Section in the Configuration Server Profile

To understand how the Configuration Server communicates with the Policy Server, review the information contained the Configuration Server profile file, `edmprof.dat` located in the `<InstallDir>\ConfigurationServer\bin` folder. It contains a `MGR_POLICY` section that specifies the host name and port number of the Policy Server as shown in the following example:

```
* Manager Policy Section *
* HTTP_HOST = Host name of Policy Server *
* Multiple hosts may be specified (space or comma *
* separated) for fail over *
* HTTP_PORT = IP Port number of Policy Server *
* NO restart required *
*-----*
[MGR_POLICY]
HTTP_HOST = XXX.XXX.XXX.XXX
HTTP_PORT = 3466
```

You should not modify this file without contacting [Persistent Systems Support](#).

Configuring the LDAP Method

If you are using LDAP, you will have to prepare your Configuration Server Database (CSDB) to use Policy Server.

You must:

- Create a connection to the LDAP method in the CSDB as discussed in this section
- Connect the users to the LDAP method as discussed in the next section, ["Connecting to the LDAP Method" on the next page](#)

Note: Your Sentient Core and Satellite installation includes templates for LDAP ZMETHOD. Modify these templates as necessary using the information in this section.

To create the LDAP method in the Configuration Server Database (CSDB):

1. Open the Sentient CSDB Editor, and go to PRIMARY.SYSTEM.ZMETHOD.
2. Right-click **Methods (ZMETHOD)**.
A shortcut menu opens.
3. From the shortcut menu, select **New Instance**.
The Create Instance dialog box opens.
4. Type `LDAP_RESOLVE` in the text box, and click **OK**.
The Sentient CSDB Editor window opens.
5. Double-click **LDAP_**.
The tree expands.
6. Double-click **LDAP_RESOLVE** in the tree view.
The attributes of LDAP_RESOLVE appear in the list view.
7. Double-click the **ZMTHNAME** attribute in the list view.
The Editing Instance dialog box opens.
8. In the **Member Name of Method** field, type `RADISH`.

Note: Configuration Servers running on UNIX® platforms are case sensitive and require RADISH entered in upper case.

9. Click **ZMTHTYPE**.
10. In the **Method Type** drop-down list, select **REXX**.
11. Click **ZMTHPRMS**.
12. In the **Parameters Passed to Method** text box, use the following values.
For HTTP:

`http:///policy/ldap?dn=<<ZDN>>&&os=<<ZOS>>`

For Microsoft Active Directory:

- To manage policies by machine (preferred method for single directory service):
 - `ldap:///dc=domainname,dc=forestname,dc=com??sub?samaccountname=<<ZUSERID>>$` (If the client uses \$MACHINE as the ZUSERID)
 - `ldap:///<ADINFO.COMPDN>`
 - `http:///policy/ldap?dn=<<COMPDN>>`

For example,

```
ldap:///dc=asdfs,dc=com??sub?samaccountname=<<ZUSERID>>$
```

- To manage policies by machine (preferred method for multiple directory services):
`ldap:///<ADINFO.COMPDN>`
- To manage policies by user:
`ldap:///dc=`
`domainname,dc=forestname,dc=com??sub?samaccountname=<<LOCALUID>>`

For example,

```
ldap:///dc=asdfs,dc=com??sub?samaccountname=<<LOCALUID>>
```

For Novell Directory Services (NDS):

- To search the entire NDS tree for policy, type:
`ldap:///o=organization??sub?cn=<<ZNTUSER>>`

For example,

```
ldap:///o=cert??sub?cn=<<ZNTUSER>>
```

- To search NDS with a specified Distinguished Name, type:
`http:///policy/ldap?dn=<<ZMASTER.DN>>`

For Netscape iPlanet:

- To manage policies by user type:
`ldap:///dc=com??sub?uid=<<ZUSERID>>`

13. Click **OK**.

The Instance Edit Confirmation dialog box opens.

14. Click **Yes** to confirm the changes. The Sentient CSDB Editor window opens.

Now, whenever a managed device connects to the Configuration Server, the null instance calls the policy method, and will point to the appropriate services for that user.

Connecting to the LDAP Method

You must connect the LDAP method to an instance in the POLICY domain for policy resolution.

To connect the user to the LDAP method:

1. Open the Sentient CSDB Editor.
2. Navigate to PRIMARY.POLICY.USER.
3. Double-click the null instance.

Note: If the null instance is connected to the Default workgroup, change the name of the instance from Default to `_NONE_`.

4. In the list view, double-click on the **`_ALWAYS_ Utility Method`** line.
The Editing Instance dialog box opens.
5. In the **Utility Method** text box, type `SYSTEM.ZMETHOD.LDAP_RESOLVE`.

6. Click **OK**.
The Instance Edit Confirmation opens.
7. Click **Yes**.

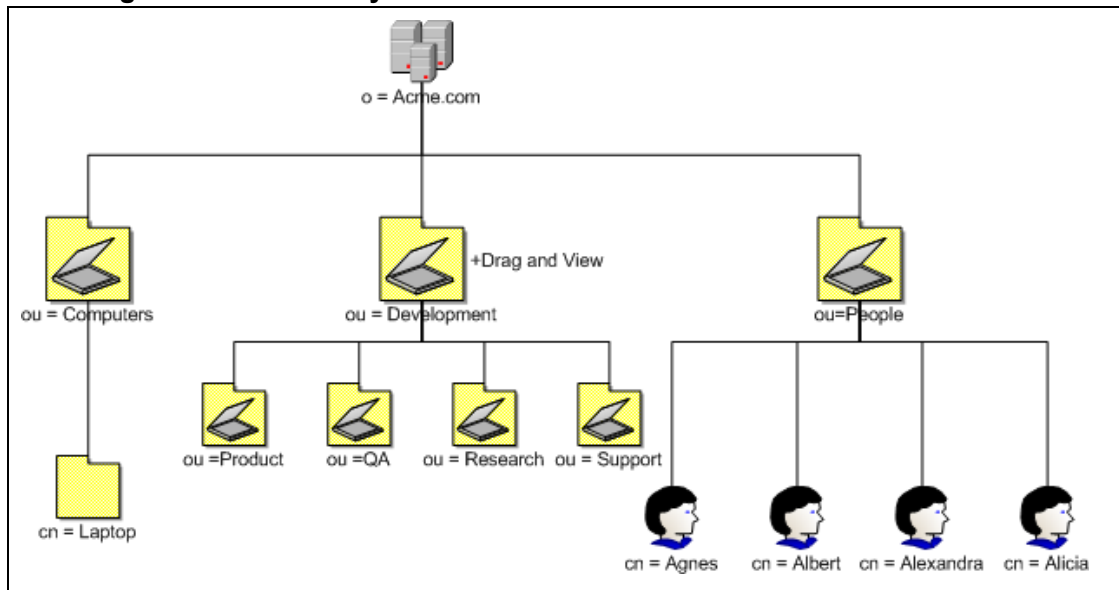
The LDAP_RESOLVE method is connected to the Null User instance.

Policy Scope and Resolution

By default, a subscriber inherits the policy from the parent of any groups it is linked to. This link can be through either the subscriber's directory service membership or through the use of the `edmlink` attribute. The figure [Acme organization directory structure](#) shows a part of the Acme organization. It has three organizational units, Computers, Development, and People.

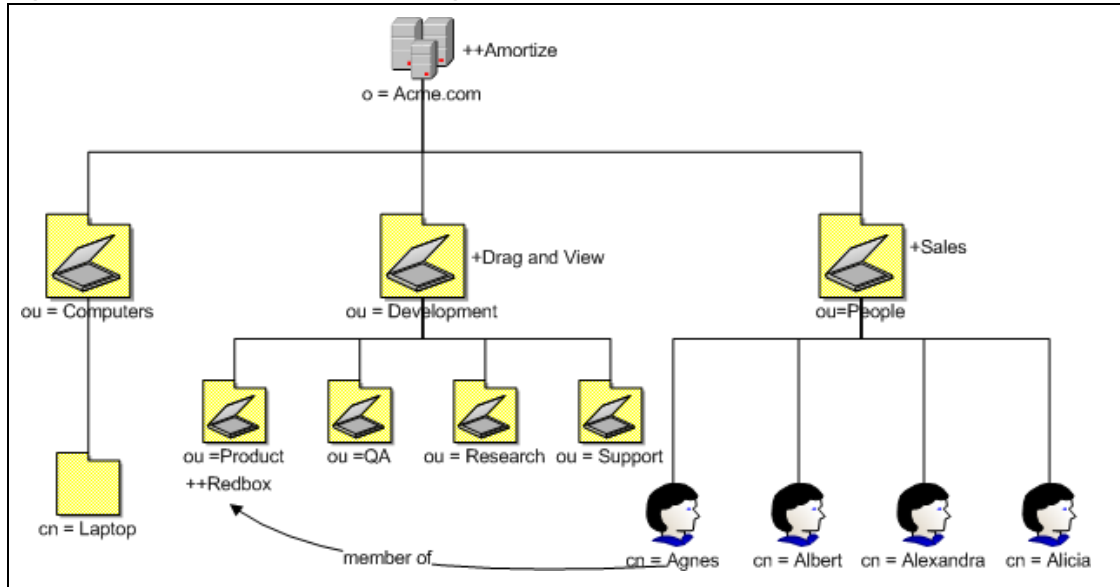
Computers holds the Laptop container. **Development** includes the Product, QA, Research, and Support organizational units. **People** includes the actual users of the enterprise.

Acme organization directory structure



In this figure, Agnes will inherit the policy of the People organizational unit and the Acme organization.

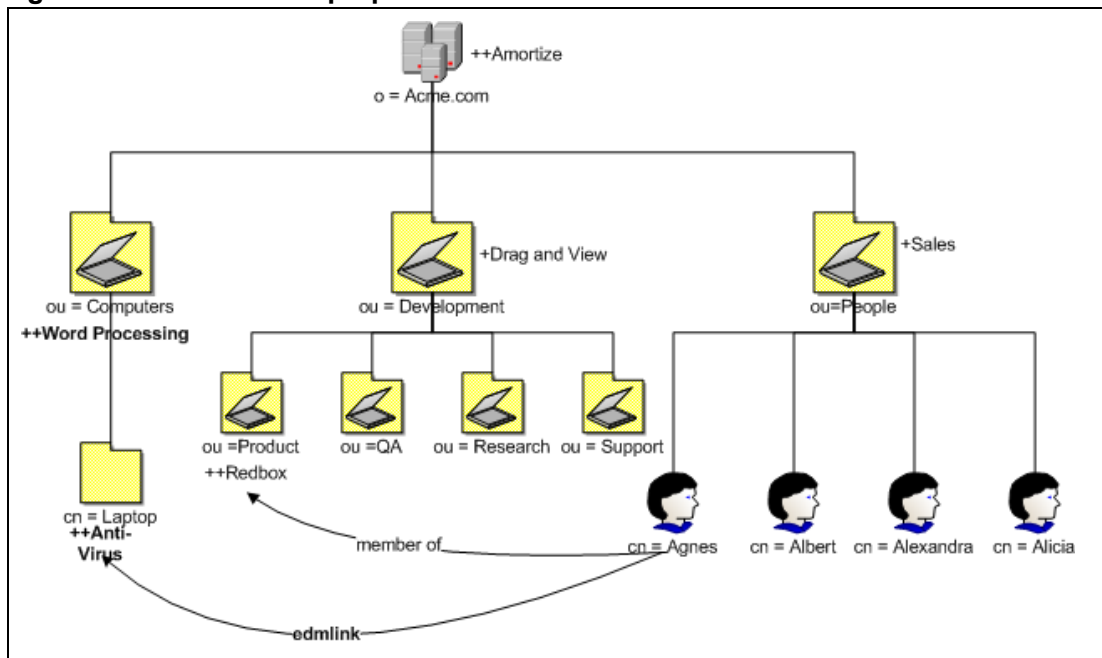
Agnes is a member of Product organizational unit



If Agnes is a member of the Product organizational unit, she will also inherit the policy from that unit and the Development organizational unit. In the figure [Agnes is a member of Product organizational unit](#), Agnes would get Sales and Amortize because she is a part of the People organizational unit. Because Agnes is a member of the Product organizational unit, she would *also* inherit Redbox *and* Drag and View.

Suppose that you need Agnes to receive the services associated with the laptop container, but she is not linked to that container through directory services. Use edmlink to connect her to that container.

Agnes is linked to the laptop container



In the figure [Agnes is linked to the laptop container](#), Agnes will receive Anti-Virus because she has been linked to the laptop container. Since laptop is part of the Computers organizational unit, she will also get Word Processing. Now, she has a total of six applications.

Managing Policy Scope

If you do *not* want to inherit the policy from the parent objects, you can limit the Policy Server's scope of resolution. You can do this either globally for the entire directory structure or for only specific objects. Manage the scope globally by modifying the Policy Server configuration file. Control policy scope for one object by using the `edmFlags` attribute.

Caution: Be sure that you have a thorough understanding of your directory structure. When designing a change to the scope of policy resolution, anticipate the result of your modifications *before* making the modifications.

Controlling Policy Scope Globally

The VIEW option allows you to control whether or not to continue up the directory tree to assign policy. Modify the VIEW option in the `DsConfigOverride` section of the individual LDAP configuration file to control the scope. See "[LDAP Configuration Files](#)" on page 1297.

The syntax for the VIEW option is:

```
VIEW {  
<attr> {view}  
}
```

Where *attr* is one of the attributes listed in the LINKS (`edmLink`, `groupmembership`, `memberof`, `aliasedobjectname`) configuration option in `pm.cfg`, and *view* is a list of LINKS the Policy Server is allowed to see. An empty list means that there is no view when visiting an object from the specified attribute. This would result in following that link and not continuing. You can list as many or as few attributes as needed.

The default values for the LINKS configuration option are `edmLink`, `memberof`, `groupmembership`, and `aliasedobjectname`. When you look at a particular object such as a group or user through the Policy Server interface, you will see only these attributes for that object. If you do not want Policy Server to inherit the policy for any parents of an `edmLink` attribute, modify the VIEW option in the `DsConfigOverride` section of the individual LDAP configuration file like this:

```
VIEW {  
edmLink { }  
}
```

This configuration with the empty brackets tells Policy Server to follow `edmLink`, but not to inherit from any parents or any links contained within the object from that branch of the directory tree.

Looking back at the Acme organization example, suppose you want Agnes to receive policy for the laptop container, but not inherit any policy from the Computers organizational unit. In the figure [Agnes is linked to the laptop container](#) Agnes will receive Anti-Virus because she has been linked to the laptop container, but she will not inherit Word Processing when `edmLink` is configured with empty brackets to not inherit from any parents.

Similarly, to follow a `memberof` attribute, and then not inherit from the parent objects, `edmLink` with `memberof` would be replaced. The `VIEW` option would look like this:

```
VIEW {  
  memberof { }  
}
```

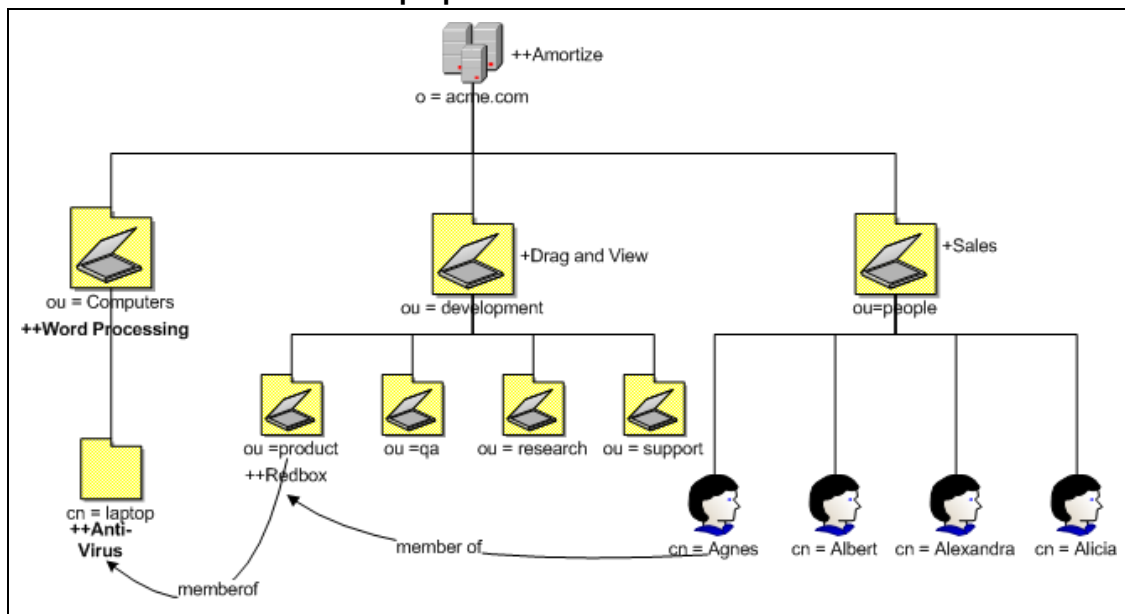
This configuration with the empty brackets tells Policy Server to follow `memberof`, but not to inherit from any parents from that branch of the directory tree or any links contained within the object.

Finally, to follow `memberof` relationships, the `VIEW` option would look like this:

```
VIEW {  
  memberof {memberof}  
}
```

This configuration with the `memberof` in quotes tells Policy Server to follow `memberof`, but not to inherit from any parents from that branch of the directory tree. When a `memberof` relationship is to be followed, `memberof` relationships is followed until an object that does not contain a `memberof` relationship is reached. In the figure [Product is a member of the laptop container](#), Agnes will get Sales, Amortize. Then she will get Redbox because she is a member of Product. Since Product is a `memberof` laptop, she will get Anti-Virus. If Laptop had any `memberof` relationships, she would follow those relationships, too. Agnes will not follow any relationships other than `memberof`.

Product is a member of the laptop container



Optimization for Single-Service Policy Resolution

To optimize the resolution of the policies for a single service, you can include domain names in the CSDB, which will be used in the case of a single service policy resolution. These domains are skipped for a detailed policy server resolution.

Caution: Domains thus specified will lose the ability to pass policy attributes into the resolution model.

The default behavior in policy resolution is to walk the tree for all calls.

To create domain names in CSDB for single-service policy resolution:

- Create instances of the POLICY.POLPRMS class with the name of the domain.
The actual domain name skipped for single service resolution is the value of the variable XDOMAIN in the instance. For example, if you create an instance of the POLICY.POLPRMS class with the name, PATCHMGR, the XDOMAIN variable in this instance will automatically get the name PATCHMGR. You may specify any number of instances as required. The XDOMAIN variable can also contain the star (*) character as a wildcard to skip all the domains with matching names.

The rules for the wildcard in the XDOMAIN variable are the following:

- Only one star maximum can be present in the XDOMAIN.
- It can be placed anywhere in the XDOMAIN.

LDAP

This section provides more information on directory services for policy administrators needing additional information. It also includes descriptions of use of substitution and expressions and URLs used for Sentient Policy Server.

Sentient Policy Server and LDAP

An LDAP directory is a hierarchically named tree of objects, where each object has a class (type) or classes, and contains potentially many named attributes, appropriate to its classes. Each attribute may contain multiple values.

It is outside the scope of this document to describe in any detail what an LDAP directory means. As a rapidly growing force in the systems management industry, many excellent sources exist for further background.

LDAP directory must support either the LDAP v2 or LDAP v3 protocols to use with Sentient.

Substitution

Two forms of substitution are provided:

- Current Object Attributes: <<nameOfAttr>, or
- Inbound Object Attributes: <<in.nameOfAttr>>

The former allows you to construct expressions based upon the value of another attribute in the current object (same one that contains the edmLink or edmPolicy), for example,

```
edmLink: cn=<<homePC>>, cn=Computes, o=Acme.  
edmLink: cn=wnt001, cn=Computers, o=Acme ; <<homePC>> == "wnt001".
```

The latter allows you to reference attributes that were supplied as input to the policy resolution, for example:

```
edmPolicy: ++SOFTWARE/STRATUS_PAD; <<in.hostname>> == "XKEZ01$"
```

Currently the minimum attributes that will exist are listed in the following table:

Default Inbound Object Attributes <<in.nameOfAttr>>

Attribute Name	Sample Value
context	{}
dn	{cn=su61er, cn=computers, dc=acme, dc=com}
dname	software
domain	{ACMEWEST\XKEZ01\$}
hostname	{XKEZ01\$}
ipaddress	192.168.0.100
mtime	{2007-06-22 18:54:35}
nvdipnetworknumber	192.168.0.0
nvdsubnet	255.255.255.0
smenclosureserialnumber	CNU0123456
smsystemproductname	{HP Compaq nc6000 (DU655C)}
smsystemmanufacturer	Hewlett-Packard
smsystemserialnumber	CNU0123456
smsystemuuid	27494E2D171E11DB09906D9908020929

Expressions

The expressions are implemented as Tcl (www.scriptics.com) expressions, where instead of using \$myVar you would use <<myAttribute>>. A simplified summary of valid expressions is provided below. Most of the standard C language expression operators are valid.

Expressions

Expression	Meaning
A && B	Logical AND
A B	Logical OR
!A	Logical NOT
<<myAttr>> == "Hello"	Test for equality (case-sensitive)

Expression	Meaning
<<myAttr>> != "Hello"	Test for inequality
<<myAttr>> < 55	Numerical comparison for less than
<<myAttr>> >= "Hello"	Dictionary comparison for greater than or equal to (C locale)

There are also a small number of specialized functions.

Specialized Function Example

Example	Meaning
[memberOf "ou=Accounting, o=Acme"]	Yields TRUE if the DN specified is part of your policy model.
[parent <<dn>>] == <<aSpecialDN>>	Yields TRUE if the parent DN of the current object is the same as the "aSpecialDN".

The LDAP Extension URL Namespace

The LDAP extension provides the following special purpose URLs:

LDAP Extension URL Namespace

URL	Description
/policy/ldap?<x-url encoded query>	Perform machine-readable policy resolution. The query arguments should be an attribute value list of inbound attributes, formatted in accordance with the X-URL encoding specification. The following attributes are currently supported and interpreted by the LDAP Policy Extension: • dn - the distinguished name or LDAP URL to perform policy resolution upon. (REQUIRED) • phase - the value may be specified as "1", "2", or "3", to view the intermediate stages of policy resolution. (default=3) • prefix - the value is the prefix to use when searching the directory for policy related attributes, i.e., <pfx>Policy or <pfx>Link. (default=edm) • debug - the value is the log level to use for this single query, a value of 9 or above will generate detailed logging in the Policy Server log file. (no default)

Domain Filtering

If you are using the Sentient Policy Server to create entitlements in your enterprise, you can filter out which domains the Policy Server will assign services from based on connect parameters.

If you are using the Policy Server with Sentient Patch Management, you will want to separate resolution of regular software services from those for Patch Management. The Policy Server filters services based on the dname passed on the radskman command line. The Policy Server configuration file, `pm.cfg`, contains filter settings in the format:


```
DNAME=<DOMAIN NAME>    { rule }
```

Where the DOMAIN NAME is the value passed in dname by RADISH. In the case of a Patch Management agent, this will be the dname parameter of radskman. Dname should be patch. If the filter name passed in dname is not found in `pm.cfg`, then the filter `DNAME=*` will be used.

The default configuration for these filters is shown below:

```
DNAME=*                { * !PATCHMGR !OS !SECURITY !AUDIT }
DNAME=PATCH           { PATCHMGR }
DNAME=OS               { OS }
DNAME=VM               { SECURITY }
DNAME=SECURITY         { SECURITY }
DNAME=AUDIT            { AUDIT }
```

In this configuration the default rule (*) will ignore PATCHMGR, OS, SECURITY and AUDIT domains and allow everything else as denoted by the use of an exclamation point (!). PATCH, OS, and AUDIT rules allow only policies for PATCH, OS, and AUDIT domains, respectively.

If the dname parameter passed by radskman is either VM or SECURITY, the rules allow only policies for SECURITY.

If, for instance, to allow any policies for AUDIT resolution the last filter to: `DNAME=USAGE { * }` is changed.

Chapter 13

Report Management

Sentient Database Configurations

The following sections list the modifications that the Reporting Server makes, based on whether you are using Oracle or SQL Server to store your databases.

Note: Reporting users must have INSERT, DELETE, SELECT, and UPDATE access to the tables created by each script (RRS_DEVICETABLE, RRS_DEVICECACHE and RRS_SESSION) in order for Reporting Server to work correctly.

Note: If you are using a 64-bit operating system, a 32-bit DSN is required on that server.

Sentient Oracle Database

A set of scripts are provided supplied with your Reporting Server media, located in the `\Prereq\Oracle\` directory. The scripts are located in the RIM, RUM, and RPM subdirectories. The following modifications are applied to each Oracle database the Reporting Server accesses.

- **Inventory:**
`Prereq\Oracle\RIM\RIM PreReq Creation Script.oracle`
Creates a view, called `DevicesPrimary`. In addition, a `DataAlias` table is created and populated.
- **Application Usage:**
`Prereq\Oracle\RUM\RUM PreReq Creation Script.oracle`
(Supports Oracle 10G and 11G. See the notes section of the script for additional information)
Adds a user defined function object called `fn_USAGESTATUS`, to calculate usage status.
- **Patch:**
`Prereq\Oracle\RPM\RPM PreReq Creation Script.oracle`
Adds a user defined function object called `fn_PATCHSTATUS`.

Sentient SQL Server Database

A set of SQL scripts are provided with the Reporting Server media, located in the `\Prereq\SQL\` directory. The scripts are located within the RIM, RUM, and RPM subdirectories. The following modifications are applied to each SQL Server database the Reporting Server accesses.

- **Inventory:**
`Prereq\SQL\RIM\RIM PreReq Creation Script.sql`
Creates a view called `DevicesPrimary`. This will add a user defined function object called `fn_DATAALIAS`.

- **Application Usage:**

`Prereq\SQL\RUM\RUM PreReq Creation Script.sql`

Adds a user defined function object called `fn_USAGESTATUS`, to calculate usage status.

- **Patch:**

`Prereq\SQL\RPM\RPM PreReq Creation Script.sql`

Adds a user defined function object called `fn_PATCHSTATUS`.

Enhancing Reporting Performance

Sentient (Usage Manager) provides several scripts and materialized views that can be applied to the Microsoft SQL Server and Oracle databases to enhance the reporting performance.

The scripts and views are available at:

- `Media\Usage\Optional Features\SQL Server` for Microsoft SQL Server database
- `Media\Usage\Optional Features\Oracle` for Oracle database

Using Views

There are two types of views, Standard Materialized Views and Filter Materialized Views. Both views enhance reporting performance. Either one can be optionally applied to a database. See the comments in the scripts for additional information about the functions of each view.

Note: The script names may abbreviate "Materialized" to "Mat", as in: `StepX_Define Filter Mat Tables and Indexes.sql`

Standard Materialized Views (SMV) - Converts all the views accessed by reports into tables. This view includes the index to enhance the query execution time. A feature where all the views (which is what the reports access) are converted into tables, and indexes are added to enhance the query speed.

Filtered Materialized Views (FMV) - Converts all the views accessed by reports into tables and requires filters to be applied before the views are converted into tables. The filters are stored in a separate table. For example, if a user selects `notepad.exe` as a filter, the FMV table is populated with the notepad details for all the devices. It is similar to SMV, but differs in that it requires filters to be applied at the time the views are converted into tables. The filters are stored in a separate table. As an example, if a filter for Notepad.exe is selected, the FMV table will be populated with only notepad details for all the devices.

Applying the scripts for SMV or FMV

1. Stop the service for the **RCA Knowledge Base Server**. The service may be stopped and started through the `Administrative Tools\Services` options of Windows Control Panel.
 2. Use normal procedures to execute the database scripts, in the given order, provided in the following locations:
 - For SQL Server:
`\SQL Server\Optional Features\Filter Materialized Views`
- Or

`\SQL Server\Optional Features\Standard Materialized Views`

- For Oracle:
`\Oracle\Optional Features\Filtered Materialized Views`

Or

`\Oracle\Optional Features\Standard Materialized Views`

Each of the above locations also includes a corresponding script to remove the view from your database. For example, the script name for the Microsoft SQL Server and Filtered Materialized Views is:

`SQLServer - Remove All Filter Mat Tables and Indexes.sql`

Utility Scripts

You as a database administrator can use the following scripts to enhance the reporting view performance:

- `Purge_Computer_Data.sql`: Deletes all data associated with the computer name. The computer name should be provided at the appropriate place in the script. The default value is MYCOMPUTER.
- `Purge_User_Data.sql`: Deletes all data associated with the computer name and the user name. The computer name and the user name should be provided at the appropriate place in the script. The default values are MYCOMPUTER and BOB.
- `Delete All Windows OS Files from Database.sql`: Deletes all Windows Operating System (OS) related files from the Usage Manager database.

Miscellaneous Scripts for Oracle

Miscellaneous scripts are additional scripts that can be applied along with the utility scripts to enhance the reporting view performance.

- `Optional_Create_Public_Synonyms.sql`: Creates public synonyms. The script may have to be edited for the Usage Manager's user names.
- `Optional_Drop_Public_Synonyms.sql`: Drops the public synonyms created by using the `Optional_Create_Public_Synonyms` script.
- `Step99a_DropAll.sql`: Drops all the tables present in the Usage Manager database.

Reporting Server Configuration Files

The Reporting Server configuration files include settings such as the ODBC DSN information and access credentials for each SQL database you are using, as well as LDAP Directory access root and credentials. Use the configuration file to enable or disable specific report types and features. The table Configuration files lists the configuration files available for modifications in the Reporting Server.

These configuration files are stored in the `<InstallDir>\ReportingServer\etc` folder. You can edit each configuration file manually using any text editor.

Note: If you need encrypted passwords, use the web browser to edit the configuration file. Passwords are automatically encrypted when the configuration file is generated. Using a text editor to edit the individual configuration files does not allow for creating encrypted passwords

Configuration files

File	Description
amp.cfg	Application Management Profile configuration
cba.cfg	Configuration Baseline Auditor configuration
ccm.cfg	Client Configuration Manager configuration
compliance.cfg	Compliance management configuration
ed.cfg	DDM Inventory configuration
rca.cfg	Sentient Management reports configuration
mobility.cfg	Contains configuration parameters for mobile device reporting.
rim.cfg	Inventory manager configuration
rpm.cfg	Patch manager configuration
rrs.cfg	Reporting server configuration and LDAP configuration
rum.cfg	Usage Manager configuration
scm.cfg	Settings management configuration
stm.cfg	Security tools management configuration
va.cfg	Virtual application management configuration
vm.cfg	Vulnerability Management configuration

The following sections provides details on the parameters available in each configuration file.

Application Management Profiles Configuration (AMP)

Application Management Profiles configuration values are stored within `amp.cfg`.

Application Management Profiles parameters

Parameter	Description
ENABLE	Enable Application Management Profiles
CACHEENABLE	Enable Application Management Profiles report caching.
CACHLIFE	Lifetime (in seconds) for cached files.

Client Configuration Manager Configuration

Client Configuration Manager configuration values are stored within `ccm.cfg`.

Client Configuration Manager Parameters

Parameter	Description
ENABLE	Enable Client Configuration Manager reports
CACHEENABLE	Enable Client Configuration Manager report caching
CACHELIFE	Lifetime (in seconds) for cached files.
URL	URL to Client Configuration Manager Host

Configuration Baseline Auditor Configuration

Configuration Baseline Auditor configuration values are stored within `cba.cfg`.

Configuration Baseline Auditor parameters

Parameter	Description
ENABLE	Enable Configuration Baseline Auditor
CACHEENABLE	Enable Configuration Baseline Auditor report caching.
CACHLIFE	Lifetime (in seconds) for cached files.
DEVICETABLE	Temporary Table in SQL to store the device list, format schema/owner.table (for example, aggregate.RRS_DEVICETABLE)
DATABASE	Database Type, if blank or missing, Reporting Server will detect. The only option available for DDM Inventory is mysql.
PREFIX	Schema/Owner for tables (for example, 'aggregate.')
DSN	ODBC DSN for Inventory database (for example, EDDB).
DSN_USER	ODBC user ID for Inventory database
DSN_PASSWD	ODBC password for Inventory database
NORMALIZECASE	Database is case sensitive when doing searches. Default is set to 0 (disabled). When enabled, all filters will convert input and searched fields to lower case for queries. Note: This may cause a performance issue with regard to indexes.

Compliance Management Configuration

Compliance Management configuration values are stored within `compliance.cfg`.

Compliance Management Configuration Parameters

Parameter	Description
ENABLE	Enable Compliance Management Reports
CACHEENABLE	Enable Compliance Management Report caching
CACHELIFE	Lifetime (in seconds) for cached files.
HOMEENABLE	Enables the report for viewing in the Reporting home page.
HOMESUMMARYENABLE	Enables the report summary in the Reporting home page.

DDM Inventory Configuration

DDM Inventory configuration values are stored within `ed.cfg`.

Enterprise Discover Configuration Parameters

Parameter	Description
ENABLE	Enable DDM Inventory reports
CACHEENABLE	Enable DDM Inventory report caching
CACHELIFE	Lifetime (in seconds) for cached files.
DEVICETABLE	Temporary Table in SQL to store the device list, format schema/owner.table (for example, aggregate.RRS_DEVICETABLE)
DATABASE	Database Type, if blank or missing, Reporting Server will detect. The only option available for DDM Inventory is mysql.
PREFIX	Schema/Owner for tables (for example, 'aggregate.')
DSN	ODBC DSN for Inventory database (for example, EDDB).
DSN_USER	ODBC user ID for Inventory database
DSN_PASSWD	ODBC password for Inventory database
NORMALIZECASE	Database is case sensitive when doing searches. Default is set to 0 (disabled). When enabled, all filters will convert input and searched fields to lower case for queries. Note: This may cause a performance issue with regard to indexes.
AGGREGATE	Enable Aggregate Reporting. If enabled (1), all reports will use aggregate data. If disabled (0), all reports will use local data only.

Sentient Management Reports Configuration

Sentient Management Reports configuration values are stored within `rca.cfg`.

Sentient Management Reports Configuration Parameters

Parameter	Description
ENABLE	Enable Sentient Management Reports.
CACHEENABLE	Enable Sentient Management Reports caching.
CACHELIFE	Lifetime (in seconds) for cached files.

Inventory Manager Configuration

Inventory Manager configuration values are stored in the `rim.cfg`.

Inventory Manager Configuration Parameters

Parameter	Description
ENABLE	Enable Inventory Manager reports
CACHEENABLE	Enable Inventory Manager report caching
CACHELIFE	Lifetime (in seconds) for cached files.
DEVICETABLE	Temporary Table in SQL to store the device list, format schema/owner.table (for example, dbo.RRS_DEVICETABLE')
DATABASE	Database Type, if blank or missing, Reporting Server will detect. Options: sql, oracle
PREFIX	Schema/Owner for tables (for example, 'dbo.')
DSN	ODBC DSN for Inventory database
DSN_USER	ODBC user ID for Inventory database
DSN_PASSWD	ODBC password for Inventory database
NORMALIZECASE	Database is case sensitive when doing searches. Default is set to 0 (disabled). When enabled, all filters will convert input and searched fields to lower case for queries. Note: This may cause a performance issue with regard to indexes.
IGNOREWILDCARDS	The Reporting Server interprets the characters that you specify in this parameter as literal characters, rather

Parameter	Description
	than considering them as wildcard characters.
USETEMPDEVTABLE	Ensures that the temporary table is used for RRS_DEVICETABLE. It is recommended that you do modify this parameter.
HOMEPAGEENABLE	Enables the report for viewing in the Reporting home page.
HOMEAGESUMMARYENABLE	Enables the report summary in the Reporting home page.

Mobile Device Configuration

The configuration parameters for mobile device reporting are stored in the file `mobility.cfg`.

Mobile device configuration parameters

Parameter	Description
ENABLE	Enables the reporting for mobile devices.
CACHEENABLE	Enables caching for mobile device reports.
CACHLIFE	Lifetime (in seconds) for cached files.
HOMEPAGEENABLE	Enables the report for viewing in the Reporting home page.
HOMEAGESUMMARYENABLE	Enables the report summary in the Reporting home page.

Patch Manager Configuration

Patch Manager configuration values are stored in `rpm.cfg`.

Patch Manager configuration parameters

Parameter	Description
ENABLE	Enable Patch Manager reports
CACHEENABLE	Enable Patch Manager report caching.
CACHLIFE	Lifetime (in seconds) for cached files.
DEVICETABLE	Temporary Table in SQL to store the device list, format schema/owner.table (for example, 'dbo.RRS_DEVICETABLE')
DATABASE	Database Type, if blank or missing, Reporting Server will detect. Options: sql, oracle

Parameter	Description
PREFIX	Schema/Owner for tables (for example, 'dbo.')
DSN	ODBC DSN for Patch Database
DSN_USER	ODBC User ID for Patch Database
DSN_PASSWD	ODBC Password for Patch Database
NOLOCK	SQL Option to disable locking on query
NORMALIZECASE	Database is case sensitive when doing searches. Default is set to 0 (disabled). When enabled, all filters will convert input and searched fields to lower case for queries. Note: This may cause a performance issue with regard to indexes.
IGNOREWILDCARDS	The Reporting Server interprets the characters that you specify in this parameter as literal characters, rather than considering them as wildcard characters.
USETEMPDEVTABLE	Ensures that the temporary table is used for RRS_DEVICETABLE. It is recommended that you do modify this parameter.
HOMEENABLE	Enables the report for viewing in the Reporting home page.
HOMESUMMARYENABLE	Enables the report summary in the Reporting home page.

Reporting Server Configuration

The Reporting Server configuration values are stored in `rrs.cfg`.

Reporting Server Configuration Parameters

Parameter	Description
LOGLEVEL	Reporting Server logging level
LANGUAGE	Reporting Server UI language
DEFAULTVIEW	Default View to load when none is specified
DEVICELIST	Show report data when no filters are specified
DEVICELISTCOUNT	Threshold to use RRS_DEVICELIST table.
EXPORTMODE	CSV or tab export mode

Parameter	Description
DEFENABLE	Enable default reports.
RMPLOGON	Enable Portal Logon Support
RMPIP	IP/Host Address for Portal
RMPPORT	Port for Portal
RMPUSESSL	Use SSL connection to Portal. When SSL enabled, Portal logon verification is done via SSL.
SSL_CADIR	CA Certificates directory
SSL_CAFILE	CA Certificates file
dl_datefmt	Date format for reports
dl_timefmt	Time format for reports
dl_gmt	Show report data in GMT Time
FORCEDFILTERNAME	Forced filter name. Applies a filter to all reports. This option is not visible.
FORCEDFILTERVALUE	Forced filter value. Applies a filter to all reports. This option is not visible.
POPULATEFROMDB	Populates reporting data from Sentient Database. Set the value of this parameter to 1 to fetch the reporting data from Sentient Database, when Directory/Group Filters are being used. If not set, the reporting data will be populated from the Active Directory. This enhancement improves the performance of the Reporting Server.

Security Tools Management (STM) Configuration

Security Tools Management configuration values are stored within `stm.cfg`.

Security Tools Management Configuration Parameters

Parameter	Description
ENABLE	Enable Security Tools Management Reports
CACHEENABLE	Enable Security Tools Management Report caching
CACHELIFE	Lifetime (in seconds) for cached files.
HOMEPAGEENABLE	Enables the report for viewing in the Reporting home page.
HOMEPAGESUMMARYENABLE	Enables the report summary in the Reporting home page.

Settings Management Configuration

Settings Management configuration values are stored within `scm.cfg`.

Settings Management Configuration Parameters

Parameter	Description
ENABLE	Enable Settings Management reports
CACHEENABLE	Enable Settings Management report caching
CACHELIFE	Lifetime (in seconds) for cached files.

Virtualization Management Configuration

Virtualization Management configuration values are stored within `va.cfg`.

Virtualization Management Configuration Parameters

Parameter	Description
ENABLE	Enable Virtualization Management reports
CACHEENABLE	Enable Virtualization Management report caching
CACHELIFE	Lifetime (in seconds) for cached files.

Vulnerability Management Configuration

Vulnerability Management configuration values are stored within `vm.cfg`.

Vulnerability Management Configuration Parameters

Parameter	Description
ENABLE	Enable Vulnerability Management reports
CACHEENABLE	Enable Vulnerability Management report caching
CACHELIFE	Lifetime (in seconds) for cached files.
DEVICETABLE	Temporary table in the database to store the device list, format schema/owner.table. For SQL, the format is <code>dbo.RRS_DEVICETABLE</code> . For Oracle, the format is <code>dbo.#RRS_DEVICETABLE</code> .
DATABASE	The database type. If the entry is blank or missing, the Reporting Server automatically detects the database. Options: <code>sql</code> , <code>oracle</code>
PREFIX	Schema/Owner for tables (for example, <code>'dbo.'</code>)
DSN	ODBC DSN for

Parameter	Description
DSN_USER	ODBC user ID for
DSN_PASSWD	ODBC password for
NORMALIZECASE	Database is case sensitive when doing searches. Default is set to 0 (disabled). When enabled, all filters will convert input and searched fields to lower case for queries. Note: This may cause a performance issue with regard to indexes.
HOMEENABLE	Enables the report for viewing in the Reporting home page.
HOMESUMMARYENABLE	Enables the report summary in the Reporting home page.

Application Usage Manager Configuration

Application Usage Manager configuration values are stored in `rum.cfg`.

Usage Manager Configuration Parameters

Parameter	Description
ENABLE	Enable Usage Manager reports
CACHEENABLE	Enable Usage Manager report caching.
CACHELIFE	Lifetime (in seconds) for cached files.
DEVICETABLE	Temporary Table in SQL to store the device list, format schema/owner.table (for example, 'dbo.RRS_DEVICETABLE')
DATABASE	Database Type, if blank or missing, Reporting Server will detect. Options: sql, oracle
PREFIX	Schema/Owner for tables (for example, 'dbo.')
DSN	ODBC DSN for Usage Database
DSN_USER	ODBC User ID for Usage Database
DSN_PASSWD	ODBC Password for Usage Database
NORMALIZECASE	Database is case sensitive when doing searches. Default is set to 0 (disabled). When enabled, all filters will convert input and searched fields to lower case for queries.

Parameter	Description
	Note: This may cause a performance issue with regard to indexes.
IGNOREWILDCARDS	The Reporting Server interprets the characters that you specify in this parameter as literal characters, rather than considering them as wildcard characters.
USETEMPDEVTABLE	Ensures that the temporary table is used for RRS_DEVICETABLE. It is recommended that you do modify this parameter.
HOMEPAGEENABLE	Enables the report for viewing in the Reporting home page.
HOMEPAGESUMMARYENABLE	Enables the report summary in the Reporting home page.

LDAP Configuration

The LDAP configuration values are stored within `rrs.cfg`.

LDAP Configuration Parameters

Parameter	Description
ENABLE	Enable LDAP browsing and filtering
CACHEENABLE	Enable LDAP browsing and filtering caching
CACHELIFE	Lifetime (in seconds) for cached files.
TYPE	Type of LDAP Connection • ldap = Standard LDAP Connection. If SSL is enabled, requires SSL Certificate File. • dsml = Directory Service Markup-Language Connection • rmp-ws = RMP Webservices (Used to connect to Portal)
SERVER	IP/Host of LDAP Server (for example, ldap.psl.com)
PORT	LDAP port (for example, 389)
BASE	Base DN to start browsing at in UI (for example, ou=groups,dc=hp,dc=com)
BIND	BIND DN (for example, dc=hp,dc=com)
WSSERVICEACT	Use service account for type rmp-ws. If true, then will use the user credential configured in this configuration file. If false, will use the logged on user credentials. Service account credentials are still required in

Parameter	Description
	either case.
USER	LDAP User ID (for example, administrator@ldap.psl.com)
PASS	LDAP Password
PAGING	LDAP paging support
PAGESIZE	LDAP page size
SHOWFRIENDLY	Show friendly names when browsing LDAP
SECURE	Use SSL connection to LDAP
CERTFILE	Path and filename to SSL certificate. Required when LDAP Server Type set to ldap.

Configuring Reporting Server

The Reporting Server configuration files includes settings such as the ODBC DSN information and access credentials for each SQL database you are using, as well as LDAP Directory access root and credentials. Use the configuration files to enable or disable specific report types and features.

Note: If you need encrypted passwords, use the web browser to edit the configuration file. Passwords are automatically encrypted when the configuration file is generated. Using a text editor to edit the individual configuration files does not allow for creating encrypted passwords.

The following sections detail specific configuration tasks available with the Reporting Server configuration files.

Normalizing Case Sensitive Databases

The configuration files contain an option for normalizing character case when applying filters to your databases. By default, Oracle is a case-sensitive database. So for example, 'A' is not equal to 'a' while SQL databases are *not* case-sensitive, so 'A' is equal to 'a.' Microsoft SQL Server databases, by default, are *not* case-sensitive.

When the normalize case option is off (set to 0), Reporting Server does no post processing on any filters. This means for an Oracle database, if you are looking for a device called DeviceA, but in the Oracle database it is listed as DEVICEA, no match is found. In this example, a SQL database *would* return a match.

If the case-sensitive option is turned on (set to 1), searching for DeviceA in the Oracle database will return a match, DEVICEA. Turning the option on for a SQL database will turn on case sensitivity and, in this example, no match would be found.

To enable normalizing character case for Inventory Management reports:

1. Use a text editor to edit the `rim.cfg` file, located in the `etc` folder of the Reporting Server directory.
2. Set `NORMALIZECASE` to 1.
3. Save your changes and refresh the Reporting Server home page.

Note: When this option is turned on (set to 1), indexes are not used which will result in reporting queries taking much longer to generate results.

The `NORMALIZECASE` settings are to be applied for individual configuration files.

Customizing Reports

The Reporting Server allows extensive report customization by modifying any of the reporting object files. Reporting object files must be extracted or created manually and added to the Reporting Server directory structure.

Extracting Report Packs

Report packs are stored in the Reporting Server's `<InstallDir>\ReportingServer\reportpacks` directory. Each file contains Reporting Server object files for specific report types.

Report Packs

Report Pack File	Reports
<code>cba.kit</code>	Configuration Baseline Auditor
<code>ccm.kit</code>	CA Starter and Standard
<code>compliance.kit</code>	Compliance Management
<code>ed.kit</code>	DDM Inventory
<code>hpca.kit</code>	Sentient Management Reports
<code>mobility.kit</code>	Mobile devices reports. A mobile device can either be a smartphone or a tablet running on an Android or iOS operating system. For more details on the supported operating system versions, see the Sentient Enterprise Support Matrix available at the URL: https://support.accelerite.com/hc/en-us/categories/200146220 .
<code>rim.kit</code>	Inventory Manager
<code>rpm.kit</code>	Patch Manager
<code>rrs.kit</code>	Reporting Server and LDAP
<code>rum.kit</code>	Application Usage Manager

Report Pack File	Reports
scm.kit	Application Management Profiles
stm.kit	Security Tools Management
vm.kit	Vulnerability Management

When extracted, report packs create the file structure necessary to modify reporting objects. Use the **extract** utility to extract reporting object files.

To extract files from a report pack:

1. From a command prompt, switch to your Reporting Server's `\bin` directory.
2. Type the following command (This example will extract the contents of `rum.kit`. Replace `rum` with the report pack you want to extract. The file extension, `.kit`, is not required.):
`extract rum`
3. Press **Enter**.
The reporting files are extracted to the Reporting Server `\reportpacks` directory in a new folder. The example above created the folder, `\reportpacks\rum\objects`, which contains all reporting object files needed to make any modifications for Application Usage Manager reports.

Note: Extracted files take precedence over files contained within a reporting pack. For example, if you've created the directory structure, `\reportpacks\rum\objects`, which contains modified reporting object files, these modifications take precedence over the reporting objects contained within `rum.kit`. This helps maintain any customization you may have made in the event of an update to Reporting Server where a newer `rum.kit` is supplied.

Reporting Object Files

Reporting object files determine what data you will see and in what format it is presented on the Reporting Server Web page.

There are multiple reporting object file types that you can modify. Each type is located within a separate sub-directory within the appropriate objects folder. For example, an extracted `rum.kit` will create the `\reportpacks\rum\objects` directory.

Note: Customizing Reporting Server reporting object files requires a basic knowledge of XML. Familiarity with tcl and HTML is also beneficial.

When generating a reporting page, the Reporting Server looks for all files of a specific extension type, `*.view`, for example, within the directories mentioned above.

Backing up Your Reporting Object Files

Before you begin modifying any reporting object files, make sure to backup the original files. It is recommended that you rename any of the files you will modify with an easily identifiable designator, your company name or initials, for example. If your company initials are ABC, then you could copy and rename the View Group object, Hardware Reports.viewgroup to ABC_HardwareReports.viewgroup.

Since the Reporting Server only looks for specific extensions, renaming your modified files allows easy incorporation into your reporting environment. Also, any updates to the Reporting Server files will be easily incorporated into your existing configuration without undoing any of your report customizations.

Understanding the Reporting Object Files

Before you customize any reporting object files, become familiar with the purpose of each file and how they are constructed.

Reporting Object File Construction

Each reporting object file is designed using XML (Extendable Markup Language) as well as tcl (Tool-Command Language) and HTML. You should have a basic understanding of each of these programming languages before you attempt to make any modifications to your reporting object files. Use any text editor to modify and save new reporting object files.

Reporting Object File Types

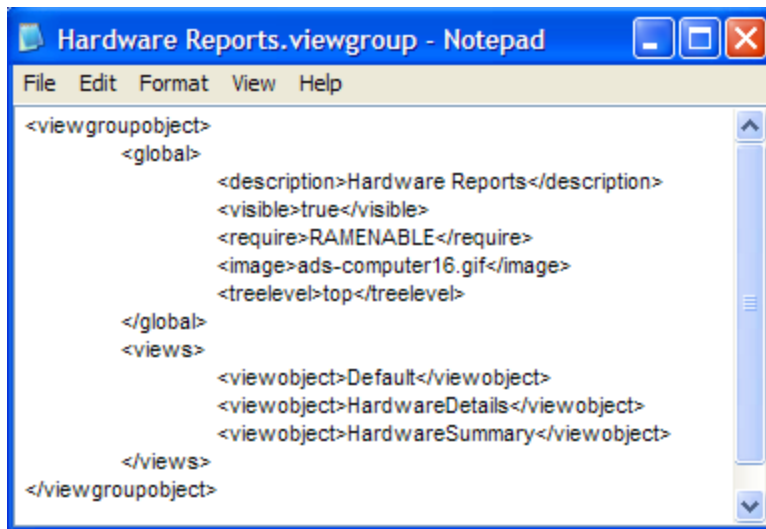
Several reporting object file types can be modified to generate custom report pages. These include:

- View Groups
- Views
- Filter Groups
- Filters
- Windows

View Group Objects and View Objects

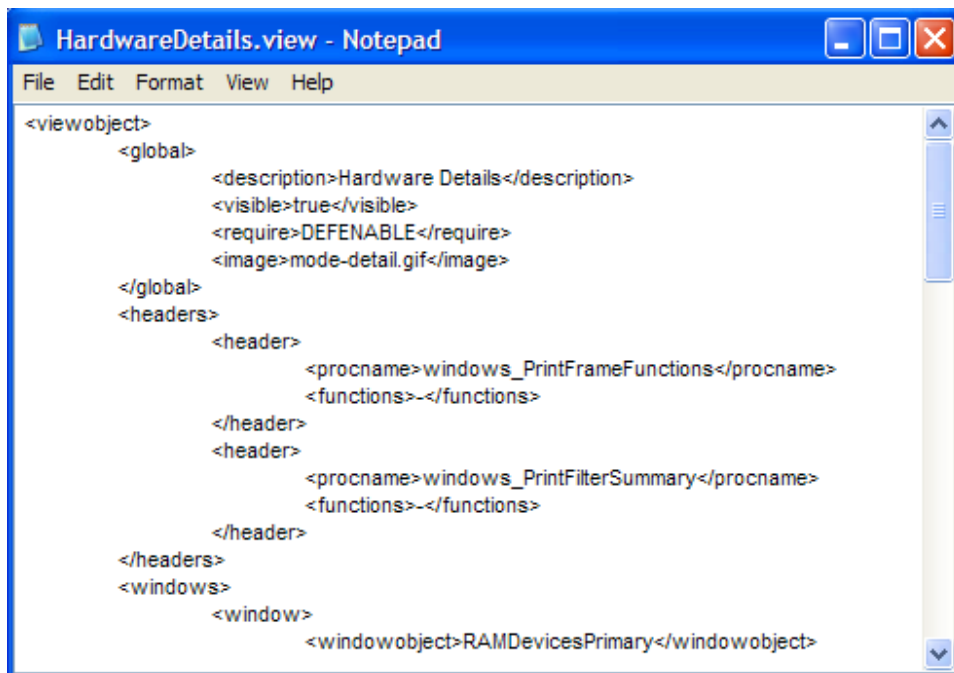
View Group (`*.viewgroup`) objects contain a selectable list of View objects available for that category. Each View Group contains a specific set of available Views. A view group can include another view group in its list to allow for multiple levels in the tree view.

Sample View Group object file



View (* .view) objects determine the currently active window object displayed on the right side of the Reporting Server web page.

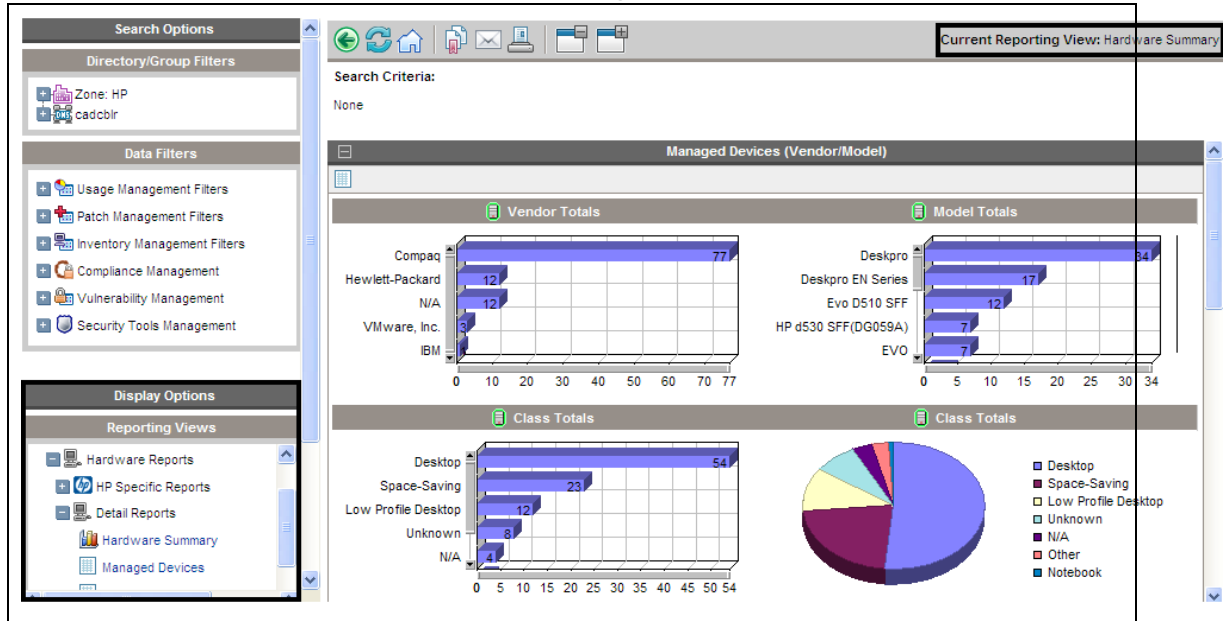
Part of a sample View object file



Modifying View Group objects and View object files will determine what is displayed on the left-hand side of the Reporting Server Web page in the two drop-down list boxes within the Display Controls/Reporting Views section.

The currently active View object is displayed in the top-right corner of the page after **Current Reporting View**.

View Groups, Views and the Current Reporting View

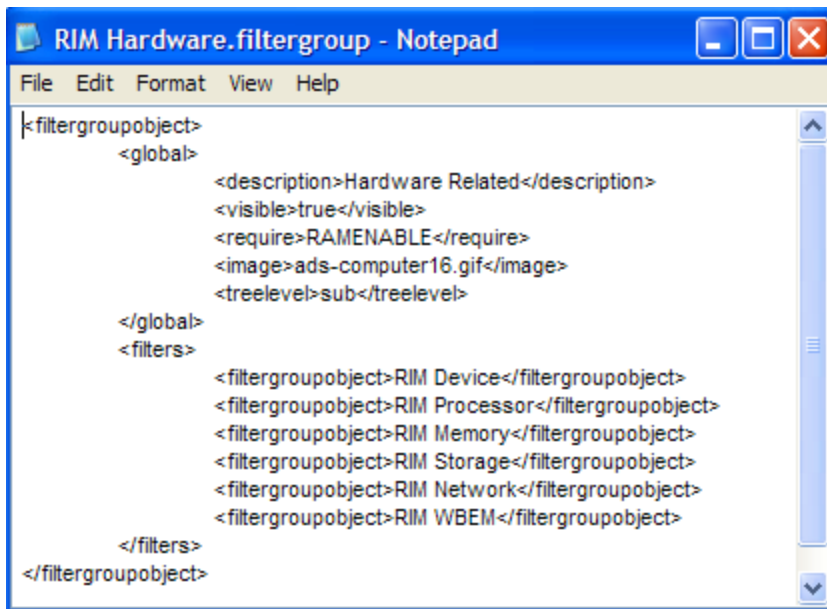


In addition to adding or altering Display Controls, View object files are used to determine which window objects are present. View objects can also determine specific Window object settings using the Window Object Overrides section. The parameter values within this section take precedence over any parameter values within a Window object file.

Filter Group Objects and Filter Objects

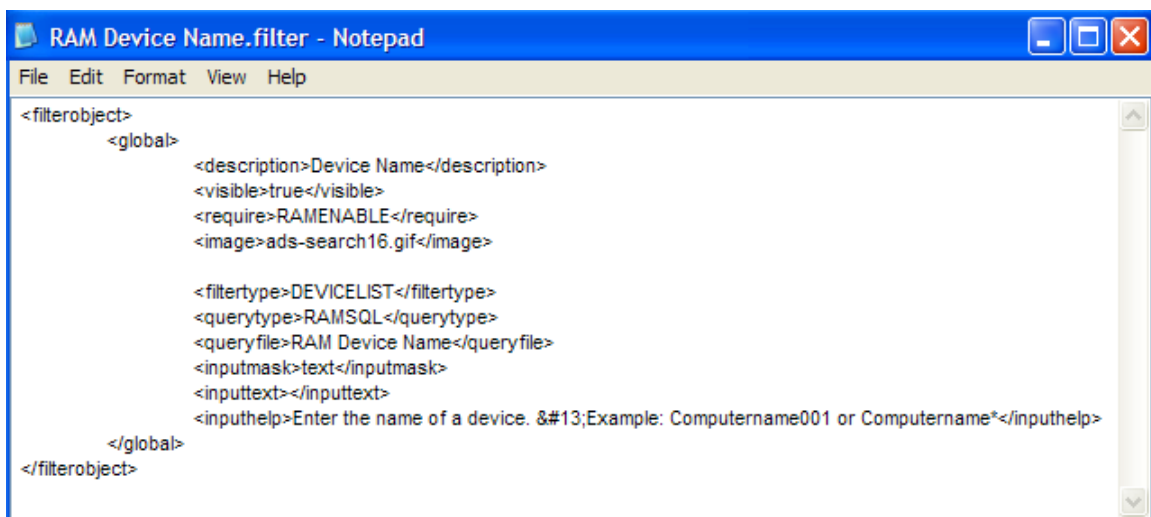
Filter Group (*.filtergroup) objects determine a selectable list of Filter objects. Like View Groups and Views, Filter Groups contain a specific set of available Filters. Depending on the Filter Groups selected, different Filter objects can be made available. Filter group objects can contain other Filter Group objects to allow for multiple levels in the tree view.

Sample Filter Group object file



Filter (*.filter) object files define which data is displayed within each reporting page.

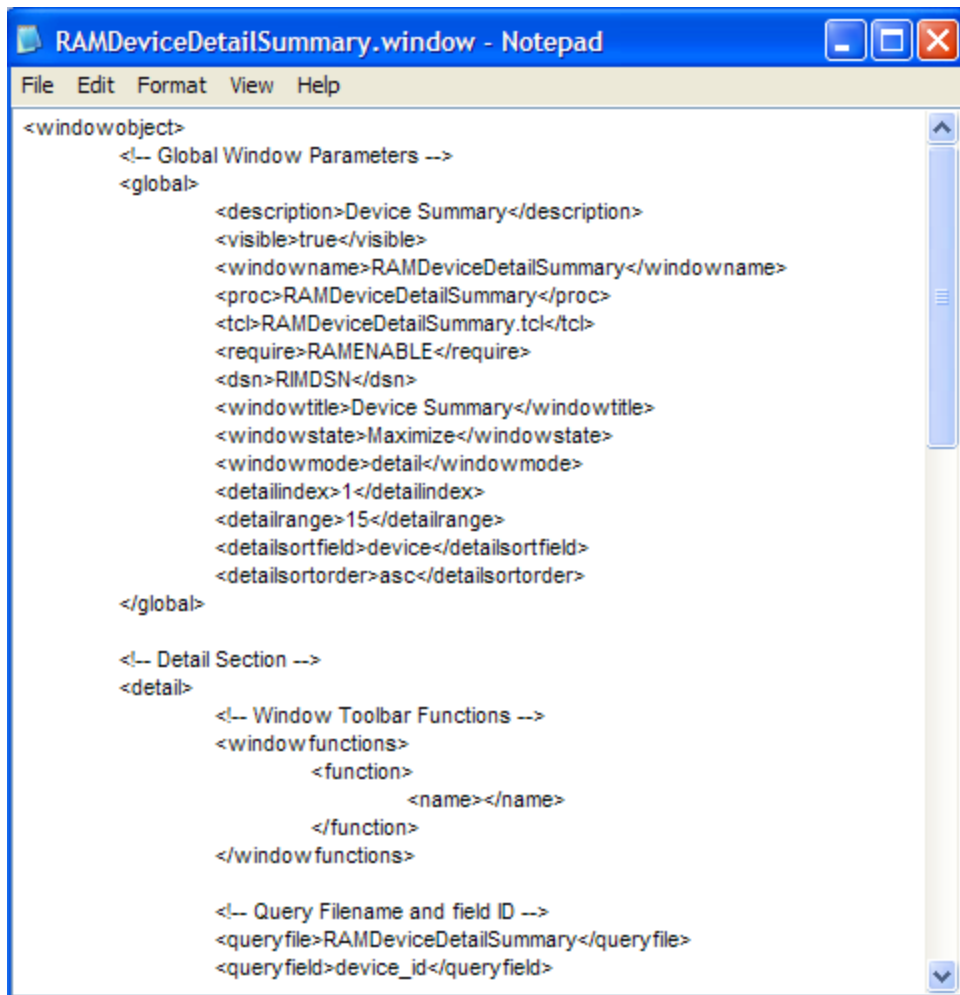
Sample Filter object file



Window Objects

Window object files (*.window) define the data layout, including column headings, icons, sort fields, default window mode, default window state, drill down elements and column data appearances.

Part of a sample Window object file



Each existing reporting object file type can be modified or copied to create new reporting pages.

Modifying Reporting Object Files

Use a text editor, like Notepad.exe, to modify your reporting object files. Each file is built using XML, tcl and HTML. Make sure you are familiar with these programming languages before you begin to modify these files.

Parameters within the reporting object files are defined using XML. For example, the global section description parameter within the HardwareDetails.view object file is defined as:

```
<description>Hardware Details</description>
```

Note the leading and trailing XML descriptor tags.

To change the HardwareDetails.view object, global description to Hardware Specifics, change the text between the XML tags to:

```
<description>Hardware Specifics</description>
```

After the file is saved, the next time you view or refresh your Reporting Server Web page, the selectable entry within the Views drop-down list box, Hardware Details, will have been replaced with Hardware Specifics.

The following paragraphs and tables explain each object file section and their associated parameters and values. Modify any of these values to change or create reporting objects.

Modifying View Group Objects

View Group object files (*.viewgroup) contain two configurable sections: global and views. View groups can contain other View Groups to allow for multiple tree-view levels.

View Group Object Global Section

The global section is used to define the View Group name within the Display Controls drop-down list box, whether or not this View Group is visible and what settings are required for this View Group to be enabled. A sample <global> section follows:

```
<global>

  <description>Hardware Reports</description>

  <visible>true</visible>

  <require>RAMENABLE</require>

  <image>ads-computer16.gif</image>

  <treelevel>top</treelevel>

</global>
```

View Group Object Global Section Parameters

Parameter	Description
<description>	What is displayed within the View Group drop-down list.
<visible>	True = This view group will be present in the View Group drop-down list. False = Hide this view group selection.
<require>	Prerequisite files or configuration file settings required for this View Group to be available.
<image>	Image used for tree view icon.
<treelevel>	top = load and show at top level. sub = load and show at sub level.

View Group Object Views Section

The views section determines which view objects are available when the view group is selected. A sample <views> section follows:


```
<views>?
  <viewobject>Default</viewobject>
  <viewobject>HardwareDetails</viewobject>
  <viewobject>HardwareSummary</viewobject>
</views>
```

View Group Object Views Section Parameters

Parameter	Description
<viewobject>	View object name. Define more than one view objects by adding additional viewobject lines.

Modifying View Objects

View object files (*.view) contain the following configurable sections: Global, Header and Windows.

View Object Global Section

The global section is used to define the View name within the Display Controls drop-down list box, whether or not this View is visible and what settings are required for this View to be enabled. A sample <global> section follows:

```
<global>
  <description>Hardware Details</description>
  <visible>true</visible>
  <require>DEFENABLE</require>
  <image>mode-detail.gif</image>
</global>
```

View Object Global Section Parameters

Parameter	Description
<description>	What is displayed within the Views drop-down list.
<visible>	True = This view will be present in the Views drop-down list. False = Hide this view selection.
<require>	Prerequisite files or configuration file settings required for this View to be available.
<image>	Image used for tree-view icon.

View Object Header Section

The View object header section contains parameters used to define what items are available within each window heading. A sample `<header>` section follows:

```
<headers>

  <header>

    <procname>windows_PrintFrameFunctions</procname>

    <functions>-</functions>

  </header>

  <header>

    <procname>PrintFilterSummary</procname>

    <functions>-</functions>

  </header>

</headers>
```

View Object Header Parameters

Parameter	Description
<code><procname></code>	Tcl procedure name to call for the header. The defaults are <code>windows_printframefunctions</code> and <code>windows_printfiltersummary</code> . The default functions draw the respective data displayed.
<code><functions></code>	List of parameters to pass to the function. Currently, most values are set to – (dash) as this is reserved for future use (with the exception of the sub-views, which have the value back to enable the back button).

View Object Windows Section

The windows section determines which window objects are available when the view is selected. The Window Object Overrides section allows you to alter the appearance of any window object by overriding parameter values set within the window object with new values. This allows for conformity between window objects available with a certain view object. A sample `<windows>` section follows:

```
<windows>

  <window>

    <windowobject>RAMDevicesPrimary</windowobject>

    <!-- Window Object Overrides -->

    <windowstate>Maximize</windowstate>

    <windowmode>detail</windowmode>
```

```
<detailindex>1</detailindex>

<detailrange>15</detailrange>

<detailsortfield>-</detailsortfield>

<detailsortorder>-</detailsortorder>

</window>

</windows>
```

View Object Windows Parameters

Parameter	Description
<windowobject>	Determines the window object to load.
Window Object Overrides	
<windowstate>	Determines the window object state.
<windowmode>	Determines window object mode.
<detailindex>	Starting record number. Usually will be 1 (to start at record number 1).
<detailrange>	Number of records to display at one time.
<detailsortfield>	Default SQL field name by which to sort data results.
<detailsortorder>	Sort order. asc = sort ascending. des = sort descending.

Modifying Filter Group Objects

Filter Group objects (*filtergroup) contain two configurable sections, global and filters. Filter Groups can contain other Filter Groups to allow for multiple levels in the tree view.

Filter Group Object Global Section

The global section is used to define the Filter Group name within the Filter Group drop-down list box, whether or not this Filter Group is visible and what settings are required for this Filter Group to be enabled. A sample <global> section follows:

```
<global>

  <description>Hardware Related Filters</description>

  <visible>true</visible>

  <require>RAMENABLE</require>

  <image>ads-computer16.gif</image>
```

```
<treelevel>sub</treelevel>

</global>
```

Filter Group Object Global Parameters

Parameter	Description
<description>	What is displayed within the Filter Group drop-down list.
<visible>	True = This filter group will be present in the Filter Group drop-down list. False = Hide this filter group selection.
<require>	Prerequisite files or configuration file settings required for this filter group to be available.
<image>	Image used for tree view icon.
<treelevel>	top = load and show at top level. sub = object is a sub-level branch.

Filter Group Object Filters Section

The filters section determines which filter objects are available when the filter group is selected. A sample <filters> section follows:

```
<filters>

  <filterobject>RIM Device Vendor</filterobject>

  <filterobject>RIM Device Model</filterobject>

  <filterobject>RAM Device Name</filterobject>

  <filterobject>RIM Device Class</filterobject>

  <filterobject>RIM Device Serial Number</filterobject>

  <filterobject>RAM Device Memory</filterobject>

  <filterobject>RIM Device CPU Count</filterobject>

  <filterobject>RIM Device CPU Speed Less</filterobject>

  <filterobject>RIM Device CPU Speed More</filterobject>

  <filterobject>RAM Drive Space Free</filterobject>

</filters>
```

Filter Group Object Filters Section Parameters

Parameter	Description
<filterobject>	Filter object name. Define more than one filter objects by adding additional filter object lines.

Modifying Filter Objects

Filter object files (*.filter) contain a global configurable section.

Filter Object Global Section

The global section determines the filter name displayed, filter type, query type and parameters as well as any required input configurations. A sample <global> section follows:

```
<global>

  <description>Device ID</description>

  <visible>true</visible>

  <require>RAMENABLE</require>

  <image>ads-search16.gif</image>

  <querytype>RAMSQL</querytype>

  <queryfile>RAM Device ID</queryfile>

  <inputmask>text</inputmask>

  <inputtext></inputtext>

  <inputhelp>Enter the device_id of a device.</inputhelp>

</global>
```

Filter Object Global Section Parameters

Parameter	Description
<description>	What is displayed within the Filter drop-down list.
<visible>	True = This view will be present in the Filter drop-down list. False = Hide this view selection.
<require>	Prerequisite files or configuration file settings required for this Filter to be available.
<image>	Image used for tree view icon.
<querytype>	Determines from where the data is be retrieved. Either, RAMSQL (for Application Manager database), RIMSQL (for Inventory Manager database), RUMSQL (for Application Usage Manager database) or RPMSQL (for Patch Manager database).
<queryfile>	The SQL command used to obtain the data.
<inputmask>	Valid input mask values are as follows: date – Inserts a calendar icon allowing the user to select a date.

Parameter	Description
	text – Simple text input. dropdown - Create a selectable dropdown list with predefined filters. Edit inputtext to enter values for the drop-down list. none - Disables input field. Requires the filter to be hard coded (for example, Device last connect > 30 days). numeric – A numeric value (for example, 256). numeric_list – A list of numeric values (for example, @256 384). decimal - A decimal value (for example, 5.0). decimal_list – A list of decimal values (for example, @5.0 4.0).
<inputtext>	Input mask values. For text, numeric, numeric_list, decimal, and decimal_list input masks, whatever is entered here will display as the default text input box value.
	For date input mask, will display default date. Must be a valid date format. For dropdown input mask value, use a space-delimited list to create the list values. For example: <pre><inputtext>a b c</inputtext></pre> Creates a dropdown list with the elements a b and c. Alternatively, a SQL query can be used to populate the dropdown list. For example: <pre><inputtext>[sql execute RIMDSN "select distinct os from deviceconfig"]</inputtext></pre> Creates a dropdown list of OS values.
<inputhelp>	Help icon text displayed on mouse-over.

Windows Toolbar Functions Sub-Section

This section allows for icons to be added to the window toolbar. In order to be useful, the icon must match the window mode (a graph cannot be exported, for example).

- **ExportCSV** adds the export to CSV Icon (Detail mode only). This button enables you to export the report as a Comma Separated Value (CSV) file.
- **ExportIQY** adds the export to IQY Icon (Detail mode only). This button enables you to export the report as an Internet Query file, which allows a live link to the report when viewed using Microsoft Excel.

- **ViewMode** adds the icon to change to graphical mode or when in graphical mode, adds the icon to change to detail mode. A corresponding mode must exist in the object.
- **NavBar** adds the right side navigation options (detail mode only).

Modifying Window Objects

Window object files (*.window) contain three main configurable sections, global, details and graph.

Window Object Global Section

The global section determines the window object description and initial settings. A view object may override some window object settings. A sample `<global>` section follows:

```
<global>

  <description>Device Summary</description>

  <visible>true</visible>

  <windowname>RAMDeviceDetailSummary</windowname>

  <proc>RAMDeviceDetailSummary</proc>

  <tcl>RAMDeviceDetailSummary.tcl</tcl>

  <require>RAMENABLE</require>

  <datasource>RIM</datasource>

  <windowtitle>Device Summary</windowtitle>

  <windowstate>Maximize</windowstate>

  <windowmode>detail</windowmode>

  <detailindex>1</detailindex>

  <detailrange>15</detailrange>

  <detailsortfield>device</detailsortfield>

  <detailsortorder>asc</detailsortorder>

</global>
```

Window Object Global Section Parameters

Parameter	Description
<description>	Window object description.
<visible>	True = This window will be visible. False = Hide this window object.
<windowname>	A unique descriptive name, same as the filename without the

Parameter	Description
	extension.
<proc>	For internal use only.
<tcl>	Tcl script file to be used.
<require>	Prerequisite files or configuration file settings required for this Filter to be available.
<datasource>	Prerequisite data source setting required within the configuration file.
<windowtitle>	Window object title displayed.
<windowstate>	Window object state.
<windowmode>	Window object mode.
<detailindex>	Starting record number. Usually will be 1 (to start at record number 1).
<detailrange>	Number of records to display at one time.
<detailsortfield>	Default SQL field name by which to sort data results.
<detailsortorder>	asc = ascending sort order. des = descending sort order.

Window Object Detail Section

The detail section determines the window toolbar functions, query files and detail column settings. Within the details section are three sub-sections: Windows Toolbar Functions, Query Filename and Field ID and Detail Mode Columns. A sample <detail> section follows:

```
<detail>

  <!-- Window Toolbar Functions -->

  <windowfunctions>

    <function>

      <name></name>

    </function>

  </windowfunctions>

  <!-- Query Filename and field ID -->

  <queryfile>RAMDeviceDetailSummary</queryfile>

  <queryfield>device_id</queryfield>

  <!-- Detail Mode Columns -->
```



```
<actions>
  <action>
    <enabled>0</enabled>
    <image></image>
    <columnname></columnname>
    <url></url>
    <urlalt></urlalt>
    <urltarget></urltarget>
  </action>
</actions>
<columns>
  <column>
    <fieldname></fieldname>
    <fieldtype></fieldtype>
    <fieldsettings></fieldsettings>
    <columnname></columnname>
    <columnsettings></columnsettings>
    <url></url>
    <urlalt></urlalt>
    <urltarget></urltarget>
  </column>
</columns>
</detail>
```

Query Filename and Field ID Sub-Section

This section determines which query file should be used as well as which query field is required. The query filename used automatically appends an extension of .sql or .oracle.

Inventory Manager and Application Manager data require a queryfield of device_id. The Patch Manager and Application Usage Manager data requires a queryfield of device_name.

Detail Mode Columns Sub-Section

The Detail Mode Columns sub-section determines the default layout for each detail column (the first two columns to the left displayed for each window object). Two additional sub-sections, `<actions>` and `<columns>` are included within this section.

Action sub-section parameters

Parameter	Description
<enabled>	1 = Show detail column. 0 = Hide detail column.
<image>	Determines whether an image file is used to portray the data. Image files are stored within the <code>ReportingServer/html/images</code> folder.
<columnname>	The name displayed for the column.
<url>	Hyperlink value. Can be an internal reporting function or an external url. For example, <code>http://[index/ \$row 2]:5800</code> , will add a link using the value from row 2 from the SQL query as a symbolic.
<urlalt>	Mouse-over text.
<urltarget>	Determines whether or not a new window is opened. For example, clicking the Remote Control (VNC) icon opens a new window.

Columns sub-section parameters

Parameter	Description														
<fieldname>	Must match SQL field name in the query 1 for 1 match.														
<fieldtype>	Set to one of the following values: <table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>text</td><td>Displays the text. This is the default.</td></tr> <tr> <td>date</td><td>Converts the incoming value to YYYY-MM-DD HH:MM:SS format.</td></tr> <tr> <td>numeric</td><td>Displays the number in comma separated format. For example, 12345 is displayed as 12,345.</td></tr> <tr> <td>image</td><td>Treats the contents of this column as image.</td></tr> <tr> <td>dropdown</td><td>Creates a drop-down list based on space separated values.</td></tr> <tr> <td>datecfg</td><td>Converts the incoming value to YYYY-MM-DD HH:MM:SS format. This value is again converted to gmt or local zone</td></tr> </table>	Value	Description	text	Displays the text. This is the default.	date	Converts the incoming value to YYYY-MM-DD HH:MM:SS format.	numeric	Displays the number in comma separated format. For example, 12345 is displayed as 12,345.	image	Treats the contents of this column as image.	dropdown	Creates a drop-down list based on space separated values.	datecfg	Converts the incoming value to YYYY-MM-DD HH:MM:SS format. This value is again converted to gmt or local zone
Value	Description														
text	Displays the text. This is the default.														
date	Converts the incoming value to YYYY-MM-DD HH:MM:SS format.														
numeric	Displays the number in comma separated format. For example, 12345 is displayed as 12,345.														
image	Treats the contents of this column as image.														
dropdown	Creates a drop-down list based on space separated values.														
datecfg	Converts the incoming value to YYYY-MM-DD HH:MM:SS format. This value is again converted to gmt or local zone														

Parameter	Description																
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td></td><td>based on the configuration in the <code>rrs.cfg</code>.</td></tr> <tr> <td>megabyte</td><td>Displays the number as the nearest MB.</td></tr> <tr> <td>text_bold</td><td>Displays the number as the nearest MB.</td></tr> <tr> <td>count_success</td><td>Displays the values greater than 0 as green.</td></tr> <tr> <td>count_failure</td><td>Displays the values greater than 0 as red.</td></tr> <tr> <td>Pieimagepct</td><td>Displays the contents as a percentage in pie-chart.</td></tr> <tr> <td>Barimagepct</td><td>Displays the contents as percentage in horizontal bar graph.</td></tr> </table>	Value	Description		based on the configuration in the <code>rrs.cfg</code> .	megabyte	Displays the number as the nearest MB.	text_bold	Displays the number as the nearest MB.	count_success	Displays the values greater than 0 as green.	count_failure	Displays the values greater than 0 as red.	Pieimagepct	Displays the contents as a percentage in pie-chart.	Barimagepct	Displays the contents as percentage in horizontal bar graph.
Value	Description																
	based on the configuration in the <code>rrs.cfg</code> .																
megabyte	Displays the number as the nearest MB.																
text_bold	Displays the number as the nearest MB.																
count_success	Displays the values greater than 0 as green.																
count_failure	Displays the values greater than 0 as red.																
Pieimagepct	Displays the contents as a percentage in pie-chart.																
Barimagepct	Displays the contents as percentage in horizontal bar graph.																
<code><fieldsetting></code>	Manipulate output of data.																
<code><columnname></code>	Friendly name to display on report. If column name is blank, no column will be displayed for that data.																
<code><columnsetting></code>	Manipulate entire column.																
<code><url></code>	Hyperlink value. Can be an internal reporting function or an external url. For example, <code>http://[index/ \$row 2]:5800</code> , will add a link using the value from row 2 from the SQL query as a symbolic.																
<code><urlalt></code>	Mouse-over text.																
<code><urltarget></code>	Determines whether a new window is opened. Set to <code>_new</code> to open a new window. For example, clicking the Remote Control (VNC) icon opens a new window.																

Window Object Graph Section

The graph section determines the settings for the graphical representation of your data. This includes two sub-sections, one for incorporating functions (Window Toolbar Functions) and another for manipulating the graphical representation of the data (Chart Objects For Graphical Mode). A sample `<graph>` section follows:

```
<graph>
```

```
<!-- Window Toolbar Functions -->

<windowfunctions>

  <function>

    <name>ViewMode</name>

  </function>

</windowfunctions>


<!-- Chart Objects for Graphical Mode -->

<charts>

  <chart>

<queryfile>RUMCoreProductSummary_Graph_UsageTotals</queryfile>

    <queryfield>devicename</queryfield>

    <charttype>chart_drawBarChartStacked</charttype>

    <arrangeorder>reverse</arrangeorder>

    <chartwidth>700</chartwidth>

    <charheight>400</charheight>

    <charttitle>Usage Totals</charttitle>

    <charturl></charturl>

    <params>

    </params>

  </chart>

</charts>

</graph>
```

In the above code, `<arrangeorder>reverse</arrangeorder>` after `<charttype>chart_drawBarChartStacked</charttype>` generates the graph in reverse order.

Generating Reports in CSV Format

The Sentient installation media includes the `CacheUtility` script that enables you to generate reports in CSV format. This script is stored at the location

`<InstallDir>\ReportingServer\tools`. The reports generated in the CSV format are similar to the reports that are available in the Sentient Core Console. Note that using the `CacheUtility` script, you cannot merge two reports or add columns to the existing reports.

Complete the following steps to generate the reports in CSV format:

1. Using a text editor, create a new file *report_name.cfg* with the list of reports to be generated and save this file at the location `<InstallDir>\ReportingServer\tools\cacheutility`. The report names should be in window file name format. For more information on how to add report names to this file, see ["Report Name and Window File Name Mapping" below](#).
2. Using a text editor, create a new file *filter_apply.cfg* with the list of filters that are applied to each report and save this file at the location `<InstallDir>\ReportingServer\tools\cacheutility`. For more information on how to add filters to this file, see ["Report Filters" on page 1351](#).
3. Copy the *nvdkit.exe* file from the `<InstallDir>\ReportingServer\bin` directory to the `<InstallDir>\ReportingServer\tools\cacheutility` directory.
4. Open the command prompt and navigate to the `<InstallDir>\ReportingServer\tools\cacheutility` directory.
5. Enter the following command:

```
Nvdkit.exe CacheUtility.tcl -reportingPath  
"InstallDir/ReportingServer" -reportingurl  
"http://localhost:3466/reportingserver" -outputpath "output_folder"  
-reportsets "report_name.cfg" -filtersets "filter_apply.cfg" -  
deletecache "YES" -deleteoutputpath "YES" -userid "console_userID"  
-password "console_password"
```

where,

output_folder is the location where you want to store the new CSV format reports.

report_name.cfg contains the list of window file names for the reports that you want to generate.

filter_apply.cfg contains the list of filters that are applied to each report that you specify in the *report_name.cfg* file.

console_userID is the user ID to logon to the Core Console.

console_password is the password for the *console_userID* variable.

Report Name and Window File Name Mapping

The report names in the *CacheUtility* script are referenced using window file names.

For example, to generate a report of all managed devices, add the following line of code in the *report_name.cfg* file:

```
"RAMDevices.window"
```

Note that the window file names are case-sensitive and each window file name must be separated by a line feed.

The following table lists the mapping between the report name and window file name that you must set in the *report_name.cfg* file when using the *CacheUtility* script for generating reports in CSV format.

Report Name and Window Mapping

Report Name	Window File Name
Manage Devices	RAMDevices.window
Count by Memory	RAMCountByMemory.window
Count by Operating System	RAMCountByOS.window
Device Summary	RAMDeviceDetailSummary.window
MSI Package Event Details	RAMDeviceMSIServices.window
Preloaded Services	RAMDevicePreloadServices.window
Count by Memory	RAMCountByMemory.window
Native Package Event Details	RAMDeviceRNPServices.window
Managed Devices	RAMDevices.window
Managed Services	RAMDeviceServices.window
Managed Services (Historical)	RAMDeviceServicesHistorical.window
Historical Connects	RAMDevicesHistorical.window
Managed Devices - Connection Statistics	RAMOperationalDeviceConnections.window
Managed Services - Event Statistics	RAMOperationalServices.window
Managed Services (Details)	RAMServiceDetailSummary.window
Managed Services (Failed Details)	RAMServiceDetailSummaryFailed.window
Managed Servicesadmi	RAMServicesPrimary.window
Managed Services (Failures)	RAMServicesPrimaryFailed.window
Managed Blades	RIMBlades.window
CD/DVD Drives (With Device Name)	RIMCIMDeviceCDROMS.window
Physical Drives (With Device Name)	RIMCIMDeviceDiskDrives.window

Report Name	Window File Name
HPUX Software Bundles (With Device Name)	RIMCIMDeviceHPUXBundles.window
Logical Drives (With Device Name)	RIMCIMDeviceLogicalDisks.window
Network Adapters (With Device Name)	RIMCIMDeviceNetworkAdapters.window
Processors (With Device Name)	RIMCIMDeviceProcessors.window
Software Elements (With Device Name)	RIMCIMDeviceSoftwareElement.window
Unix® Local File System (With Device Name)	RIMCIMDeviceUnixLocalFileSystem.window
Count by CPU	RIMCountByCPU.window
Executable File Scan (With Device Name)	RIMDeviceFileAudit.window
Managed Software (With Device Name)	RIMDeviceManagedSoftware.window
Managed Devices (Baseboard ID) (With Device Name)	RIMDevicesBaseBoardID.window
Blades by Enclosures (With Device Name)	RIMDevicesbyEnclosures.window
Blades by Racks (With Device Name)	RIMDevicesbyRacks.window
Managed Devices (CPU/Memory) (With Device Name)	RIMDevicesCPUMEM.window
RCA Hardware Alerts (With Device Name)	RIMDevicesHPBiosEvent.window
RCA Hardware Alerts (Boot Events)	RIMDevicesHPBiosSensor.window
RCA BIOS Settings (With Device Name)	RIMDevicesHPBiosSettings.window
Logical Drives	RIMDevicesLogicalDisks.window

Report Name	Window File Name
(With Device Name)	
Managed Devices (Battery Information)	RIMDevicesPortableBattery.window
Managed Devices (Serial/BIOS)	RIMDevicesSERIALBIOS.window
S.M.A.R.T. Drives Predicting Failure	RIMDevicesSMART.window
{S.M.A.R.T. Attributes Exceeding (Below) Threshold	RIMDevicesSMARTAttributeData.window
TPM Chipset Information (With Device Name)	RIMDevicesTPM.window
Managed Devices (Vendor/Model) (With Device Name)	RIMDevicesVENDORMODEL.window
Discovered Software by Application - by Device	RIMFileAuditApplicationDetail.window
Discovered Software by Application Version	RIMFileAuditApplicationDiscovered.window
Managed Software by Application	RIMFileAuditApplicationManaged.window
Managed Software by Application - by Device	RIMFileAuditApplicationManagedDetail.window
Discovered Software by Application Version - by Device	RIMFileAuditApplicationVersionDetail.window
Discovered Software by Application Version	RIMFileAuditApplicationVersionDiscovered.window
Managed Software by Application Version	RIMFileAuditApplicationVersionManaged.window
Discovered Software	RIMFileAuditApplicationVersionManagedDetail.window

Report Name	Window File Name
by Application Version - by Device	
Discovered Software File Details	RIMFileAuditFileDetail.window
Discovered Software by Product - by Device	RIMFileAuditProductDetail.window
Discovered Software by Product	RIMFileAuditProductDiscovered.window
Managed Software by Product	RIMFileAuditProductManaged.window
Managed Software by Product - by Device	RIMFileAuditProductManagedDetail.window
Discovered Software by Product Version - by Device	RIMFileAuditProductVersionDetail.window
Discovered Software by Product Version	RIMFileAuditProductVersionDiscovered.window
Managed Software by Product Version	RIMFileAuditProductVersionManaged.window
Managed Software by Product Version - by Device	RIMFileAuditProductVersionManagedDetail.window
Discovered Software by Vendor - by Device	RIMFileAuditVendorDetail.window
Discovered Software by Vendor	RIMFileAuditVendorDiscovered.window
Managed Software by Vendor	RIMFileAuditVendorManaged.window
Managed Software by Vendor - by Device	RIMFileAuditVendorManagedDetail.window
RCA BIOS Settings	RIMHPBIOSEvent.window
RCA Hardware Alerts	RIMHPBIOSSensor.window

Report Name	Window File Name
(Boot Events)	
RCA Hardware Alerts	RIMHPBIOSSettings.window
Group Accounts	RIMNVDDDeviceGroupAccounts.window
Registered Application Scan	RIMNVDDDeviceInstalledApps.window
Add/Remove Programs Scan	RIMNVDDDeviceInstalledUninstall.window
NIS Group Accounts	RIMNVDDDeviceNISGroupAccounts.window
NIS User Accounts	RIMNVDDDeviceNISUserAccounts.window
User Accounts	RIMNVDDDeviceUserAccounts.window
Power Assistant	RIMPowerAssistant.window
Windows Printers	RIMPrintersPrimary.window
Windows Vista Readiness Report	RIMReadinessVista.window
Windows 7 Readiness Report	RIMReadinessWindows7.window
CD/DVD Drives	RIMWBEMDeviceCDROMS.window
Physical Drives	RIMWBEMDeviceDiskDrives.window
Graphics Adapters}	RIMWBEMDeviceGraphicsAdapters.window
Logical Drives	RIMWBEMDeviceLogicalDisks.window
Monitors	RIMWBEMDeviceMonitors.window
Network Adapters	RIMWBEMDeviceNetworkAdapters.window
Physical Memory	RIMWBEMDevicePhysicalMemory.window
Printers	RIMWBEMDevicePrinters.window
Windows Processes	RIMWBEMDeviceProcesses.window
Processors	RIMWBEMDeviceProcessors.window
Windows System Services	RIMWBEMDeviceServices.window
User Accounts	RIMWBEMDeviceUserAccounts.window
Windows Experience Index Report	RIMWindowsExperienceIndex.window

Report Name	Window File Name
User Accounts	RIMWBEMDeviceUserAccounts.window
Patch Device Status	RPMCompliancebyDevices.window
Managed Services	RAMDeviceServices.window
Bulletin Status	RPMCompliancebyBulletins.window

Report Filters

You must set the filter that you want to apply to the reports in the file *filter_apply.cfg*.

For example, to create a filter that filters for device name starting from WIN2k3-, add the following line of code in the *filter_apply.cfg* file:

```
"RAM Device Name.filter" "WIN2k3-"
```

The following table lists the filters that you can manually set in the *filter_apply.cfg* file when using the *CacheUtility* script for generating reports in CSV format.

Report Filters List

Filter	Filter Description
RAM Client Version.filter	{RCA Agent Version}
RAM Device CPU Vendor.filter	{CPU Vendor}
RAM Device ID.filter	{Device ID}
RAM Device Last Connect GT.filter	{Device Last Connect >}
RAM Device Last Connect Hour.filter	{Connected in Last X Hours}
RAM Device Last Connect LT.filter	{Device Last Connect <}
RAM Device Last Connect.filter	{Device Last Connect <}
RAM Device Memory Less.filter	{Memory (MB) Less Than}
RAM Device Memory More.filter	{Memory (MB) More Than}
RAM Device Memory Range.filter	{Memory Range}
RAM Device Memory.filter	{Memory (MB) }
RAM Device Name.filter	{Device Name}
RAM Device Not Connect 30 Days.filter	{Device Last Connect > 30 Days}
RAM Device Not Last Connect Hour.filter	{Not Connected in Last X Hours}
RAM Drive Size.filter	{Drive C: Minimum Size (GB) }

Filter	Filter Description
RAM Drive Space Free.filter	{Drive C: Space Free (GB)}
RAM IP Address.filter	{IP Address}
RAM Last User.filter	{Last Logged on User}
RAM MAC Address.filter	{MAC Address}
RAM Operating System.filter	{Operating System}
RAM Operational Last Connected12Months.filter	{Last Connected by Month}
RAM Operational Last Connected30Days.filter	{Last Connected by Day}
RAM Operational Total Connections12Months.filter	{Total Connections by Month}
RAM Operational Total Connections24Hours.filter	{Total Connections within 24 Hours}
RAM Operational Total Connections30Days.filter	{Total Connections by Day}
RAM OS Level.filter	{Operating System Level}
RAM RCA ID.filter	{RCA Agent ID}
RAM Service Event Failed.filter	{Service Event Failed}
RAM Service Event Successful.filter	{Service Event Successful}
RAM Service Event.filter	{Service Event}
RAM Service Failed Event.filter	{Service Failed Event}
RAM Service ID.filter	{Service ID}
RAM Service Installed Failed.filter	{Service Install Failed}
RAM Service Installed.filter	{Service Installed}
RAM Service Not Subscribed.filter	{Service Not Subscribed}
RAM Service Other Failed.filter	{Service Other Failed}
RAM Service Other.filter	{Service Other}
RAM Service Repaired Failed.filter	{Service Repair Failed}
RAM Service Repaired.filter	{Service Repaired}
RAM Service Subscribed.filter	{Service Subscribed}

Filter	Filter Description
RAM Service Uninstalled Failed.filter	{Service Uninstall Failed}
RAM Service Uninstalled.filter	{Service Uninstalled}
RAM Service Updated Failed.filter	{Service Update Failed}
RAM Service Updated.filter	{Service Updated}
RAM Service Verified Failed.filter	{Service Verify Failed}
RAM Service Verified.filter	{Service Verified}
RAM Subnet Address.filter	{Subnet Address}
RAM WF Service Domain.filter	{Service Domain}
RAM WF Service ID.filter	{Service Name}
RIM Device BaseBoardID.filter	{Device Baseboard ID}
RIM Device BIOS Version.filter	{Device BIOS Version}
RIM Device Class.filter	{Device Classification}
RIM Device CPU Count.filter	{CPU Count}
RIM Device CPU Speed Less.filter	{CPU Speed (MHz) Less Than}
RIM Device CPU Speed More.filter	{CPU Speed (MHz) More Than}
RIM Device CPU Speed Range.filter	{CPU Speed Range}
RIM Device CPU Speed.filter	{CPU Speed (MHz)}
RIM Device Model.filter	{Device Model}
RIM Device Serial Number.filter	{Device Serial Number}
RIM Device Vendor.filter	{Device Vendor}
RIM FileAudit Application Name Managed.filter	{Application Name Managed}
RIM FileAudit Application Name Unmanaged.filter	{Application Name Unmanaged}
RIM FileAudit Application Name.filter	{Application Name}
RIM FileAudit Product Name Managed.filter	{Product Name Managed}
RIM FileAudit Product Name Unmanaged.filter	{Product Name Unmanaged}
RIM FileAudit Product Name.filter	{Product Name}

Filter	Filter Description
RIM FileAudit Product Version.filter	{Product Version}
RIM FileAudit Vendor Name Managed.filter	{Vendor Name Managed}
RIM FileAudit Vendor Name Unmanaged.filter	{Vendor Name Unmanaged}
RIM FileAudit Vendor Name.filter	{Vendor Name}
RIM Network Adapter.filter	{Network Adapter}
RIM Power Date GT.filter	{Scan Date >}
RIM Power Date LT.filter	{Scan Date <}
RIM Power Date.filter	{Last Scan Date}
RIM WBEM Graphics Adapter.filter	{Graphics Adapter}
RIM WBEM Printer Driver Name.filter	{Printer Driver Name}
RIM WBEM Process Name.filter	{Windows Process Name}
RIM WBEM Service Name.filter	{Windows Service Name}
RIM WF FileAudit Application Name.filter	{Application Name}
RIM WF FileAudit Application Version.filter	{Application Version}
RIM WF FileAudit Managed Software.filter	{Managed Software}
RIM WF FileAudit Product Name.filter	{Product Name}
RIM WF FileAudit Product Version.filter	{Product Version}
RIM WF FileAudit Unmanaged Software.filter	{Unmanaged Software}
RIM WF FileAudit Vendor Name.filter	{Vendor Name}
RIM Windows Experience Index.filter	{Windows Experience Index Overall Score}
RIM_WF_HPBIOSensor_Name.filter	{Hardware Alert Name}
RIM_WF_HPBIOSensor_State.filter	{Hardware Alert State}
RIM_WF_HPBIOSetting_Name.filter	{BIOS Setting Description}
RIM_WF_HPBIOSetting_Value.filter	{BIOS Setting Value}
RIM_WF_SMARTAttributeData_ID.filter	{S.M.A.R.T. Attribute ID}

Filter	Filter Description
RPM Device Name.filter	{Device Name}
RPM WF Bulletin Name.filter	{Bulletin Name}

Scheduling CSV Format Reports

You can create a Job Action Template that generates the reports in CSV format, and then schedule a job based on this template.

Prior to scheduling a job, make sure that the Core agent is installed on the Sentient Core server. The DTM job is then run for this Core agent.

While creating the template, enter the following value in the Command text box under Action Parameters view in the Job Action Template Parameters page of the Job Action Template Creation wizard:

```
Nvdkit.exe ..\ReportingServer\tools\cacheutility\CacheUtility.tcl -
reportingPath "InstallDir\ReportingServer" -reportingurl
"http://localhost:3466/reportingserver" -outputpath "output_folder" -
reportsets "..\ReportingServer\tools\cacheutility\report_name.cfg" -
filtersets "..\ReportingServer\tools\cacheutility\filter_apply.cfg" -
deletocache "YES" -deleteoutputpath "YES" -userid "console_userID" -
password "console_password"
```

where,

report_name.cfg contains the list of window file names for the reports that you want to generate.

filter_apply.cfg contains the list of filters that are applied to each report that you specify in the *report_name.cfg* file.

output_folder is the location where you want to store the reports.

console_userID is the user ID to logon to the Core Console.

console_password is the password for the *console_userID* variable.

For instructions on how to create a Job Action Template, see the *Sentient Enterprise Console Online Help*.

Exporting Large Reports to CSV Format

To export reports with large number of records, such as more than one million, to CSV format, perform the following steps:

1. Navigate to <InstallDir>\ReportingServer\etc and open *rrs.cfg* in a text editor.
2. Add a parameter - *QUICKMODELIST* - under *Core* configuration if it does not exist.
3. Add .window file names under *QUICKMODELIST* as shown in the example below:

```
-QUICKMODELIST { "RIMFileAuditApplicationDiscovered"
"RIMFileAuditApplicationVersionDiscovered"
"RIMFileAuditProductDiscovered"
"RIMFileAuditProductVersionDiscovered"
"RIMFileAuditVendorDiscovered" }
```

As shown in the example, if you need to add multiple file names to this parameter, add them in a space separated list.

4. Change value of the `EXPORTMODE` parameter to `comma`.
5. Save the file.
6. Open Sentient Console and navigate to the report in which you have made changes and click the icon for **Export to CSV**.

When you open corresponding view, maximum 500,000 rows are displayed on the screen. However when you export the report, complete set of records is exported in CSV format.

Chapter 14

RCA Backend Services Management

Portal Services

The Portal is a backend component of any Sentient environment that provides an engine, Web Services, and an OpenLDAP database for the devices being managed in your environment. The Portal allows you to perform administrative and operational tasks from the Sentient Console on any piece of your Client Automation infrastructure. The capabilities of the Portal include:

- **Authentication**
Sentient Console users are authenticated against the Portal directory.
- **Device Categories**
The Portal captures detailed information about device hardware, operating system, Client Automation infrastructure and managed services and stores it in the Portal Directory in self-managed device categories. This simplifies notification of all devices for a given classification in a single step.
- **OpenLDAP Directory for Device and Group Tasks**
From the Sentient Console, many activities, such as Notify, are performed on the target device groups that you select. The Portal provides the OpenLDAP directory hosting the device and device groups. The RCA Portal OpenLDAP directory is referred to and managed by the Core as the Sentient Enterprise Directory Service.
- **Agent Deployment**
The Portal facilitates the remote deployments of Sentient agent to Windows, Linux, and Macintosh systems.
- **OS Deployment**
The Portal database stores the OS Management policies in your Sentient environment.
- The Portal service enables you to assign policies and manage device objects.

Portal Architecture

The Portal contains the following:

- The **Portal Run-time** contains the RCA Portal service (RCA-RMP) and the `RMP.TKD` module (located in the `\modules` directory).
- The **Portal Zone Directory**, is an OpenLDAP directory service in the `RCA\DirectoryService` directory. When the Portal starts, it loads the database objects that represent a given instance of the Portal, or Zone. The database objects include all information needed to manage a given set of infrastructure at a given location:
 - Managed devices
 - Managed Services Catalog

- Directory Services Definition
- Device group memberships
- Chassis container for blade enclosures and racks
- Device Categories
- Job Status and Job History
- Users
- Configurations for Entitlements, Tasks, and Services
- Networks
- The **Management Agent**, installed on the remote devices when the Sentient Agents are installed, performs tasks on behalf of the Portal.

Portal Zone

A zone is a logical set of devices, infrastructure, and software that is represented and managed in directory services and administered by the Portal through the Sentient Console.

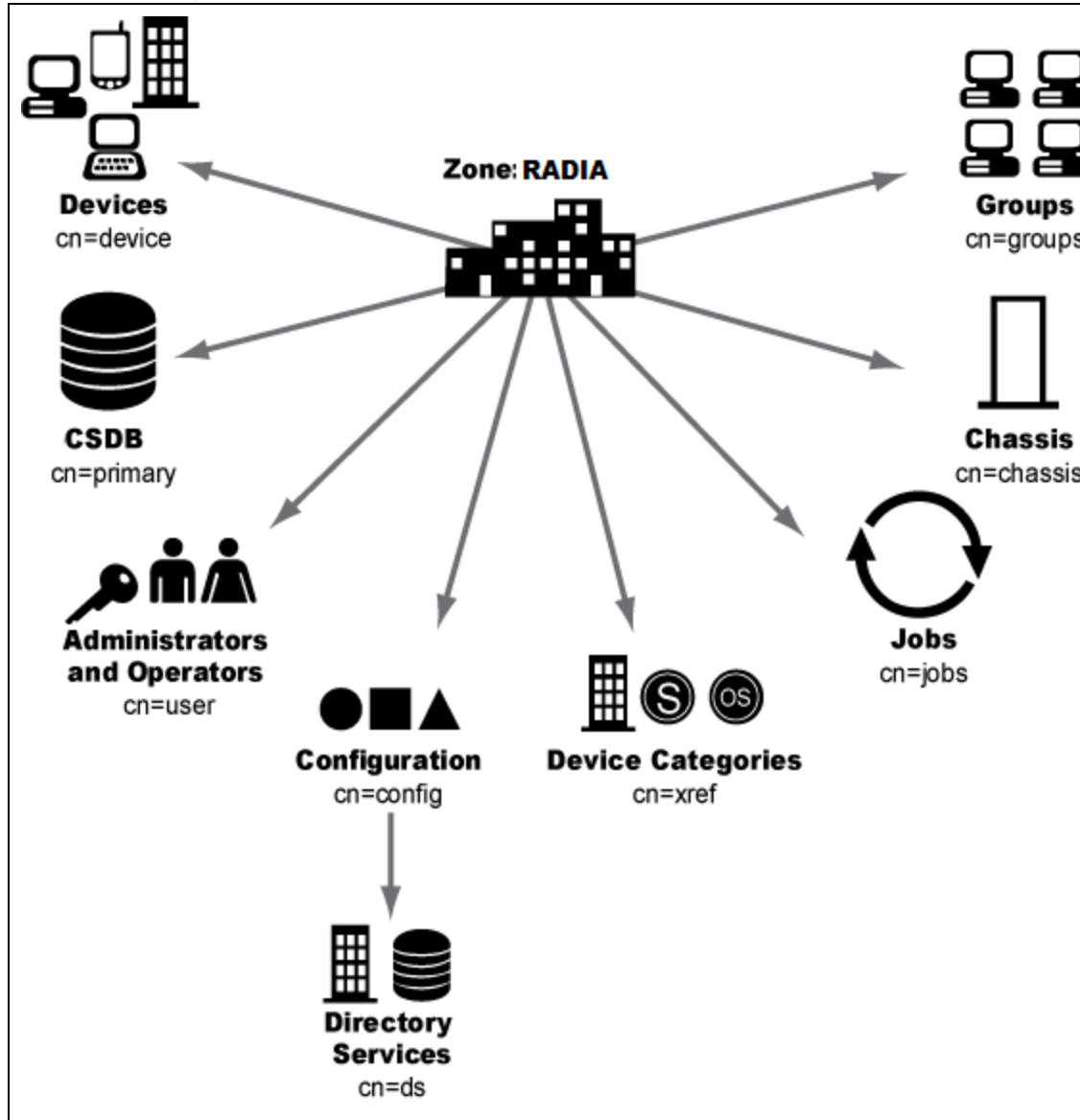
A zone is created whenever the Portal is installed (as part of the Sentient installation), and all objects in the zone include the high-level qualifier of the zone name. The properties for the zone object, itself, include the URL information needed to access the zone.

The Zone Directory Structure

Every Portal zone has the same directory structure and same-named containers at the highest levels.

The next figure illustrates the zone directory structure and containers. See the *Sentient Enterprise Console Online Help* for a description of each container and how they are used.

Portal directory of a zone



About Object Names in a Zone

The Portal, itself, is a directory service containing objects of various object classes. Each object is assigned a common name (**cn=name**). The common name given to an object must be unique among all objects in that class. For example, all zone names in your enterprise must be unique. Within a given zone, all common names of objects of the same class must be unique. The common names of the zone containers are pre-assigned and the same across all zones in your enterprise.

Each entry within a zone may be identified by its location. For example, the location of the **Devices** container entry in the figure above is **cn=device, cn=HP, cn=radia** and the location of the PRIMARY File on the Configuration Server is **cn=Primary, cn=HP, cn=radia**.

Administrative Functions

Setting Configuration Parameters

Separate topics discuss how to modify the `rmp.cfg` file for:

- Customizing Domain Filters for Policy Resolution (see page ["Customizing Domain Filters \(DNAMES\) for Policy Resolution" on page 1363](#)).
- Managing Management Agent Signal Processing (see page ["Managing Management Agent Signal Processing" on page 1364](#))

The following table lists the parameters you can add to or modify in the `rmp.cfg` file for options that are not related to any of the topics listed above.

Additional Portal Configuration Parameters in RMP.CFG

Parameter	Definition
LINKS	Specifies the policy configuration links to enable when policy has been applied to the objects in the Chassis container and related Device Categories containers for server blade devices.
LISTENING_ADDRESS	Specifies a valid network address (either an IP address, hostname, or DNS address) that is to be passed to Management Agents, and then used by them to connect back to the Portal. Use a LISTENING_ADDRESS when the Management Agents are experiencing communication failures with the Portal and are unsuccessful in registering back to the Portal or performing remote tasks on behalf of the Portal. This can occur when the Portal resides on a machine with dual-NIC cards or is using a dynamic IP address. Specify a network address using the format that works best in your environment: <pre>LISTENING_ADDRESS IPaddress</pre> or <pre>LISTENING_ADDRESS hostname</pre> or <pre>LISTENING_ADDRESS DNS</pre> Note: Ensure the network address you enter points to the current Portal Zone. If it does not, results are unpredictable.
NETSCAN	This parameter controls the device discovery as it

Parameter	Definition
	enables you to discover OOB devices on your network. Specify <code>NETSCAN</code> as 0, to disable device discovery. Device discovery can be disabled if there is excessive network traffic on large networks. By default device discovery is enabled.
<code>RCS_AUTO_CONNECT</code>	When a Primary Configuration Server directory service is defined for the Portal, controls an automatic connection to the Primary Configuration Server whenever the Portal is started and the <code>ds-rcs Startup</code> property is set to Auto or Manual. The <code>RCS_AUTO_CONNECT</code> is not enforced when Startup is set to Disabled. Default value is 1 (enabled). Enter <code>RCS_AUTO_CONNECT 0</code> to disable the automatic connection to the Primary Configuration Server; and revert to the connections as defined by the Startup property when the Directory Service was configured.
<code>REFRESHMSC</code>	When a Primary Configuration Server directory service is defined for the Portal, controls how often the Portal updates its <i>Managed Services Catalog</i> with those available in the source Configuration Server database. The Managed Services Catalog serves as the RCA Definitive Software Library, and is accessible from the Services object (<code>cn=services</code>) located in the root of the Portal directory. Default value is 600 seconds, or 10 minutes. Specify a different interval for the refresh of the Managed Services Catalog in seconds.
<code>USE_FQDNHOST_NAME</code>	Specifies that Portal should contact remote hosts using either fully qualified domain names or short names (that is, the left-most portion of a fully qualified domain name). Customers whose DNS tables contain imperfect entries may want to switch to the use of short names. Sample operations that involve contacting a remote host include a Notify, a Proxy preload or purge, stopping or starting services via the Management Agent, and contacting the Management Agent. - Type <code>USE_FQDNHOST_NAME 0</code> to use short names (that is, the left-most portion of a fully qualified name). Customers whose DNS tables contain imperfect entries may want to switch to the use of short names. - Type <code>USE_FQDNHOST_NAME 1</code> to return to the use of fully qualified domain names (the default).

Parameter	Definition
WOL_MCAST_ADDR	Permits Wake-on-LAN (WOL) support in multicast-enabled environments. Default is no support for multicast WOL. - Type <code>WOL_MCAST_ADDR <IP_address></code> where the <code><IP_address></code> specifies the multicast address to use to revolve a WOL request. - Type <code>WOL_MCAST_ADDR 0</code> to return to standard WOL support (no multicast WOL support). This is the default.
AUTH_LEVEL	Enables faster Sentient users authentication when you integrate an external directory service with Sentient. If <code>AUTH_LEVEL</code> is set to 1, Sentient authenticates the users that are direct members of the Authentication Group DN you configured in the Sentient Console. Sentient does not authenticate any sub group members in the Authentication Group DN. Hence, when you configure <code>AUTH_LEVEL</code>, you must make sure that all Sentient users are added as direct members of the Authentication Group DN.
DISABLE_MACADDR_MATCHING	When multiple devices share a NIC card (for example USB NIC), or use the same docking station at different times, the devices appear as having same Mac address. This may cause confusion and result in one device object under Zone:Sentient for multiple devices. In such a situation, Mac address should be ignored by setting this parameter to 1.

Configuring for a Custom LDAP Policy Extension Prefix

Many Policy Server implementations use the default LDAP Policy Extension prefix of `edm`—as in `edmPolicy`. If you have defined an LDAP Directory Service for policy tasks, but it uses a policy extension prefix other than `edm`, use the following procedure to define its LDAP Policy Extension prefix value to the Portal. This procedure adds a `PREFIX` parameter to the `rmp.cfg` file where you specify a policy prefix value other than `edm`.

To configure the Portal for a Custom LDAP Policy Prefix (other than `edm`):

1. Stop the RCA Portal service (`RCA-RMP`).
2. Use a text editor to open the Portal configuration file, `rmp.cfg`, located by default in `<InstallDir>\ManagementPortal\etc`.
3. Insert the `PREFIX` parameter (must be uppercase) into this file before the finishing curly bracket (`}`) as shown in the code sample here.

```
#
rmp::init {
    URL/
```

```
PREFIX    rad

    }
#
# END OF CONFIG
#
```

4. Use one or more spaces to separate the PREFIX parameter and its value. Specify the value using the same case as is entered for the LDAP Policy Extension prefix defined in the Policy Server.

Parameter to Configure a Custom Policy Prefix

Parameter	Explanation
PREFIX	Defines an LDAP Policy Extension prefix other than the default value of edm. Enter one or more spaces to separate the PREFIX parameter and its value. The value must match the LDAP Policy Extension prefix defined in the Policy Server. For example: <code>PREFIX rad</code> defines a policy prefix of rad instead of edm.

5. Save and close the file.
6. Restart the RCA Portal service (RCA-RMP) and open the Portal.

Customizing Domain Filters (DNAMES) for Policy Resolution

If you have modified the domain filter settings defined in your Policy Server `pm.cfg` file, you can port your modified filter settings to the Portal. The modified filter settings will be available from the Dname drop-down list box on the Resolve Policy task page.

Domain filtering is defined in your Policy Server. Any custom filter settings must be properly defined in the Policy Server configuration file, `pm.cfg` using the format:

```
DNAME=<DOMAIN NAME>    { rule }
```

Note: For more information on domain filtering and syntax, see ["Domain Filtering" on page 1308](#).

To port your custom domain filter settings to the Portal Resolve Policy task you must modify the `RCA-RMP.rc` file, which is located in the `etc` directory of where the Portal is installed. Add the following custom code to the end of the `RCA-RMP.rc` file using the format:

```
namespace eval policy {

default cfg(DNAME=<DOMAIN NAME>)    { rule }

}
```

where `DNAME=<DOMAIN NAME>` and `{ rule }` correspond to a custom filter setting in your `pm.cfg` file. The code sample below displays the end of the `RCA-RMP.rc` file configured for

custom policy filters. This example shows a modified definition for the default (*) filter as well as a new AUDIT filter.

```
namespace eval policy {  
  
  default cfg(DNAME=*)          { * !PATCHMGR !OS !AUDIT}  
  
  default cfg(DNAME=PATCH)     { PATCHMGR }  
  
  default cfg(DNAME=OS)         { OS }  
  
  default cfg(DNAME=AUDIT)      { AUDIT }  
  
}
```

Save the changes to the `RCA-RMP.rc` file and restart the Portal service.

Managing the Portal Zone Directory

The Portal Zone Directory is an OpenLDAP directory.

- To backup and restore the Portal Zone OpenLDAP directory, see the Admin Guide available at <http://www.openldap.org/>.
Use standard tools available from <http://www.openldap.org/> to backup and restore this directory.
- For information on Portal Directory Troubleshooting and logging the Slapd service, see the *Sentient Enterprise Troubleshooting Guide*.

Terms for Database Recovery

slapd - The stand-alone OpenLDAP daemon. It is started when the Sentient Client Automation Directory Service Windows service is started on the Core.

For more information on the use of this service with an OpenLDAP directory, see <http://www.openldap.org/>. Slapd is discussed on this page:
<http://www.openldap.org/doc/admin24/intro.html>.

Managing Management Agent Signal Processing

The Portal uses three types of dedicated thread pools to handle the incoming requests from Management Agents. You can adjust the number of threads assigned to each pool by adding or updating these parameters in `rpm.cfg`.

You can also adjust the maximum number of Management Agent signals the Portal will process at a time.

The parameter changes you make in the `rpm.cfg` file will take affect when the Portal is restarted.

- The Portal limits the number of Management Agent signals it will accept for concurrent processing. This is defined in the `OPEN_RMA_SIGNAL_SOCKETS_MAX` parameter.

- All incoming Management Agent requests are handled initially by the `RMA_SIGNAL_RECEIVER_THREADS` pool. These lightweight threads handle only the simplest tasks, such as Management Agent status checks when the device DN is known.
- Management Agent requests requiring a database update for a known DN are passed to the `RMA_SIGNAL_PROCESSOR_THREADS` pool.
- Management Agent requests requiring any Management Agent registration look-up or creation work (that is, the device DN is not known) are passed to the `RMP_REGISTRATION_THREADS` pool. These threads perform the heaviest work.

The following table summarizes the default and valid values for each parameter related to Management Agent signal processing.

Management Agent Signal Processing Parameters (`rmp.cfg`)

Management Agent Signal Processing Parameters (`rmp.cfg`)

Parameter, Default, Valid Values	Definition
<code>OPEN_RMA_SIGNAL_SOCKETS_MAX</code> Default: 1024 Valid Values: 256 or greater	Maximum number of Management Agent signals concurrently being processed by the Portal. After reaching this maximum, the Portal will reject additional incoming signals.
<code>RMA_SIGNAL_RECEIVER_THREADS</code> Default: 20 Valid values: positive number	Number of lightweight threads to use to accept incoming Management Agent requests. These threads handle Management Agent status checks when the DN is known, or pass requests to appropriate Management Agent Signal Processor or Management Agent Registration pool.
<code>RMA_SIGNAL_PROCESSOR_THREADS</code> Default: 3 Valid Values: positive number	Number of threads to process database updates when the Management Agent device DN is known.
<code>RMP_REGISTRATION_THREADS</code> Default: 1 Valid Values: positive number	Number of threads to process Management Agent registration look-up or creation work when the DN is not known. If these threads have no work, they will automatically assist with any Management Agent signal processor work.

Managing the Portal Web Services Token

The Portal Web Services (WS) require a valid token holding user credentials in order to perform Sentient Enterprise activities.

The `WS_TOKEN_TTL` parameter in the `rpm.cfg` determines how long, in seconds, a given Portal Web Services (WS) user credential token is valid before it expires. During normal usage of the Sentient Core Console, the token is routinely refreshed. If no user activity is detected within the `WS_TOKEN_TTL` period, the user's session will expire and they will be asked to log in again.

The default `WS_TOKEN_TTL` period is 1200 seconds (20 minutes).

Valid Values: positive number

Default: 1200 seconds

Routing Messages to Portal

By default all internal LDAP messages to the Portal except Satellite server messages, Proxy server messages, and OS Manager server messages are disabled. The `ZTASKEND` REXX program routes the messages that are not configured to be sent to the Portal to the RDBMS using Messaging server. When the external directory service is not being used, these messages help in providing internal policy resolution if the Management agent (formerly known as RMA) is not installed on the target device.

To modify the default `ZTASKEND` behavior, such that all messages are sent to the Portal, complete the following steps:

1. Open the file `ZTASKEND` file from the location
`<InstallDir>\ConfigurationServer\rexx\NOVADIGM.`
2. Delete the following lines of code:
 - `Domname = strip( Upper(edmgetv( "PREFACE", "ZDOMNAME"))) )`
 - `If ( ( Ctype = "SATELLITE" | Ctype = "RPS" ) | (Domname = "OS" ) )`
`Then Do`
`BootCmd = BaseCmd || " -to CORE.RMP,CORE.ODBC -priority 10 "`
`BootObjects`
`End`
`Else Do`
`End`

Make sure that you do not delete the following line in the Else condition:

```
BootCmd = BaseCmd || " -to CORE.RMP,CORE.ODBC -priority 10 "
BootObjects
```

3. Save and close the `ZTASKEND` file.

Note that when you enable all messages to the portal, the load increases on the Portal, which can further result in the reduced Portal performance.

Messaging Services

The Sentient Enterprise Messaging Server (Messaging Server) is a robust messaging service that provides a means to forward data from one piece of the Sentient infrastructure to another. It can be used as a point to aggregate different types of data as well as to segregate data accumulated from various Sentient servers by type. Its job is to monitor predefined data queues and dynamically route data objects to one or more external destinations. The Messaging Server provides retry, rerouting, and failover capabilities to ensure all data is transferred efficiently and reliably.

On the Sentient Configuration Server (Configuration Server), the Messaging Server operates hand-in-hand with the executable, `QMSG`, to handle the transfer of data reported from Sentient agents to the appropriate ODBC reporting database or external Sentient Server.

Features and Benefits

The Messaging Server provides an efficient and flexible messaging system. It can:

- Route a single message to multiple destinations.
- Automatically retry a message delivery.
- Re-route messages to a new host after an unsuccessful delivery attempt (failover capability).
- Route data from one Messaging Server to another one (store and forward capability).
- Maintain multiple data queues on Store and Forward Messaging Servers.

The Messaging Server provides the following benefits:

- Multiple, specialized queues that enable separate workers to operate on each queue, and therefore allows for parallel processing of object messages.
- Additional workers can be configured to drain a queue more quickly.
- Independent data delivery agents allow for modular upgrades.
- Eliminates bottlenecks on the Configuration Server.
- Reliability of processing Inventory and Patch data is maintained, due to:
 - Built-in retry capability.
 - Ability to reroute messages remaining in a queue to a failover location.
 - Retry, holding, and re-routing features eliminate potential loss of data caused by network failures.
- Improved memory control related to creating the Inventory Manager SQL tables and loading the SQL commands upon startup. As a result, these SQL tasks are always performed upon Messaging Server and Data Delivery Agent startup, and support for the previous `STARTUPLOAD` parameter has been removed.
- Improved queue control and throttling capability for posting Sentient Portal data.
- New support for customized message routing to multiple DSNs using ODBC.

Messaging Server Processing on the Sentient Server

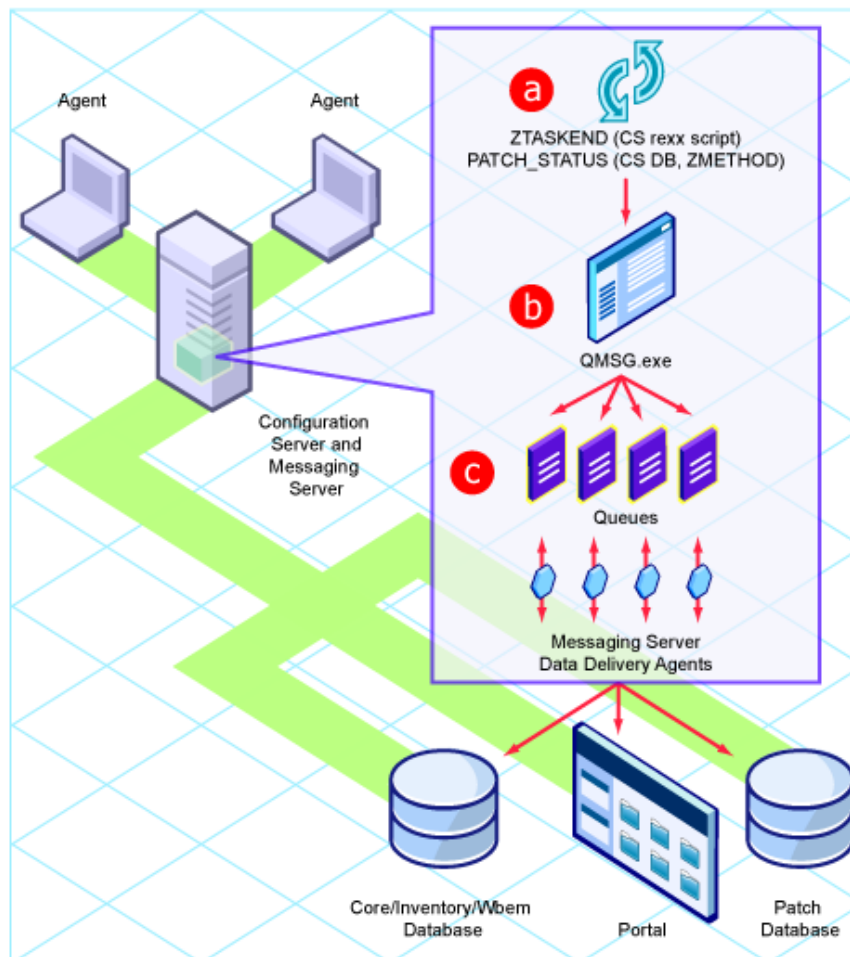
The Messaging Server acts as a delivery service between the Core Server or the Satellite server and external ODBC databases or Sentient services. When a customer has multiple Satellite servers, each one will have a co-located Messaging Server and the ability to route object data to the appropriate target location.

The figure "[Messaging Server ProcessingLegend](#)" on the next page provides an overview of Messaging Server Processing.

The Sentient agent connects to the Configuration Server to resolve its desired state. At the end of each agent connection, the agent passes objects back to the Configuration Server. Different agent objects are passed according to each specific phase of the agent connect process.

The Messaging Server refers to CORE objects as the standard Sentient agent objects that are exchanged between the agent and the Configuration Server. Examples of CORE objects are ZMASTER, PREFACE, and SESSION. Other types of objects that can be exchanged are multi-heap WBEM reporting objects, FILEPOST objects created by an inventory agent audit process (called INVENTORY objects) and the DEERROR, BUSTATUS, DESTATUS, and PASTATUS agent objects collected for Sentient Patch Manager processing.

Messaging Server Processing



Legend

a - ZTASKEND is called

b - QMSG.EXE assembles object data

c - Messaging Server Data Delivery Agents poll the queues and transfer data

ZTASKEND is Called

On the Configuration Server, the ZTASKEND rexx method is called at the end of the agent connect process. ZTASKEND creates the commands to invoke the QMSG executable. The commands to QMSG contain parameters that define the appropriate objects to send as well as the designated queues in which to place the objects. ZTASKEND is responsible for naming all objects forwarded to QMSG—with the exception of objects needed for Patch reporting. For Patch objects, five methods in the PRIMARY.SYSTEM.ZMETHOD class of the Configuration Server Database call QMSG for the PATCH objects.

QMSG.EXE Assemble Object Data

QMSG assembles object data into XML files and creates header files with the appropriate metadata “address” needed to deliver the message by the Messaging Server. When the two message files (XML and meta data files) are created, QMSG places these files in one or more predefined data queues on the Configuration Server.

Note: QMSG places objects in separate queues, based on the parameters specified in ZTASKEND and in the four PATCH_* methods (DEERROR, BUSTATUS, DESTATUS, and PASTATUS). The Messaging Server is configured to use individual Data Delivery Agents (DDAs) to process messages from these queues. The table "[Data Queues and Data Delivery Agents](#)" lists the Data Delivery Agents that operate on each data queue location.

Messaging Server Data Delivery Agents poll the queues and transfer data

The Messaging Server DDA's poll the message queues and automatically pick up and transfer data files to the appropriate external locations using the routing type and locations defined in the specific data delivery agent's configuration file.

Data Queues and Data Delivery Agents

Data Queue	Data Delivery Agent
..\Data\MessagingServer\core	core.dda
..\Data\MessagingServer\patch	patch.dda
..\Data\MessagingServer\usage	usage.dda

Data Delivery Agents

The Data Delivery Agents are function-specific modules created to transfer certain types of message data. There are Data Deliver Agents available for CORE, PATCH, and USAGE message data.

- The CORE message data refer to agent objects typically exchanged during a standard agent connect process. The CORE message data also includes the FILEPOST objects created during the agent audit process and the WBEM reporting object data. Examples of CORE type message objects include ZMASTER, SESSION, and ZCONFIG. CORE objects support reports for Vulnerability Management, Windows CE Thin Clients and Application Management Profiles, among others.
- The PATCH message data is comprised of DESTATUS, BUSTATUS, DEERROR and PASTATUS agent object data.

The Data Delivery Agents (DDAs) have the ability to post messages using ODBC into a SQL database that can be used for reporting. For processing efficiency, each DDA works independently on its own queue.

The Messaging Server loads these independent data delivery agents, whose configurations define how and where the messages for CORE, INVENTORY, WBEM, and PATCH data objects will be delivered.

Note: The CORE data delivery agent is configured to post CORE object data to an Inventory Manager database, as well as CORE object data to a Portal directory.

The data delivery agent modules are located in the \MessagingServer\modules folder. The modules are loaded using "dda.module load" statements in the Messaging Server configuration file (rms.cfg).

Each data delivery agent is configured from its own configuration file (`*.dda.cfg`). These configuration files are located in the `\MessagingServer\etc` folder.

Routing Inventory Data

This Messaging Server supports direct ODBC posting of Inventory Manager data to a back-end SQL Inventory Database. The CORE.DDA has the ability to route CORE, INVENTORY and WBEM data messages to a back-end SQL database using ODBC. It is recommended to use this routing option for best performance.

Creating SQL Tables for the Inventory Database

The CORE Data Delivery Agent posts the message data into the SQL table, for which the default definitions and associated SQL queries are found in the following Messaging Server directories:

```
/etc/<module name>/rca
```

Customized scripts can be placed in the directories:

```
/etc/<module name>/
```

The script necessary to map the CORE object data to the related SQL table column is `taskend.tcl`. The script necessary to map the INVENTORY object data (FILEPOST object) is called `filepost.tcl`. Both of these scripts are found in the `/etc/<module name>/lib` directory of the Messaging Server.

All inventory and the application deployment related tables have a foreign key constraint with the table `DeviceConfig`. This foreign key constraint ensures that when a device is deleted, all the records corresponding to that device are deleted from all the other tables.

The SQL tasks to create the tables for the Inventory Database and load the SQL commands into memory are performed when the Messaging Server service and the Data Delivery Agents are initially started.

Note: All SQL tasks are performed at service startup and any STARTUPLOAD value found in a configuration file is ignored.

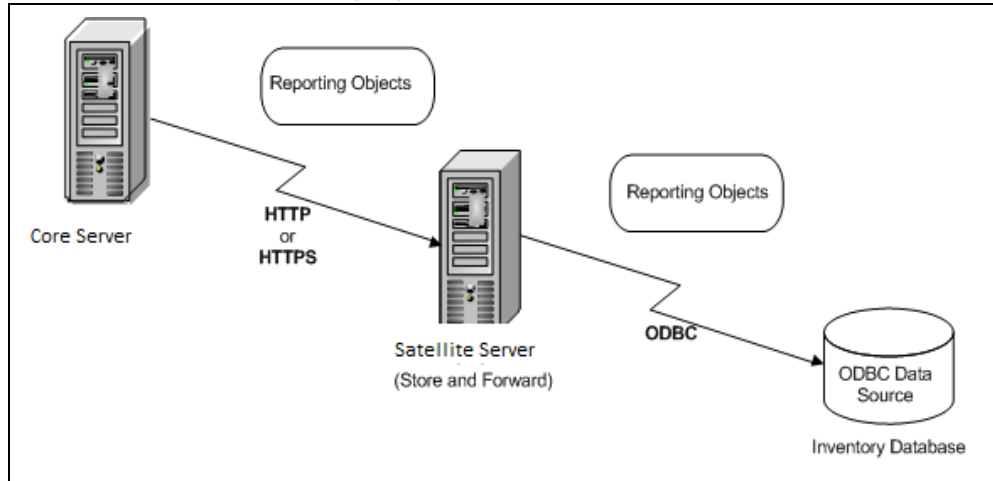
Using Store and Forward

The Messaging Server includes store and forward capabilities that allow you to forward messages to another Messaging Server using HTTP or HTTPS. In addition, if the Messaging Server is not located close to the SQL database, it is a good practice to forward the messages to one that is located as close to the SQL database as possible.

The Messaging Server supports forwarding and receiving messages using multiple queues.

The following figure Store and Forward Messaging Server illustrates a typical configuration that forwards messages to another Messaging Server, prior to posting data to the Inventory database using ODBC.

Store and Forward Messaging Server



1. The Messaging Server on the Configuration Server is configured to have the CORE data delivery agent forward the data to another Messaging Server using HTTP or HTTPS. This configuration makes use of the *coreforward*, *inventoryforward*, and *wbemforward* sections that are provided in the data delivery agent configuration files.
2. The Store and Forward Messaging Server is located close to the Inventory Manager SQL database. It receives the CORE, INVENTORY, and WBEM objects.
3. The attending CORE data delivery agent on the receiving Messaging Server posts the data objects to the Inventory Manager database using ODBC.

Configuring and Tuning Messaging Server

Messaging Server Processing

This topic explains how the methods on the Configuration Server work hand-in-hand with the Messaging Server to collect, queue, and then deliver data to the appropriate external location.

Getting Agent Information

Agent objects exchanged with or created on the Configuration Server during an agent connect session are formatted into messages for the Messaging Server via the Configuration Server binary executable QMSG. The QMSG executable is part of the Sentient media. This executable is invoked during agent taskend processing by the rexx method ZTASKEND or a connection to the Patch Manager methods. The calls to QMSG can include parameters specifying what queue to place the messages in, the priority in which the message is to be processed, the objects that are to be included in the messages, and the “destination address” or routing identifier for the file.

The QMSG executable formats the object data into an XML file. Each XML file can be made up of multiple objects, such as when processing the CORE objects (for example, ZMASTER, SESSION and ZCONFIG) or it can be a single multi-heap object such as a *wbem* or *filepost* object. Each call to QMSG will produce two files, the XML file created from the object data and a file which contains the metadata. Metadata are attributes describing the XML file, how big it is, when it was created and the routing identifier. The actual file names are created with a timestamp

format. This enables the Messaging Server to process the oldest messages first. The Messaging Server always processes in a “First In First Out” mode when the messages have the same processing priority.

When queue designations are specified when invoking QMSG, messages are placed in a directory with the specified queue name. When no queue identifier is used, messages are placed in a directory named default. Each data delivery agent uses its own unique queue for its particular messages. This segregation of messages according to type allows for the simultaneous processing of all queues and leads to more efficient operation.

In Summary:

- For agent information needed by the Patch Manager, QMSG is executed as a result of the `SYSTEM.ZMETHOD.PATCH*` instances: `PATCH_DEERROR`, `PATCH_BUSTATUS`, `PATCH_DESTATUS`, and `PATCH_PASTATUS`.
- For information needed by the Inventory Manager, Portal, and Application Management Profiles, the Configuration Server REXX method, ZTASKEND, is used to trigger the call to QMSG. The format of the QMSG call is included in the discussion of ZTASKEND which follows.

Patch Methods for Collection

Four methods call the QMSG executable with the parameters in the ZMTHPRMS attribute of the method. The default value of this attribute looks like this:

```
ZMTHPRMS      -to PATCH5 -queue patch <<object-name>>
```

The `-to PATCH5` parameter specifies that the messages will be placed in a queue called `../data/patch` relative to the location of where QMSG is executed. This is the default location specified in the install of the `patch.dda`.

The `<<object-name>>` parameters (`DEERROR`, `BUSTATUS`, `DESTATUS`, and `PASTATUS`) specify the objects that will be included in the message files created by QMSG.

ZTASKEND REXX Method

The ZTASKEND REXX method on the Configuration Server is called at the end of each agent connect, while objects associated with the present session are still available in storage on the Configuration Server. ZTASKEND invokes QMSG when agent data needs to be collected for another service, such as Inventory Manager, Application Management Profiles, or the Portal. QMSG collects the data and places the messages in specified queues for pickup and processing by the Messaging Server and its data delivery agents.

Note: Different object types (core, inventory and wbem objects) are placed in separate queues whenever the Data Delivery Agents have been installed for those data objects.

An important job of ZTASKEND is to ensure unique agent data is collected at the appropriate agent connect phase. For efficiency, ZTASKEND also groups identical messages, having the exact same object content, to minimize the number of calls made to QMSG.

This topic explains:

- How ZTASKEND determines when to call QMSG for the various agent connect phases and which objects are collected.
- The basic syntax of calls to QMSG.
- How the QMSG -to parameter establishes one or more destinations for message processing.
- How the QMSG -priority parameter establishes Messaging Server processing order.

ZTASKEND Calls to QMSG

This topic reflects the ZTASKEND method.

Processing Phase-Dependent Objects

Each time an Sentient agent connects to the Configuration Server, the agent declares its connection intent or phase. An important role of the ZTASKEND REXX code is to minimize the collection of duplicate information. The ZTASKEND method invokes QMSG depending on the agent connection phase and the objects present. ZTASKEND restricts message posting to five specific connection phases. The phases of interest are:

- BOOTSTRAP (Client Operations Profiles or COP)
- AGENT SELF MAINTENANCE
- CATALOG RESOLUTION
- SERVICE RESOLUTION
- AGENT REPORTING

Processing CORE Objects by Phase

There are "critical objects" collected for each of the core targets for Inventory Manager (RIM) and Portal (RMP). The potential critical objects are:

For RIM: APPEVENT MSIEVENT SYNOPSIS RNPEVENT

For RMP: SYNOPSIS IDENTITY

In addition to the above critical objects, the table "Critical Objects collected by phase" identifies critical objects collected for each phase.

Critical Objects collected by phase

Phase	Critical Objects
BOOTSTRAP (COP)	SESSION PREFACE ZSTATUS SMINFO
AGENT SELFMAINTENANCE	SESSION PREFACE ZSTATUS SMINFO
CATALOG RESOLUTION	SESSION PREFACE ZSTATUS ZCONFIG ZMASTER SMINFO
SERVICE RESOLUTION	SESSION PREFACE ZSTATUS SMINFO

Phase	Critical Objects
AGENT_REPORTING	SESSION PREFACE ZSTATUS SAPSTATS ZRSTATE SMINFO

ZTASKEND uses the following logic:

1. Whenever a critical object is presented, the critical object and the additional objects are processed by QMSG and deposited into queues for processing by Messaging Server (RMS).
2. If a critical object is not present, then the code invokes QMSG when an agent connects during the phases CATALOG_RESO and AGENT_REPORTING.
3. If a critical object is not found, then code does not invoke QMSG for the following agent phases: BOOTSTRAP (COP), SERVICE RESOLUTION and AGENT_SELF MAINTAINANCE.
4. Finally, ZTASKEND does not invoke QMSG to obtain error message objects (ZERRORM & ZERROR).

The table "ZTASKEND calls to QMSG for CORE Data for RIM and RMP" summarizes the Agent Connect phases and the objects collected during the ZTASKEND calls to QMSG, for CORE data going to Inventory Manager or the Portal.

ZTASKEND Calls to QMSG for CORE Data for RIM and RMP

Agent Connect Phase	QMSG call if not critical object?	QMSG call if critical object
Bootstrap - COP Resolution: for Client Operations Profile.	No	Collects these objects for CORE.RIM and CORE.RMP destinations: APPEVENT MSIEVENT SYNOPSIS RNPEVENT SESSION PREFACE ZSTATUS SMINFO
Agent Maintenance:	No	Collects these objects for CORE.RIM and CORE.RMP destinations: APPEVENT MSIEVENT SYNOPSIS RNPEVENT SESSION PREFACE ZSTATUS SMINFO
Catalog Resolution: Agent connects to the Configuration Server to obtain service resolution list.	Always. Collects these objects for CORE.RIM and CORE.RMP destinations: SESSION	See previous column, but also collects APPEVENT MSIEVENT SYNOPSIS RNPEVENT.

Agent Connect Phase	QMSG call if not critical object?	QMSG call if critical object
	PREFACE ZCONFIG ZMASTER ZSTATUS SMINFO	
Single Service Resolution: For each service to be resolved, agent makes another connection to the Configuration Server.	No	Collects the following objects for CORE.RIM and CORE.RMP destinations: APPEVENT MSIEVENT SYNOPSIS RNPEVENT SESSION PREFACE ZSTATUS SMINFO
Agent Reporting: At the end of service resolution. Agent data is reported back to the Configuration Server.	Always. Collects these objects for CORE.RIM destination: SESSION PREFACE ZSTATUS SAPSTATS ZRSTATE SMINFO	See previous column, but also collects APPEVENT MSIEVENT SYNOPSIS RNPEVENT

Adding Items to a Critical Object List

The ZTASKEND rexx code can be configured to add object names to the critical and additional object lists. The rexx variables `CriticalRIMObjects` and `CriticalRMPObjcts` contain the object names for each of these targets. The rexx function call to `BuildObjectList` is used to build the object list for each phase.

```
Call BuildObjectList ....
:
:
:
:
CriticalRIMObjects = "APPEVENT MSIEVENT SYNOPSIS RNPEVENT"
CriticalRMPObjcts = "SYNOPSIS"
```

The ZTASKEND rexx code contains additional information on how to alter these items.

Processing Always Objects

There is a section of the ZTASKEND rexx code to define objects that will always be processed, independent of the phase being processed. The larger Inventory Manager objects, FILEPOST,

WBEMAUDT and CLISTATS are processed this way. If these objects exist, then they are sent to the specified target. In addition to the larger Inventory Manager objects, the job objects: JOBSTAT, JOBPARM, and JOBTASK are also processed this way.

Adding Custom Objects to the BuildAlways Object List

This part of the rexx code can also be configured to add "custom" objects that are always delivered to the specified target. The rexx function `BuildAlways` is used to configure the target, queue and objects to process. The rexx code contains additional information on how to alter these items.

```
Call BuildAlways "inventory", InventoryQueue, "FILEPOST"
```

QMSG Method Syntax

The QMSG command/method is used to post Configuration Server objects for Inventory Manager, Portal and Application Management Profiles. QMSG reads the specified object and converts it to XML and then writes it to the specified queue.

The syntax of the QMSG method is given below:

```
qmsg -to <destination(s)> -queue <queue> -priority <priority> object1
object2 ... objectn
```

-to <destination(s)>

must be explicitly coded with one or more destinations, or targets. Messages going to multiple destinations have comma-separated entries. For example:

```
-to CORE.RIM,CORE.RMP
```

The Messaging Server destination values used by the delivered QMSG include:

```
-to CORE.ODBC
```

```
-to CORE.RMP
```

```
-to INVENTORY.ODBC
```

```
-to WBEM.ODBC
```

Each message destination requires an equivalent ROUTE defined for it in the `msg::register` router section of the appropriate Data Delivery Agent's `*.dda.cfg` file. The table "QMSG Destinations and DDA Configuration Locations" gives the destination values of QMSG and the configuration file and section used to define its ROUTE.

QMSG Destinations and DDA Configuration Locations

QMSG -to destination	Configuration File in <i>MessagingServer\etc</i> folder	Required ROUTE section
-to CORE.ODBC	<code>core.dda.cfg</code>	<code>msg::register corerouter</code>
-to CORE.RMP	<code>core.dda.cfg</code>	<code>msg::register corerouter</code>
-to INVENTORY.ODBC	<code>core.dda.cfg</code>	<code>msg::register inventoryrouter</code>

QMSG -to destination	Configuration File in <i>MessagingServer\etc</i> folder	Required ROUTE section
-to WBEM.ODBC	core.dda.cfg	msg::register wbemrouter

-queue <queue>

The queue is a directory relative to where QMSG.EXE exists. QMSG is located in the \bin directory of the Configuration Server. For example, if QMSG resides at

C:\Program Files\PSL\RCA\ConfigurationServer\bin\qmsg.exe

Then the queues would reside at:

C:\Program Files\PSL\RCA\ConfigurationServer\data\blue

C:\Program Files\PSL\RCA\ConfigurationServer\data\default

C:\Program Files\PSL\RCA\ConfigurationServer\data\green

C:\Program Files\PSL\RCA\ConfigurationServer\data\red

The parent directory of all queues is "data". For illustrative purposes, this example shows the existence of the (fictitious) blue, green and red queues. Note that the queue named "default" is the default queue.

Note: Queue locations are defined near the top of ZTASKEND. For Patch routing, the queue location is named in the parameters passed to QMSG from the five

SYSTEM.ZMETHOD.PATCH_* instances. The instances are named DEERROR, BUSTATUS, DESTATUS, and PASTATUS. For example:

Method = qmsg

Parameter = -to PATCH5 -queue patch <<object>>

To modify the parameters to pass, edit the value of the ZMTHPRMS attribute in the PATCH_* instance.

-priority

The priority is available to establish Messaging Server processing priority. If omitted, a default priority of 10 is given to the message. Valid values are 00 (highest priority) to 99 (lowest priority). For more information on Messaging Server processing priority, see the topic ["How Priority Establishes Messaging Server Processing Order"](#) below.

object1 [objectn]

The rest of the command line includes the names of the objects to queue, object1 object2... objectn. The objects are processed in the order specified; thus, depending on the destination, there might be a dependent order.

How Priority Establishes Messaging Server Processing Order

When ZTASKEND calls QMSG, the optional **-priority** parameter in the call assigns a processing priority to the message. Priority values can range from 00 to 99, with 00 reserved for critical processing and 99 being the lowest priority.

For messages waiting to be processed in the same queue, the Messaging Server processes all messages assigned to a higher priority (such as 10) *before* processing any messages assigned

to a lower priority (such as 20). Within a given priority, messages are processed using first in, first out (FIFO) order.

Note: The message priority remains the same for the life of the message. For example, if a message is forwarded from one Messaging Server to another, the message priority remains the same.

- Priority 10 is the default if a priority is not specified.
- Previously, ZTASKEND called QMSG with parameters to assign a lower priority of 20 to the larger objects collected for Inventory Manager Reports: these include file audits, wbm reporting data, and agent statistics (CLISTATS).
- This is no longer necessary because of the segregation of queues by object type.

If the Messaging Server is not able to process the messages as fast as they are delivered from QMSG, the lower priority messages will accumulate at the bottom of a queue location, even though newer messages with higher priorities are still being processed.

Configuring the Messaging Server

Use these topics to reconfigure or tune the Messaging Server after installation, or reconfigure or tune the data delivery agents for core or patch data.

Editing the Configuration Files

The Messaging Server and Data Delivery Agents standard installation allows configuration of several of the configuration parameters contained in the respective configuration files. You will need to edit the configuration file with a text editor to achieve a more customized environment.

All of the configuration files for the Messaging Server and Data Delivery Agents are found in the `\etc` directory of where the Messaging Server was installed.

All the configuration files for the Messaging Server and the associated Data Delivery Agents have similar configuration sections. Understanding these sections and the syntax used to configure them will aid in customizing your environment.

The structure for the RMS configuration file sections is given below:

```
msg::register <unique identifier> {
  TYPE          <RMS registered TYPE>
  <Configurable Variable for the registered TYPE>      <Value for that
Variable>
  <Configurable Variable for the registered TYPE>      <Value for that
Variable>
}
```

Each configuration section starts with the command `msg::register`. This signals the start of a configuration parameter for the Messaging Server and its modules.

A unique identifier follows the `msg::register` command. Within an instance of the Messaging Server, which includes the configuration files for the Messaging Server and all of the DDA modules, this unique identifier label can only be used once. The unique identifier is followed by a

curly brace "{". The configuration section for this TYPE must be ended by a closing curly brace for the entire configuration to work.

All configuration file sections have a TYPE identifier, which indicate the kind of work to be done by this section. These are the current acceptable TYPE designations:

- QUEUE
- ROUTER
- HTTP
- HTTPS
- HTTPD
- FILTER
- SQLBALANCER (only configurable in `core.dda.cfg`)

The table "Glossary of Section TYPEs and Configurable Parameters" describes the different section TYPES and their configurable parameters. The sections are listed in the general order in which they appear in the configuration files.

Glossary of Section TYPEs and Configurable Parameters

Section TYPE	Configurable Parameters
QUEUE Defines the directory where messages are placed for processing.	DIR The directory name of the queue. USE Specify the name of the unique identifier that will signify how to dispatch the message. Usually this is a ROUTER type. POLL By default, the Messaging Server is configured to poll the queue location every 10 seconds and post up to 100 objects at a time. To change the poll interval, modify the POLL parameter. COUNT Maximum number of objects to post at a time (during the polling interval defined by POLL). Default is 100 objects. ATTEMPTS Maximum number of attempts to retry a failed message delivery before discarding the message. Default is 200 attempts. (By default, the Messaging Server and Data Delivery Agents are configured to retry any failed posts every hour, and make up to 200 attempts.) DELAY Number of seconds to wait between attempts to retry a failed message delivery. Default is 3600 seconds, or one hour.

Section TYPE	Configurable Parameters
	Note: To calculate the maximum time that a message could stay in the queue before being discarded, take the DELAY time and multiply it by the ATTEMPTS value. Using the default settings, this is a DELAY of 3600 seconds x 200 ATTEMPTS, or approximately eight days.
ROUTER Defines where the messages are going to be sent.	ROUTE Delimit each Route by curly braces TO This is the address of the message. It is contained in the meta data file of each message when the message is created. USE Specify the unique name of a Messaging Server TYPE that will be used to dispatch the message, such as HTTP or HTTPS. (In a DDA file, the USE entries may also be COREODBC, WBEMODBC and PATCHDDAODBC).
HTTP This is a way to post the message to another server location using http protocol. The target server can be another Messaging Server where the message can be re-queued or it can be another part of the infrastructure, such as the Portal.	ADDRESS Delimit each address using curly braces. Multiple URL destinations can be specified within each HTTP TYPE as long as each has its own ADDRESS label and a different priority. PRI Denotes the priority in which messages are sent to the companion URL. The default priority is 10. The priority setting only matters if multiple ADDRESS entries are configured. If multiple ADDRESS entries are configured the lowest priority is tried first and if that fails the next priority URL is tried. This allows failover capability if there are network problems. URL Specifies the URL to use to send the message.
HTTPS This is a way to post the message using a secure socket. The HTTPS parameters are configured the same as the HTTP parameters--with the exception of the URL specification. The URL uses the <code>https://</code> convention.	ADDRESS Same as TYPE of HTTP. PRI Same as TYPE of HTTP. URL Specifies the URL using the <code>https://</code> convention.
HTTPD	PORT

Section TYPE	Configurable Parameters
Defines the parameters the Messaging Server will use to receive incoming messages.	Defines the Port used to “listen” for messages. Only one port specification can be used for a given Messaging Server. URLHANDLER Delimit with curly braces. If the incoming messages are to be deposited into different queues, depending upon the URL, the URLHANDLER must be used to delimit the USE and URL parameters. USE Specify the name of the QUEUE type that will receive the incoming messages. URL Specify the URL prefix that that will be accepted by the Messaging Server. When messages are received with the designated URL they are deposited in the associated queue. The QUEUE type must be defined when messages are received. All URL's specified must start with /proc/.
FILTER A means to route PATCH data into multiple SQL tables.	USE Specify the name of the unique identifier that will signify how to dispatch the message. TO This is the address of the message. It is contained in the meta data file of each message when the message is created. FILTER Used for PATCH object processing into multiple tables
COREODBC Used to post CORE messages into a SQL database.	DSN Data Source Name USER User ID for the DSN PASS Password for the DSN DSN_ATTEMPTS Number of attempts to connect to the DSN. Default is 1. DSN_DELAY Delay in seconds between attempts to connect to the DSN. Default is 5 seconds. DSN_PING Delay in seconds between pinging the database connection to verify the DSN is available. Default is 300 seconds.

Section TYPE	Configurable Parameters
	STARTUPLOAD No longer supported; any existing value is ignored. Note: The SQL tasks to create the tables and load the scripts for the Inventory Database are always performed upon Messaging Server and Data Delivery Agent startup.
PATCHDDAODBC Used to post PATCH data messages into a SQL or Oracle database. Only configurable in <code>patch.dda.cfg</code>	DSN Data Source Name USER User ID for the DSN PASS Password for the DSN DBTYPE Database type of MSSQL (for Microsoft SQL Database) or ORACLE (for an Oracle Database).
SQL The SQL database connection that handles all operational and inventory related data flows.	DSN Data Source Name SERVER The name of the SQL server USER User ID for the DSN PASS Password for the DSN AUTOCOMMIT The transaction management mode for the SQL server that decides whether to commit or roll back the data. DSN_DELAY Delay in seconds between attempts to connect to the DSN. Default is 5 seconds. DSN_PING Delay in seconds between pinging the database connection to verify the DSN is available. Default is 300 seconds. TRUNCATE

Section TYPE	Configurable Parameters
	Truncates the data if it exceeds the column size. If disabled, the Messaging Server throws an error message. DISABLE_TABLE_UPDATES The Messaging Server disables the provision to update the specified tables in the database. You can specify the tables for which you want to disable the updates. For example, to disable the tables <code>HDeviceConfig</code> and <code>HAppEvent</code> from being updated, add the following entry in the <code>*.cfg</code> file: <pre>DISABLE_TABLE_UPDATES {HDeviceConfig HAppEvent}</pre> Note: Patch management and Security and Compliance management tables get created even if their entries are specified in <code>SQL_DISABLE_TABLE_CREATION</code> parameter. If some of the tables are already created, specifying <code>SQL_DISABLE_TABLE_CREATION</code> parameter will not drop the tables. You will have to manually drop these tables. This parameter also disables index creation and insertion on the tables. Therefore, it is not required to use <code>DISABLE_TABLE_UPDATES</code> for the tables that are specified within <code>SQL_DISABLE_TABLE_CREATION</code> parameter. If this parameter is specified before First Time Setup Wizard, the tables specified in this parameter are not created. Note: Irrespective of the table entries in <code>SQL_DISABLE_TABLE_CREATION</code> parameter, when the Messaging server receives data pertaining to these tables for processing, it throws an error stating "Table or View does not exist" indicating that the respective table is not created. ENABLE-CORE Flag to enable the routing of CORE data objects placed in the queue locations for the core objects. ENABLE-INVENTORY Flag to enable routing of INVENTORY data objects placed in the queue locations for filepost (file audit inventory). ENABLE-WBEM

Section TYPE	Configurable Parameters
	Flag to enable routing of WBEM data objects placed in \data\wbem queue location. ENABLE-VM AUTOCREATE A switch to enable the creation of a new SQL file and table in the Inventory ODBC database when a new object class is received. Default is 0. 0 – Does not create a SQL file or table entry for a new object class. 1 – Creates a new SQL file and table entry for a new object class. AUTOLOAD Enables autoloading of schemas SQLPATH The path where the schemas are stored. MAPPATH The path where the mappings (*.tcl files) are stored. WBEM-PKEYS Enables creation of WBEM tables on demand with primary keys instead of unique indexes. ARCHIVE Enables archiving of messages. If enabled, the messages are moved to the Archive location. REJECTS Moves the messages to the REJECTS handler. The messages that fail to get processed, after maximum number of attempts have been tried to deliver this message, are moved to the REJECTS handler. Maximum number of attempts to retry a failed message VALIDATE-NLS_PARAMS National Language Support (NLS) connection settings that can be set for Oracle.
SQLBALANCER Routes all messages for a given device to a single sub-queue	QUEUE_DIR The directory where the queue is stored. For example, <InstallDir>\\Data\MessagingServer\core.

Section TYPE	Configurable Parameters
based on the host name, with a single worker process processing the data in the queue.	QUEUE_POLL By default, the Messaging Server is configured to poll the queue location every 1 second and post up to 100 objects at a time. To change the poll interval, modify the POLL parameter. QUEUE_COUNT Maximum number of objects to post at a time (during the polling interval defined by POLL). Default is 5000 objects. QUEUE_DELAY Number of seconds to wait between attempts to retry a failed message delivery. Default is 3600 seconds. QUEUE_ATTEMPTS Maximum number of attempts to retry a failed message delivery before discarding the message. Default is 200 attempts. (By default, the Messaging Server and Data Delivery Agents are configured to retry any failed posts every hour, and make up to 200 attempts.) QUEUE_REJECTS Moves the messages to the REJECTS handler. The messages that fail to get processed are moved to the REJECTS handler. SQL_DSN Data Source Name SQL_SERVER SQL_USER User ID for the DSN SQL_PASS Password for the DSN SQL_AUTOCOMMIT The transaction management mode for the SQL server that decides whether to commit or roll back the data. SQL_DSN_DELAY Delay in seconds between attempts to connect to the DSN. Default is 5 seconds. SQL_DSN_PING Delay in seconds between pinging the database

Section TYPE	Configurable Parameters
	connection to verify the DSN is available. Default is 300 seconds. SQL_ENABLE-CORE Flag to enable the routing of CORE data objects placed in the queue locations for the core objects. SQL_ENABLE-WBEM Flag to enable routing of WBEM data objects placed in \data\wbem queue location. SQL_ENABLE-INVENTORY Flag to enable routing of INVENTORY data objects placed in the queue locations for filepost (file audit inventory). SQL_ENABLE-VM SQL_AUTOCREATE A switch to enable the creation of a new SQL file and table in the Inventory ODBC database when a new object class is received. Default is 0. 0 – Does not create a SQL file or table entry for a new object class. 1 – Creates a new SQL file and table entry for a new object class. SQL_AUTOLOAD Enables autoloading of schemas

You can adjust default values and routing options by editing the *.cfg files, located in the \etc directory of where the Messaging Server was installed.

The base Messaging Server configuration file (rms.cfg) loads the individual Data Directory Agent (DDA) modules for posting the following object types: core, inventory, wbem, and patch. Each DDA has its own configuration file (*.dda.cfg) that defines where and how the objects are routed.

See the following topics for more information on how to configure or modify each configuration file.

- ["RMS.CFG File" on the next page](#)
- [" CORE.DDA.CFG File" on page 1390](#)
- [" PATCH.DDA.CFG File" on page 1392](#)
- [" Additional Tuning Topics" on page 1393](#)

Note: To configure the `USAGE.DDA.CFG` file, see ["Configuring Usage Data Files Store and Forward" on page 320](#).

To edit a Messaging Server or Data Delivery Agent `*.cfg` file:

1. Stop the Messaging Server before editing the `rms.cfg` file.
2. Edit the appropriate `*.cfg` file using any text editor. By default, the `*.cfg` files are located at: `<InstallDir>\MessagingServer\etc`.
3. Modify the sections using the information given in the following topics.

Note: All path entries in the configuration files must be specified using forward slashes.

4. Save your changes and restart the Messaging Server.

To encrypt a password entry for a database DSN in a configuration file:

1. The `PASS` value in all the `.cfg` files where specification of DSN parameters is necessary has to be encrypted. When the value is entered during the install process, the installation program takes care of encryption using AES, by default. If you need to modify the password, you can use the `nvdkit` utility to create an encrypted password using the AES or DES encryption method, and specify this encrypted value within the appropriate section of the configuration file. Enclose the encrypted value in quotation marks.

2. Open a command prompt and go to the directory where the Messaging Server is installed.

3. Enter the following command: `nvdkit`

4. At the `%` prompt, type the following command for AES encryption (the installation default):
`nvdkit encrypt <password_value> aes`

Alternatively, type the following command for DES encryption:

`nvdkit encrypt <password_value>`

The utility will return an encrypted password value.

5. Cut and paste this encrypted password value into the `configuration` file as the `PASS` value. Enclose the value in quotation marks.

RMS.CFG File

The Messaging Server Configuration file, `rms.cfg`, has the following main sections after the header. The configuration file loads separate modules for data delivery agents (DDAs), whose job is to post the objects to the configured locations.

There are separate data delivery agents for core data (Inventory Manager and Portal objects) and patch data (for Patch Manager)

Optional entries in the `rms.cfg` file can include SSL support, a `"msg:register httpd"` section if this Messaging Server is receiving forwarded messages from another Messaging Server, and a `"msg:register default"` section if this Messaging Server is posting messages from a single queue location of `\data\default`.

Additional Sections in the RMS.CFG File

- **Required Packages**

Do not remove the following lines at the top of the `rms.cfg` file

```
package require nvd.msg
```

```
package require nvd.httppd
```

- **SSL Configuration Parameters**

If the Messaging Server is SSL enabled, this Overrides Config { } section in the `rms.cfg` is used to define the necessary certificates and parameters for SSL support. It also includes the command `module load tls`, which loads the code necessary to support SSL.

- **log::init**

Logging configuration. These settings apply to all Messaging Server logging. For details on changing the logging configurations, see ["Configuring the Log Level, Log Size and Number" on page 1395](#). The log files are located in the Logs directory of where the Messaging Server was installed.

- **loglevel 3**

Default logging level for entries written to the log files. Default is 3. Normally, this is not changed.

- **loglines 200000**

The default number of lines contained in a log before the log is rolled over.

- **loglimit 7**

The default number of rolled logs to keep.

- **Load Data Delivery Agents for Posting Objects**

Include the following lines at the end of `rms.cfg` to load the data delivery agents needed to post each type of object.

- **dda.module load core**

Posts CORE message data to a SQL or Oracle database and, optionally, CORE message data to the Portal directory. See ["CORE.DDA.CFG File" on the next page](#) for more information on how to configure the posting of core objects.

- **dda.module load patch**

Post PATCH message data to a SQL or Oracle database. See ["PATCH.DDA.CFG File" on page 1392](#) for information on how to configure the posting of patch objects.

- **(Optional) msg::register default, msg::register router, and msg::register**

`<rim|rmp|other>`

These sections, if they exist, define how the Messaging Server handles the messages placed by QMSG in an existing `/data/default` location (or `/data/default queue`).

- **(Optional) Configure the maximum log size and number of logs**

Note that these options apply for each Worker assigned to process the `\data\default` queue. For more information, see ["Configuring the Log Level, Log Size and Number" on page 1395](#).

- **(Optional) Enable usage management tables**

If you enable usage management table creation, `USAGEENABLED` value in the Overrides Config section gets modified to **local** indicating the creation of usage tables. If it is not

enabled, the usage management tables are not created and the value of USAGEENABLED remains **none**.

- **(Optional) Configure the frequency to check the Satellite server health**

You can configure the frequency when the Satellite server health is checked and the data is sent to the Core server. The default frequency is set to 21600 seconds (six hours). You can set the HEARTBEAT attribute in the Overrides Config section in rms.cfg on the Satellite server. You can set the minimum value as 1 second and the maximum value as 86400 seconds.

Snippet to set the frequency to 12 hours:

```
# Set HEARTBEAT frequency to 12 hours
Overrides Config {
  HEARTBEAT 43200
}
```

CORE.DDA.CFG File

The `core.dda.cfg` file defines where and how to route objects placed in queue locations for core objects. As mentioned previously, the core objects are objects created on the agent, available during the agent connect process and used in reports. Examples of core objects are ZMASTER, ZCONFIG, and SESSION.

- To activate the `core.dda` module, the command `dda.module load core` must be included at the end of the `rms.cfg` file.
- For a Messaging Server co-located with a Configuration Server, the queue locations are folders where the QMSG executable places messages:
Queue folder for core messages: `<InstallDir>\data\core`
- For a Messaging Server receiving messages forwarded from another Messaging Server `core.dda` module, URLs define the locations on which to listen for messages:
URL for core messages: `http://localhost:3461/proc/core`

Several configurations are possible.

- Core object data can be routed using ODBC directly to the back-end SQL database. This option is best used when the database is close to the current location.
- Core data can be forwarded to another Messaging Server. This option is used to place the objects as close as possible to the SQL database, before ODBC posting, in order to avoid slow network response.
- Core messages for the Portal can be routed using HTTP to a Portal Zone, or discarded using the built-in `/dev/null` location.

The `core.dda.cfg` file has the following main sections:

- **msg::register core**

This is the SQLBALANCER type for the `core.dda` configuration. Routes all messages for a given device to a single sub-queue based on the host name, with a single worker process processing the data in the queue.

The parameters are summarized below:

- DIR defines the full path of the `/data/core` location. This is set at installation time.
- USE defines where the routing information for each TO label is located.
- POLL and COUNT establish the polling interval and post quantity for the Messaging Server, which determines how often and how many messages are posted at a time. To adjust this, see the topic ["Configuring the Poll Interval and Post Quantity" on page 1393](#).
- Retry Attempts (DELAY and ATTEMPTS), after maximum retry attempts, a message is automatically discarded
- **msg::register core.odbc**
This is the SQL type. Routes messages to sub-queues based on the hostname. Defines a DSN, User, and Password to post core data directly to an ODBC database. For details on the entries, see the topic ["ODBC Settings for CORE, INVENTORY and WBEM Objects" below](#).
- **(Optional) Configure the maximum log size and number of logs**
Note that these options apply for each Worker assigned to process the specific queue. For more details, see ["Configuring the Log Level, Log Size and Number" on page 1395](#).

ODBC Settings for CORE, INVENTORY and WBEM Objects

The following settings are configured in the **msg::register core.odbc** section of `core.dda.cfg`:

DSN	Specify the Data Source Name (DSN) for the Inventory ODBC database. Enclose the entry in quotes.
USER	Specify the user name for the Inventory ODBC database identified in the DSN parameter.
PASS	Specify the password for the user of the Inventory ODBC database. When modifying a password entry, obtain an encrypted entry using the procedure "To encrypt a password entry for a database DSN in a configuration file: " on page 1388 . Both AES (the installation default) and DES password encryption methods are supported.
DSN_ATTEMPTS	Number of attempts to connect to the Inventory Manager database DSN. Default is 1.
DSN_DELAY	Delay in seconds between attempts to connect to the Inventory Manager database DSN. Default is 5 seconds.
DSN_PING	Delay in seconds between pings to the database connection to verify the DSN is available. Default is 300 seconds.
AUTOCREATE	A switch to enable the creation of a new SQL file and table in the Inventory ODBC database when a new object class is received. Default is 0. 0 – Does not create a SQL file or table entry for a new object class. 1 – Creates a new SQL file and table entry for a new object class.
STARTUPLOAD	No longer supported; any coded value is ignored.

	Note: The SQL tasks to create the tables and load the scripts for the Inventory Database are always performed upon Messaging Server and Data Delivery Agent startup.
SQLPATH	The path where the schemas are stored.
MAPPATH	The path where the mappings (*.tcl files) are stored.
WBEM-PKEYS	If true, enables creation of WBEM tables on demand with primary keys instead of unique indexes.
ARCHIVE	Enables archiving of messages. If enabled, the messages are moved to the Archive location.
REJECTS	Moves the messages to the REJECTS handler. The messages that fail to get processed, after maximum number of attempts have been tried to deliver this message, are moved to the REJECTS handler.
VALIDATE-NLS_PARAMS	The NLS connection setting that can be set for Oracle.

PATCH.DDA.CFG File

The `patch.dda.cfg` file defines where and how to route the objects placed in the `\data\patch` queue location. Most of the configuration is done automatically during the installation of the Messaging Server.

The `patch.dda.cfg` file has the following main sections after the header and required lines:

```
# DO NOT REMOVE FOLLOWING LINE
package require nvd.msg.patchodbc
package require nvd.msg
```

- **msg::register patchq**

This is the QUEUE type for the `patch.dda.cfg`. Defines how the Messaging Server handles the messages placed by QMSG in the `/data/patch` location (or `/data/patch queue`). The parameters are summarized below:

- TYPE of QUEUE defines a Messaging Server location and polling values for picking up messages.
- DIR defines the full path of the `/data/patch` message location. This is set at installation time.
- USE defines where the routing information for the TO PATCH objects are located.
- POLL and COUNT establish the polling interval and post quantity for the Messaging Server, which determines how often and how many messages are posted at a time. To adjust this, see the topic "[Configuring the Poll Interval and Post Quantity](#)" on the next page.
- Retry Attempts (DELAY and ATTEMPTS), after maximum retry attempts, a message is automatically discarded

- **msg::register patchrouter**

This is the ROUTER type for the `patch.dda.cfg`. Configures routing assignments for each –To patch type of message. At least one route is specified for each –To type:

-To PATCH

-To PATCH5

- **msg::register patchddasummarize**

Translates any ZOBJSTAT patch reporting objects from pre-5.0 agents into the PATCH reporting objects used in Version 5.0 or higher patch reporting. Not configurable.

- **msg::register patchddaodbc**

Defines a DSN, User, Password and DBTYPE to post patch data directly to an ODBC database. For details on the entries, see ["ODBC Settings for PATCH Objects" below](#).

Note: This section may be pre-configured during the install.

- **(Optional) Configure the maximum log size and number of logs.**

Note that these options apply for each Worker assigned to process the specific queue. For more information, see ["Configuring the Log Level, Log Size and Number" on page 1395](#).

ODBC Settings for PATCH Objects

The following settings are configured in the **msg::register patchddaodbc** section of `patch.dda.cfg`:

DSN	Specify the Data Source Name (DSN) for the Patch Manager ODBC database. Enclose the entry in quotes.
USER	Specify the user name for the Patch Manager ODBC database identified in the DSN parameter. Enclose the entry in quotes. Default value is {sa}.
PASS	Specify the password for the user of the Patch Manager ODBC database. Enclose the entry in quotes.
DBTYPE	Specify <code>SMSSQL</code> for a Microsoft SQL Server, or <code>ORACLE</code> for an Oracle SQL Database. Enclose the entry in quotes. Default is MSSQL.

Additional Tuning Topics

Configuring the Poll Interval and Post Quantity

By default, the Messaging Server is configured to poll the queue location every 10 seconds and post up to 100 objects at a time. To change the poll interval, modify the POLL parameter in the appropriate configuration file and **msg::register** section with a TYPE of QUEUE.

To change the maximum number of objects to be posted at a time, modify the COUNT parameter in the same section.

If the objects being posted are very large, it is recommended to increase the POLL interval to give sufficient time to complete the posting.

Configuring for Retry Attempts

The Messaging Server and the Data Delivery Agents are configured to retry any message that fails to post. By default, the Messaging Server will retry posting the message every hour, and make up to 200 attempts. These values are defined by the DELAY and ATTEMPTS entries in the sections of the configuration files that have a TYPE of QUEUE. See the table ["Configuration File and Section Used to Modify Queue Processing"](#) for a list of msg::register sections used to modify QUEUE processing.

Note: After the last attempt, the message is automatically discarded from the queue without being posted.

To calculate the maximum time that a message could stay in the `/data/default` queue, take the DELAY time and multiply it by the ATTEMPTS value. Using the default settings, this is a DELAY of 3600 seconds x 200 ATTEMPTS, or approximately eight days.

You can adjust the DELAY and ATTEMPTS values in configuration files to establish a different maximum time that a message could stay in the `/data/default` queue. Specify the DELAY in seconds.

Configuring for Failover

You can configure the Messaging Server with one or more servers defined for failover support when defining HTTP types. When defining failover servers, each one is assigned a PRI value. If the Messaging Server fails to connect with the first server (that is, the server with the lowest PRI value) it will try the next server on the list (or, the next higher PRI value).

Note: This PRI value is separate from the `-priority` value assigned by QMSG for processing priority. The PRI value and the QMSG -priority value are not related.

To set failover in a `*.dda.cfg` file:

Failover support is added by inserting additional ADDRESS entries to the appropriate section of an HTTP type in a DDA cfg file.

The URL entries will be tried in order of PRI (priority) starting with the *lowest* PRI value.

The code sample below shows sample modifications to the msg:register rim section of `core.dda.cfg` for failover. The code in **bold** was added to define a failover server for Inventory Manager processing.

```
msg::register rim {
    TYPE          HTTP
    ADDRESS       {
        PRI       10
        URL       http://rim1:3466/proc/rim/default
    }
    ADDRESS {
      PRI  20
      URL  http://rim2:3466/proc/rim/default
}
```

```
}
}
```

Configuring the Log Level, Log Size and Number

The log files for the Messaging Server (`rms.log`) reside in the Logs folder of the MessagingServer directory,
`<InstallDir>\ConfigurationServer\MessagingServer\logs`.

Changing the Logging Level

The `log::init` section at the beginning of the configuration file establishes the logging level for all Messaging Server logging. The default logging level is 3. Valid levels are 0 (no logging) to 10 (maximum logging level). Normally, the log level is not changed unless requested by a customer support person for troubleshooting purposes. The following lines show the log level increased to 4:

```
log::init {
    -loglevel 4
}
```

Changing the Size and Number of Log Files

The Messaging Server writes entries to a set of log files for each WORKER. There is generally one WORKER attending each queue location. Queue locations include:

```
\data\core
\data\inventory
\data\patch
\data\wbem
```

or

```
\data\default
```

By default, the Messaging Server creates and retains seven log files per worker, each file having a maximum of 5000 lines. The log files are located in the Messaging Server `\logs` directory.

The following line in the `rms.cfg` file establishes the default logging:

```
-loglines 200000
```

To control the size and number of logs created for each worker, add or modify the following entries below the `log::init` section of the `rms.cfg` file:

```
-loglines maximum_lines
-loglimit maximum number of logs
```

where *maximum_lines* is the maximum number of lines for a given log file. After the maximum is reached, another log file is created, until the *maximum number of logs* specified in the `log.configure -size` entry is reached. After the maximum number of logs is reached, the oldest log files are deleted.

The next code sample illustrates an RMS.CFG file containing entries to limit each log file to 1000 lines, and allow up to 10 log files to be retained.

```
log::init {
    -loglevel 3
    -loglines 1000
    -loglimit 10
}
```

Configuring the Messaging Server to Discard or Drain Messages

The location of `/dev/null` is built into the Messaging Server for discarding messages. When the `USE` parameter is set to `/dev/null` in any of the `ROUTER` type sections of the Messaging Server or Data Delivery Agent configuration files, the messages being processed will be successfully discarded without an error.

Example: Discarding Messages for the Portal

For example, to discard all RMP messages placed in the `\data\core` queue (these messages have a `TO` label of `CORE.RMP`), specify the following `ROUTE` in the `msg::register` `corerouter` section of `core.dda.cfg`:

```
msg::register corerouter {
    TYPE      ROUTER
    . . .
    ROUTE      {
        TO      CORE.RMP
        USE      /dev/null
    }
}
```

Example: Draining a Message Queue

As another example, to quickly drain an entire queue, temporarily replace `USE router` in the `msg::register` section for the queue with `USE /dev/null`. See the table ["Configuration File and Section Used to Modify Queue Processing"](#) for a list of the configuration files and sections that control each queue type. After draining the queue, reset it back to `USE router`.

Configuring the Messaging Server to Route Portal Messages

About the Portal Data Queue (rmpq) in CORE.DDA.CFG

The default `core.dda.cfg` configuration re-queues `CORE` messages that are to be posted to the Portal directory into its own queue, named `rmpq`. This allows for separate throttling of the `CORE` messages being posted via `HTTP` to the Portal directory, as opposed to the `CORE` messages being posted using `ODBC` to another database.

The following code shows the sections from `core.dda.cfg` used for this purpose.

```
# Requeue and process just RMP data to throttle the data flow
msg::register rmpq {
```



```

TYPE          QUEUE

DIR           ../ConfigurationServer/data/rmp
USE           rmpqrouter

POLL          10
COUNT        30
DELAY         3600
ATTEMPTS      200
}

msg::register rmpqrouter {
TYPE          ROUTER

ROUTE         {
TO            CORE.RMP
USE           rmpqhttp
}
}

msg::register rmpqhttp {
TYPE          HTTP

ADDRESS       {
PRI           10
URL           http://localhost:3471/proc/xml/obj
}
}

```

Restoring Routing for Portal Messages

If you initially installed the Messaging Server to discard Portal messages, use the steps below to begin routing Portal data to a Portal Server and Port.

To modify `core.dda.cfg` for posting Portal data:

1. Use a text editor to edit the `core.dda.cfg` file, located in the `etc` folder of the Messaging Server directory.
2. Look for the section starting with `msg::register corerouter`, and then find the entry for the ROUTE defining `CORE.RMP` messages. The Portal data that is being discarded will show the following entry with a USE value set to `/dev/null`:

```

ROUTE         {
TO            CORE.RMP
USE           /dev/null

```

3. Change the USE value from `/dev/null` to `rmpq`, as shown below:

```

TO            CORE.RMP
USE           rmpq

```

4. Next, locate the `msg::register rmpqhttp` section near the end of the `core.dda.cfg` file, and find the default URL entry, shown below:

```
msg::register rmpqhttp {
    TYPE      HTTP

    ADDRESS   {
        PRI    10
        URL     http://localhost:3466/proc/xml/obj
    }
}
```

5. Edit the URL value of `localhost:3466` to indicate the host and port of your Portal server. For example, a Portal with a hostname of `PORTALSVR` on port `3471` is defined with the following URL entry:

```
URL      http://PORTALSVR:3471/proc/xml/obj
```

Host names can be specified using an IP address or a DNS name.

6. Save your changes and restart the Messaging Server.

Enhancing Messaging Server Performance

If the Messaging Server posts large amount of objects to the ODBC and the Portal, the Portal throughput for Messaging Server object degrades. To improve Messaging Server performance for posting objects to the Portal, apply the following configurations in the `rmp.cfg` file. The modified configuration decouples the Messaging Server object posting to the Portal from Messaging Server's main ODBC path. The Messaging Server stores the Portal outbound objects to the disk. The Portal then invokes a separate thread dedicated for processing these messages at its own pace.

The Messaging Server posts the messages into the Portal queue; the Portal then initiates a thread that reads from this folder and updates the destination (Database, OpenLDAP, or Configuration Server database).

These configuration should be applied only to the core Messaging Server.

To improve the Messaging Server performance for posting objects to the Portal, complete the following steps:

1. Stop the RCA Portal service.
2. Add the following parameters in the `<InstallDir>/ManagementPortal/etc/rmp.cfg` file in the `rmp::init { }` block:

```
ENABLE_RMSQUEUE 1
RMSQUEUE_PATH
RMSQUEUE_REJECTS_PATH
```

where,

`RMSQUEUE_PATH` is the path to the Messaging Server folder where the Portal objects are stored.

`RMSQUEUE_REJECTS_PATH` is the path to the rejects folder

For example:

```
ENABLE_RMSQUEUE 1
RMSQUEUE_PATH {<InstallDir>/Data/MessagingServer/rmp}
RMSQUEUE_REJECTS_PATH {<InstallDir>/Data/MessagingServer/rejects}
```

3. Add the following line of code in the `<InstallDir>/ManagementPortal/etc/RCA-RMP.rc` file in the `Overrides Config { }` block:
- ```
RMS_STORE_SUBSCRIBERS 0
```

4. Restart the RCA Portal Service.
5. Ensure that ATTEMPTS is set to the desired value in the  
`<InstallDir>/ManagementPortal/etc/rms.cfg` in the `msg::register rmp { }`  
 block.

## Disabling Processing of Messages in a Queue

The objects in a disabled queue are not polled or posted. You may want to disable processing during peak agent connect periods if resources are in contention, or if you know a target server is down.

You can re-enable the processing at night or during slower periods to allow the Messaging Server to transfer the messages.

To disable queue processing using `WORKERS -1` (minus 1) value:

To disable a queue from being polled and its contents posted, set a `WORKERS` value of `-1` (minus one) at the end of the appropriate `msg::register` section for that queue.

### Configuration File and Section Used to Modify Queue Processing

| Queue                                  | Configuration File in<br>MessagingServer\etc<br>folder | msg::register<br>section    |
|----------------------------------------|--------------------------------------------------------|-----------------------------|
| \data\core                             | core.dda.cfg                                           | msg::register coreq         |
| \data\inventory                        | core.dda.cfg                                           | msg::register<br>inventoryq |
| \data\patch                            | patch.dda.cfg                                          | msg::register patchq        |
| \data\wbem                             | core.dda.cfg                                           | msg::register wbemq         |
| \data\default (in earlier<br>versions) | rms.cfg                                                | msg::register default       |

To add a `WORKERS` value to create multiple processes (`WORKERS`):

1. Use a text editor to open the appropriate \*.cfg file for the Messaging Server or Data Delivery Agent processing the queue to be disabled. The table "[Configuration File and Section Used to Modify Queue Processing](#)" identifies the configuration file to use for each queue type.
2. Locate the 'msg::register' section named in the table "[Configuration File and Section Used to Modify Queue Processing](#)"; it will be defined with `{ TYPE QUEUE }`.
3. Add or modify a line for `WORKERS` with a value of `-1` (minus 1) to the end of the section. A sample entry for disabling a wbem queue is shown below with a `WORKERS` value of `-1`.

```
msg::register wbemq {
 TYPE QUEUE

 DIR C:/Progra~/Hewlet~/CM/ConfigurationServer/data/wbemq
 USE router

 POLL 10
```

```
COUNT 100
DELAY 3600
ATTEMPTS 200
WORKERS -1
}
```

4. Save your changes and exit the editor.

To enable a disabled queue:

To enable a previously disabled queue, change the WORKERS value in the msg::register section of appropriate configuration file from -1 to 1. The number of WORKERS indicates the number of independent and lightweight processes to be started for this queue. It is recommended to use 1 as the default configuration value for a Messaging Server that is processing multiple queues with Data Delivery Agents.

## Modifying the Priority in which Messages are Processed

**Note:** With the adoption of specific queues for each object type, the priority feature is no longer applicable. However, the code for processing messages in increasing priority order has not been disabled.

To modify the priority in which messages are processed, see the earlier topic ["How Priority Establishes Messaging Server Processing Order"](#) on page 1378.

## Configuring Messaging Server to Report Differenced Objects

Sentient Messaging Server provides an efficient and flexible messaging system to route the data objects. It monitors pre-defined data queues and dynamically routes data objects to one or more external destinations. Sentient agent does not check whether the data objects have changed since last scan to send the difference in the data objects to the Messaging Server. Instead, Sentient agent sends complete data to the Messaging Server. Sometimes the Messaging Server queue grows too large, which impacts performance and causes delay in reporting. With the Reporting Differencer feature enabled, Sentient agent sends only the differenced data objects to the Messaging Server.

Sentient agent invokes the `RepObjDiff.exe` based on the settings configured in the REPTDIFF class in the CLIENT domain. `RepObjDiff.exe` creates a SQLite database on the managed device that contains all the data objects. The data objects in the OUTBOX folder are replaced with the differenced data objects containing fewer messages instead of complete data. The Messaging Server is configured to handle the 'Differenced' Reporting Objects. The PATCH.DDA file processes the differenced object data as well as full object data to generate the reports.

Complete the following steps to configure the Messaging Server to report difference data objects:

1. Click **Start > Programs > Sentient Enterprise Administrator > Sentient Enterprise Administrator CSDB Editor**. The logon dialog box opens.
2. Type your User ID and Password. By default, the user name is `ADMIN` and the password is `secret`.

3. Click **OK**. The Sentient CSDB Editor window opens.
4. Navigate to the PRIMARY.CLIENT.REPTDIFF.
5. Set the following parameters:
  - a. Set FLUSHDB to Y to recreate the reporting reference information.
  - b. Set FULLREFR to Y to refresh the complete data and sends it to Messaging Server.
  - c. Set REPTDIFF to Y to enable the Reporting Differencer.
  - d. Set REPTOBS to one or more data objects for differencing, such as DESTATUS.edm, BUSTATUS.edm, and PASTATUS.edm. Sentient agent invokes the RepObjDiff.exe only if one of these objects are available in the OUTBOX folder and the Reporting Differencer is enabled.

For more information on REPTDIFF class, see ["RepObs Differencer \(REPTDIFF\) Class" on page 951](#).



# Chapter 15

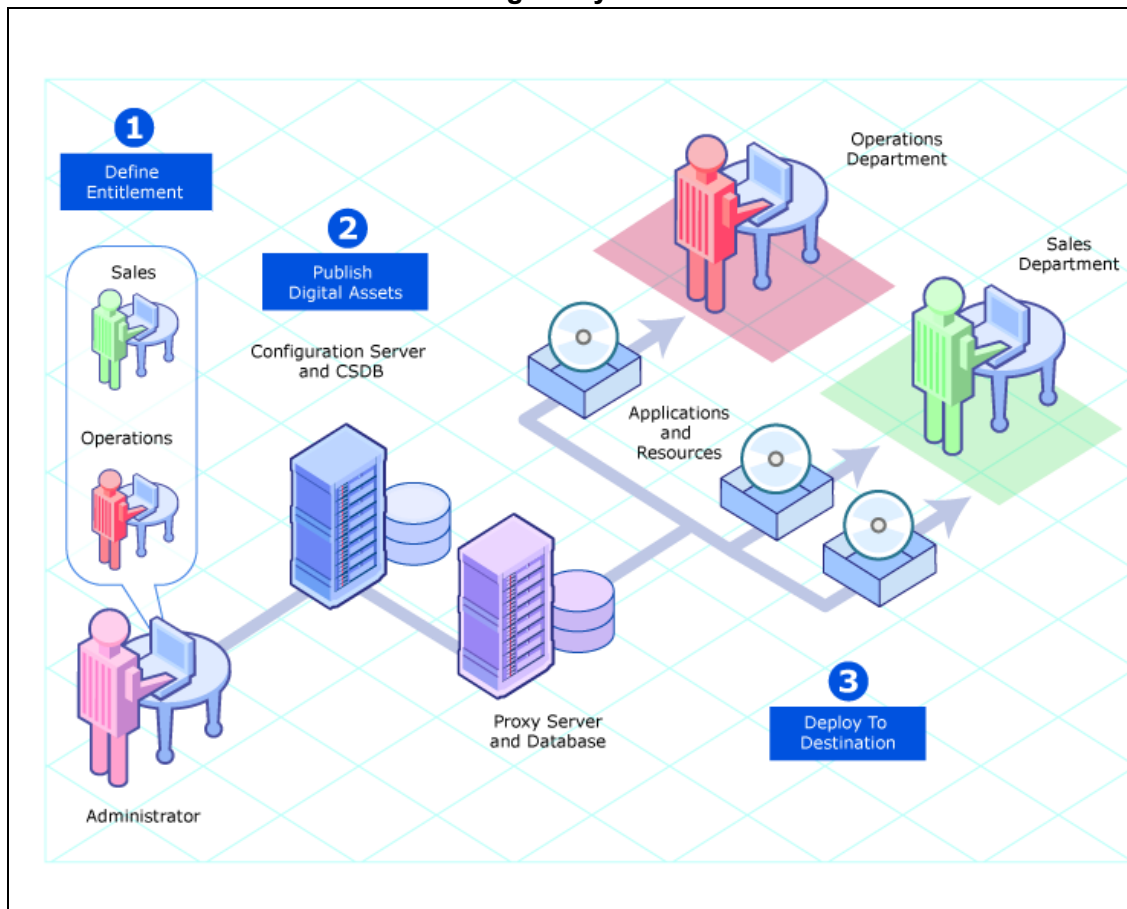
## Proxy Server Management

### Introduction

For Windows platforms, the Proxy Server is a part of the Satellite server and it is used to cache data. You can also install the Proxy server on Linux platforms. For more information on how to install the Proxy Server on Linux system, see *Sentient Enterprise Installation and Upgrade Guide*. The Proxy Server has the ability to service multiple, concurrent Sentient agent requests. The Proxy server supports two types of data cache: Static and Dynamic.

When the Proxy Server is used, it is the primary repository for Sentient agent data. After Sentient agent determines the resources required for its desired state, it can request those resources from the Proxy Server. The Proxy Server has the ability to service multiple, concurrent Sentient agent requests. The following figure depicts the Sentient infrastructure using a Proxy Server to deploy applications to Sentient agents.

#### Client Automation Infrastructure Using Proxy Server



## Proxy Server Processing

The logical flow for a typical Sentient agent request to a Proxy Server is as follows (assuming all components are enabled):

1. The Sentient agent's resolution process indicates it needs resources, and it should request them from a Proxy Server.
2. The Sentient agent sends a request to the Proxy Server, which is received by the Proxy Server front-end protocol. The agent request is validated, and passed to the cache manager.

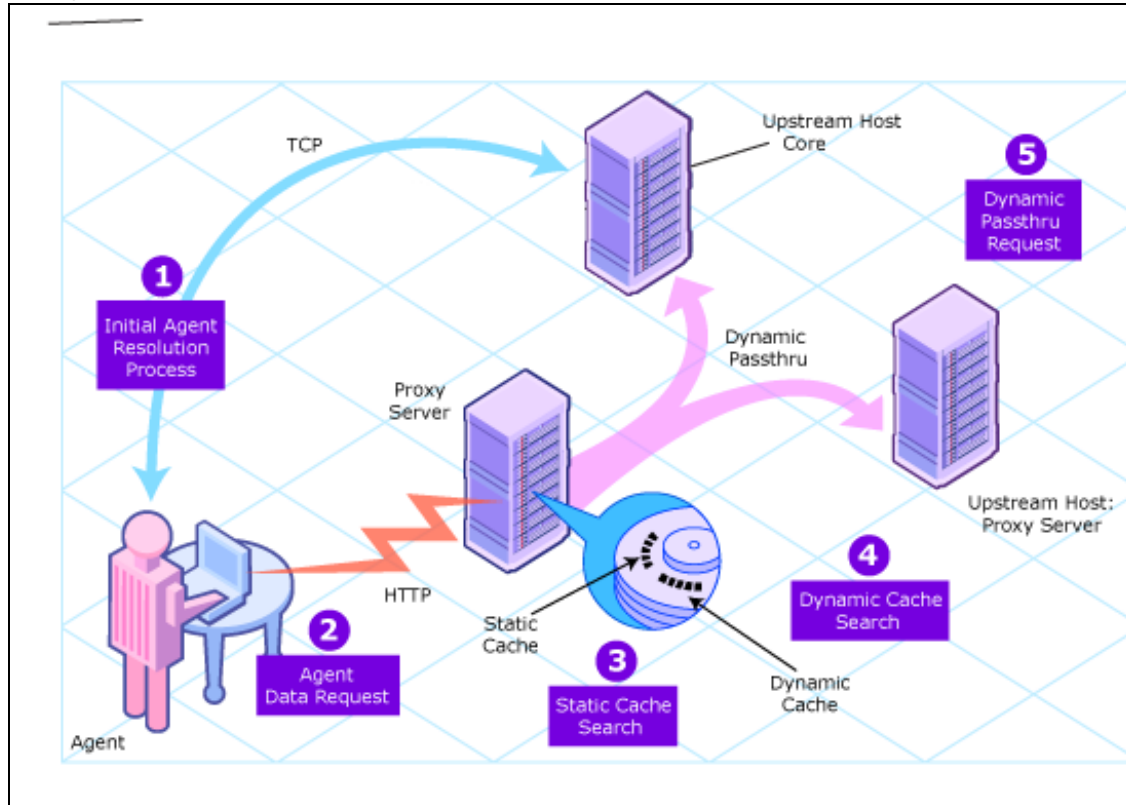
**Note:** The default (and recommended) protocol for Sentient agent communication with the Proxy Server is HTTP. Optionally, TCP is also available. A Proxy Server can service multiple, concurrent agent requests using both protocols simultaneously.

3. The local static cache is searched, and if the data is found, the request is satisfied and the data is sent to the Sentient agent.
4. If the data is not in the local static cache, the local dynamic cache is then searched. If the data is found, the request is satisfied.
5. If the file does not reside in the dynamic cache, Dynamic PassThru requests the file from the defined upstream host (typically, another Proxy Server) using HTTP. If there is a connection failure with the upstream host, and multiple upstream hosts were defined (for fail-over support), Dynamic PassThru tries the next host on the list. The original request is satisfied, and the file is stored in the dynamic cache for future requests.

The following figure illustrates the Proxy Server process flow for handling Sentient agent requests.



### Proxy Server process flow



## When to use a Proxy Server

Proxy Servers are beneficial in your environment if you have many Sentient agent computers requesting the same resources from the same location. When data is cached on the Proxy Server, the demand placed on the Configuration Server is decreased, allowing the Configuration Server to allocate more resources to other tasks.

Placing Proxy Servers at strategic points in your network increases the rate at which data is transferred. The connection between subscribers and the Proxy Server may be more efficient than the connection between the subscribers and the Configuration Server. The factors that determine the efficiency of a connection between a server and an agent computer include hardware capability, network bandwidth, workload on the servers, network traffic patterns, and the volume of software to be distributed.

**Note:** The Proxy Server is not a generic proxy, but rather specifically designed to manage and distribute Sentient resources.

## Preloader and Desired-State Policy Resolution

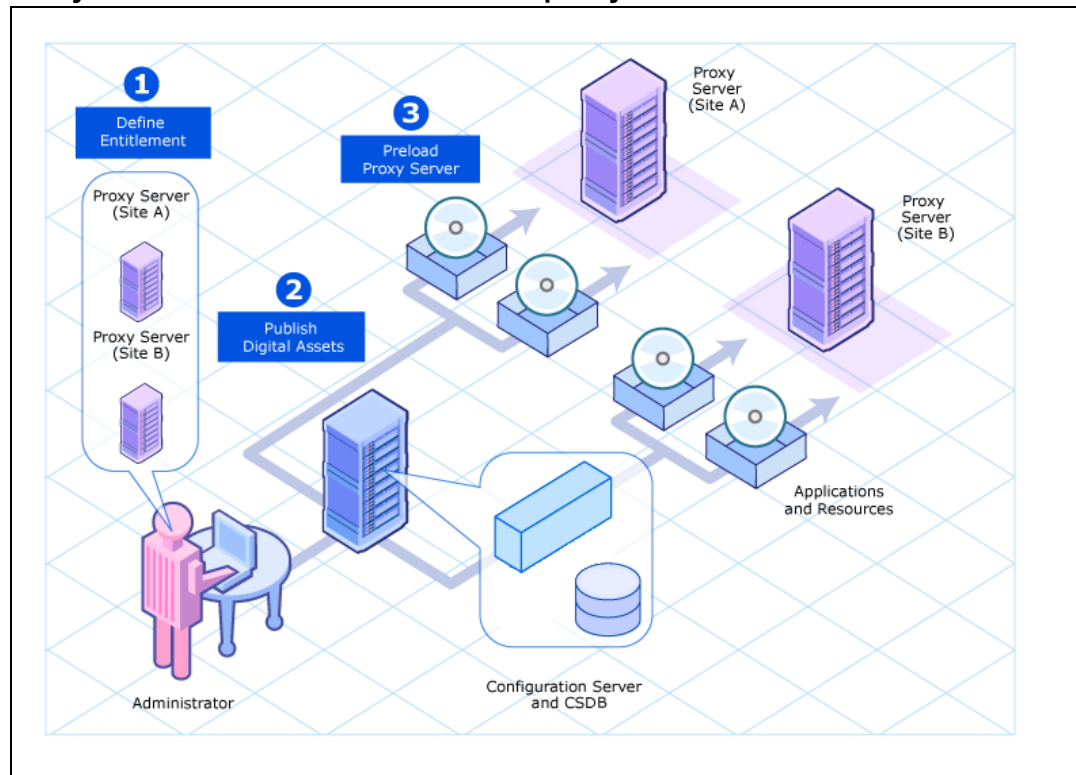
The preload process is an application of the usual Sentient desired state policy resolution for Sentient agents, with the following specific elements. Each element is illustrated in figure *Proxy Server Preload as a desired state policy resolution*.

### 1. Define Entitlement

The entitlement for preloading a Proxy Server is defined in the POLICY Domain as follows:

- **User**  
The machine identity of the Proxy Server being managed (or preloaded).
  - **Applications**  
The software that is being preloaded to the Proxy Server's static cache. This should include all applications normally requested by the set of Sentient agents that will be assigned to the Proxy Server.
2. **Publish Digital Assets**
    - **Application Files**  
The components that make up the applications. When publishing MSI applications for distribution from a Proxy Server, use the techniques discussed in this guide to have the Administrator Control Point (ACP) preloaded to the Proxy Servers but not distributed to the Sentient agents.
  3. **Preload Proxy Servers**
    - **Deployment Source and Protocol**  
Resources to preload the Proxy Servers can come from the Configuration Server or another Proxy Server. Resources can be deployed using TCP or HTTP.
    - **Deployment Destinations**  
In this case, the Proxy Server's static cache location is the preload deployment destination.

### Proxy Server Preload as a desired state policy resolution



## Dynamic Cache Management

Dynamic PassThru also manages this cache, purging files that have not been requested in a previously configured number of days. (This avoids keeping files in the dynamic cache after they

exist in the static cache.) To support the purge process, an index file is maintained to keep track of when the files were last used. The purging of aged files will also run whenever the index file is saved. You can schedule the index file to be saved on a daily or more frequent basis. However, a scheduled save of the index file is skipped if dynamic cache resources were not used since the last save.

The dynamic cache can also be purged, as necessary, when a user-specified maximum file size is exceeded. The least-used files are deleted until the dynamic cache size is below the maximum file size. A "freespace" purge option allows you to purge down to a low-water mark, which is a certain percentage below the maximum file size. This "freespace" eliminates constant purging in an active dynamic cache environment.

Another purge option allows you to define "large files", and then exempt these large files from the first pass of a size-based purge. If large files are purged and then later requested, an undue load could be placed on the network. Use the large file options to alleviate this load.

## Proxy Server Configuration File

The Proxy Server configuration file, `rps.cfg`, is located in the `ProxyServer\etc` directory. Review the configuration parameters and make modifications to the `rps::init` section for the front-end communications protocol or the static or dynamic cache. The table *Configuration file rps.cfg parameters* in the section "RPS.CFG Configuration Parameters" on page 1409 defines all parameters in the `rps.cfg` file and their default values.

## Sample rps.cfg File After an Installation

The code below shows a sample `rps.cfg` file for Proxy Server for Windows. Your `rps.cfg` file may show additional entries.

```

rps::init {
 -stager 0
 -stager-port 3461
 -stager-trace 0

 -httpd 1
 -httpd-prefix "/RESOURCE"

 -static-root
"C:/Program~/PSL/RCA/Data/ProxyServer/static"
 -static-trace 0
 -static-type agent
 -static-host RCS IP
Address or hostname
 -static-port 3464

```

```

 -static-http-port 3466
 -static-https-port 443
 -static-user RPS_
satellite_hostname
 -static-logsize ""
 -static-proxy-host ""
 -static-proxy-port ""
 -static-proxy-user ""
 -static-proxy-pass ""
 -static-ssl 0
 -static-use-datauri 1

 -dynamic 0
 -dynamic-root
"C:/Program~/PSL/RCA/Data/ProxyServer/dynamic"
 -dynamic-prefix
"/ups/RESOURCE"
 -dynamic-url
"http://localhost:3466"
 -dynamic-trace 0
 -dynamic-maxdays 0
 -dynamic-maxsizeMB 0
 -dynamic-proxy-host ""
 -dynamic-proxy-port ""
 -dynamic-proxy-user ""
 -dynamic-proxy-pass ""
 -dynamic-makeidx 1
 -dynamic-savetod -1
 -dynamic-savefreq 90
 -dynamic-largefile-size 0
 -dynamic-largefile-mindays 0
 -dynamic-freespace 10
 -dynamic-allow-shared-resource-purge 0
 }

```

```

END OF CONFIG

DO NOT REMOVE

rps::start
```

By default, after the installation:

1. HTTP front-end protocol is enabled; Stager front-end protocol (for TCP) is disabled.
2. Static cache is configured with entries from the install. The `-static-root` entry defines the data store location, and the `-static-host`, `-static-port` and `-static-user` entries define the static cache preload operation.
3. Dynamic cache operations are disabled.

## RPS.CFG Configuration Parameters

The table "Configuration file `rps.cfg` parameters" summarizes the parameters in the `rps.cfg` file and their default values. Parameters are listed in alphabetical order.

The configuration parameters in the `rps.cfg` file fall into three categories:

- **Front-end RCA agent communication options: `-httpd` (http) or `-stager` (TCP/IP)**  
By default, the Proxy Server is configured with the `-httpd` front end enabled. This uses the HTTP front end protocol to communicate with Sentient agents. When required, the `-stager` front-end option is also available to communicate with agents using TCP/IP.
- **Static cache settings**  
The group of `-static*` parameters configure the static cache and its TCP connection to the Configuration Server to obtain a preload resolution. Modify the `-static*` parameters in the `rps.cfg` file to change the values set during the installation or to set additional options, such as tracing.
- **Dynamic cache settings**  
The dynamic cache is not configured during the install. You must manually enable it (that is, change `-dynamic` from 0 to 1 in `rps.cfg`) and specify its options by editing the `rps.cfg` file. For more information, see "Enabling Dynamic Cache" in the *Sentient Enterprise Console Online Help*.

### Configuration File `rps.cfg` Parameters

| Parameter              | Default | Description                                                                                                                                                                                                                              |
|------------------------|---------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>-dynamic</code>  | 0       | When set to 1, the dynamic cache is enabled and an entry in <code>-dynamic-url</code> is required to specify the upstream host. When set to 0, the dynamic cache is disabled, and all <code>-dynamic*</code> parameters are disregarded. |
| <code>-dynamic-</code> | 0       | When set to 0 (recommended), safeguards against                                                                                                                                                                                          |

| Parameter                   | Default | Description                                                                                                                                                                                                                                                                                                                                                                                                      |
|-----------------------------|---------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| allow-shared-resource-purge |         | purging dynamic cache from a CSDB. Set to 1 to remove the safeguard.                                                                                                                                                                                                                                                                                                                                             |
| -dynamic-defdmn             |         | HTTP upstream URL (Configuration Server's) domain.                                                                                                                                                                                                                                                                                                                                                               |
| -dynamic-defcls             |         | HTTP upstream URL (Configuration Server's) class.                                                                                                                                                                                                                                                                                                                                                                |
| -dynamic-freespace          | 10      | <p>A percentage of the <code>-dynamic-maxsizeMB</code> to be left as free disk space after executing a size-based purge. The default is 10 %.</p> <p>When a size-based purge is triggered, the purge will delete files until the cache file size is reduced by this percentage below the maximum size.</p> <p>Used with <code>-dynamic-maxsizeMB</code>.</p>                                                     |
| -dynamic-largefile-mindays  | 0       | <p>Specifies the minimum days a 'large file' should be retained (during a size-based purge). Disabled when set to 0 (default).</p> <p>Used with <code>-dynamic-maxsizeMB</code> and <code>-dynamic-largefile-size</code>.</p>                                                                                                                                                                                    |
| -dynamic-largefile-size     | 0       | <p>Specifies the size of a 'large file' in bytes. To specify the size in MB, enclose in quotation marks, such as: "250 mb". When not equal to 0, files of this size or greater are excluded from deletion during the first pass of a size-based purge. Disabled when set to 0 (default)</p> <p>Used with <code>-dynamic-maxsizeMB</code>.</p>                                                                    |
| -dynamic-maxdays            | 0       | <p>The number of days to keep un-requested resources in the dynamic cache (if a file is regularly requested, it would never be deleted). Defines "aged" files for dynamic cache cleanup.</p> <p>Files that have not been requested for longer than this period are deleted from the cache whenever the dynamic index file is saved. (See <i>dynamic-savetod</i> and <i>-dynamic-savefreq</i> for frequency.)</p> |
| -dynamic-maxsizeMB          | 0       | Maximum size of dynamic cache. 0 indicates no maximum. To set a maximum size, enter the number of MB, such as 200 for 200 MB.                                                                                                                                                                                                                                                                                    |
| -dynamic-makeidx            | 1       | When set to 0, turns off making the dynamic cache idx file (no purge). A date-based purge of aged files is performed                                                                                                                                                                                                                                                                                             |

| Parameter           | Default                                                          | Description                                                                                                                                                                                                                                 |
|---------------------|------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                     |                                                                  | whenever the index file is saved. (See <i>dynamic-maxdays</i> ).                                                                                                                                                                            |
| -dynamic-prefix     | For Windows:<br>"/ups/RESOURCE"<br><br>For Linux:<br>"/RESOURCE" | HTTP upstream URL prefix (append <code>Domain.Class.OID</code> ).                                                                                                                                                                           |
| -dynamic-proxy-host | ""                                                               | To pass through an Internet proxy machine or a Firewall proxy when connecting to the Dynamic Cache Upstream Host machine, enter the Internet proxy's hostname or IP address.                                                                |
| -dynamic-proxy-pass | ""                                                               | To pass through an Internet proxy machine or a Firewall proxy when connecting to the Dynamic Cache upstream host machine, enter the connecting User's password.                                                                             |
| -dynamic-proxy-port | ""                                                               | To pass through an Internet proxy machine or a Firewall proxy when connecting to the Dynamic Cache Upstream Host machine, enter the Internet Proxy port number to use.                                                                      |
| -dynamic-proxy-user | ""                                                               | To pass through an Internet proxy machine or a Firewall proxy when connecting to the Dynamic Cache Upstream Host machine, enter the connecting User name.                                                                                   |
| -dynamic-root       | See description.                                                 | The fully-qualified location to store the dynamic cache. For example:<br><br>For Windows:<br>"C:/Program Files/PSL/RCA/Data/ProxyServer/dynamic"<br><br>For Linux:<br>"/opt/PSL/RCA/IntegrationServer/data/rps/dynamic"                     |
| -dynamic-savefreq   | 90                                                               | When <code>-dynamic-savetod</code> is disabled (-1), specifies how often (in minutes) to save index file and cleanup cache. The save index file process is skipped when there are no changes to be applied.                                 |
| -dynamic-savetod    | -1                                                               | Specifies time of day (hh:mm:ss) to save dynamic index file and cleanup cache each day. Overrides <code>-dynamic-savefreq</code> . Disabled when set to -1. The save index file process is skipped when there are no changes to be applied. |
| -dynamic-trace      | 0                                                                | When set to 1, information is recorded for diagnostic tracing.                                                                                                                                                                              |

| Parameter          | Default                                                                             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|--------------------|-------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -dynamic-url       | For Windows:<br>"http://localhost:3466"<br><br>For Linux:<br>"http://upstream:3466" | HTTP upstream URL (append <code>prefix</code> ). Replace upstream with the upstream host the Proxy Server makes a connection with for obtaining dynamic cache. The upstream host can be a Core Server or another Proxy Server.<br><br>Optionally, define multiple upstream hosts for fail-over support. Enclose all entries in one set of quotation marks. Space-separate each upstream host. For example:<br><br>"http://upstream1:3466<br>http://upstream2:3466" |
| -httpd             | 1                                                                                   | When set to 1, HTTP front end is enabled, supporting HTTP communication with Sentient agents. Do not change.                                                                                                                                                                                                                                                                                                                                                       |
| -httpd-prefix      | "/RESOURCE"                                                                         | The URL prefix registered to Tcl web server. Do not change.                                                                                                                                                                                                                                                                                                                                                                                                        |
| -httpd-trace       | 0                                                                                   | This parameter is reserved for future use.                                                                                                                                                                                                                                                                                                                                                                                                                         |
| -static-host       | RCS IP Address or hostname                                                          | Agent upstream host (example, Configuration Server). Used for initiating the Preloader connect. Replace with IP address of the upstream host.                                                                                                                                                                                                                                                                                                                      |
| -static-http-port  | 3466                                                                                | Agent upstream HTTP port for preloader.                                                                                                                                                                                                                                                                                                                                                                                                                            |
| -static-https-port | 443                                                                                 | Agent upstream HTTPS port for preloader.                                                                                                                                                                                                                                                                                                                                                                                                                           |
| -static-logsize    | ""                                                                                  | Specifies the size of the Preloader <code>connect.log</code> file in bytes. <code>connect.log</code> is in the <code>ProxyServer\logs\rps</code> directory.<br><br>When the logsize is reached, a backup file ( <code>.BAK</code> ) is created. By default, this file is <code>connect.bak</code> . If a backup file already exists, it is overwritten.                                                                                                            |
| -static-port       | 3464                                                                                | Agent upstream TCP port. Used during preload.                                                                                                                                                                                                                                                                                                                                                                                                                      |
| -static-proxy-host | ""                                                                                  | To pass through an Internet proxy machine or a Firewall proxy when connecting to the Agent upstream host during preload, enter the Internet proxy's hostname or IP address.                                                                                                                                                                                                                                                                                        |
| -static-proxy-pass | ""                                                                                  | To pass through an Internet proxy machine or a Firewall proxy when connecting to the Agent upstream host during the preload, enter the connecting User's password.                                                                                                                                                                                                                                                                                                 |



| Parameter           | Default                       | Description                                                                                                                                                                                                                        |
|---------------------|-------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -static-proxy-port  | ""                            | To pass through an Internet proxy machine or a Firewall proxy when connecting to the Agent upstream host during the preload, enter the Internet Proxy port number.                                                                 |
| -static-proxy-user  | ""                            | To pass through an Internet proxy machine or a Firewall proxy when connecting to the Agent upstream host during the preload, enter the connecting User name.                                                                       |
| -static-root        | <i>See description</i>        | The fully qualified location of the preloader's data store (IDMDATA). For example:<br><br>For Windows:<br>"C:/Program Files/PSL/RCA/Data/ProxyServer/static"<br><br>For Linux:<br>"/opt/PSL/RCA/IntegrationServer/data/rps/static" |
| -static-type        | agent                         | When set to agent, the static cache is populated by the Preloader.                                                                                                                                                                 |
| -static-ssl         | 0                             | When set to 1,the Preloader uses SSL.                                                                                                                                                                                              |
| -static-trace       | 0                             | When set to 1, information is recorded for diagnostic tracing.                                                                                                                                                                     |
| -static-use-datauri | 1                             | Enables preload using upstream URL.                                                                                                                                                                                                |
| -static-user        | <i>RPS_satellite_hostname</i> | Agent upstream identity on the CSDB in Primary.Policy.User (ZUSERID) used during preloading. The services connected to this user ID will be preloaded into the static cache on the Proxy Server.                                   |

## Editing the RPS.CFG File

To edit the `RPS.CFG` File:

Before modifying the `rps.cfg` file, stop the service for the Proxy Server. For starting and stopping the Proxy Server service on windows, see ["Starting and Stopping the Proxy Server service" on page 1424](#). For starting and stopping the Proxy Server service on Linux, see *Sentient Enterprise Installation and Upgrade Guide*.

1. Make your modifications to the `rps::init` section of the `rps.cfg` file.
2. After making your modifications, restart the service.

## Syntax Notes

- When the following parameters are disabled (that is, set to 0), all other options related to the parameter are disregarded:

```
-dynamic
-httpd
-stager
```

For example, if `-dynamic` is 0, the entire set of `-dynamic*` parameters are disregarded.

- Use quotation marks to enclose entries that include special characters or spaces. For example, the following `-dynamic-root` entry uses quotation marks to enclose the fully-qualified location of the dynamic cache store.

```
-dynamic-root "C:/Program Files/PSL/RCA/Data/ProxyServer/dynamic"
```

- Use slashes [/] to specify the paths for the `-dynamic-root` and `-static-root` parameters in both Windows and Linux environments. For example:

```
-dynamic-root "C:/Program Files/PSL/RCA/Data/ProxyServer/dynamic"
```

## Configuring the Dynamic Cache Parameters

**Note:** See the table "Configuration file `rps.cfg` parameters" for complete definitions of each parameter listed in the table "Dynamic Cache parameter summary".

Dynamic Cache parameter summary

| Objective                                                                                                                                                                                                                                                                                                                                                                             | RMS.CFG Parameters to Use                                                                                                                                                                                                                                   |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Specify an upstream host and HTTP port; either a Configuration Server enabled for HTTP downloads, or another Proxy Server. Specify multiple hosts for fail-over support.<br><br>Required.                                                                                                                                                                                             | <code>-dynamic 1</code><br><code>-dynamic-url</code><br><code>"http://localhost:3466"</code><br><br>or<br><br><code>-dynamic-url</code><br><code>"http://localhost:3466</code><br><code>http://upstream2:3466</code><br><code>http://upstreamN:3466"</code> |
| Specify where to store the dynamic cache on the Proxy Server.                                                                                                                                                                                                                                                                                                                         | <code>-dynamic-root</code>                                                                                                                                                                                                                                  |
| Save the index file that is used to track when files were last requested. A save also purges "aged" dynamic cache files not requested in a specific number of days. Schedule the index file save daily or every <i>nn</i> minutes.<br><br><b>Note:</b> A scheduled index file save (and thus the data-based purge) is skipped if there are no changes in the index file at that time. | <code>-dynamic-maxdays</code><br><code>-dynamic-makeidx</code><br><code>-dynamic-savefreq</code><br><code>-dynamic-savetod</code>                                                                                                                           |

| Objective                                                                                                                                                                                                                                                                                                                 | RMS.CFG Parameters to Use                                                                                                |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------|
| Set a maximum cache size. If the maximum cache size is exceeded, the least recently used files are deleted until the maximum size is reached.<br><br>Options:<br><br>1. Set a low-water mark (freespace) for a size-based purge.<br><br>2. Define 'large files' to exempt during the first pass of the size-based delete. | -dynamic-freespace<br><br>-dynamic-maxsizeMB<br><br>-dynamic-largefile-size<br><br>-dynamic-largefile-mindays            |
| Revise the safeguard against purging the dynamic cache stored on a shared resource, such as the Configuration Server.                                                                                                                                                                                                     | -dynamic-allow-shared-resource-purge                                                                                     |
| Set diagnostic tracing.                                                                                                                                                                                                                                                                                                   | -dynamic-trace                                                                                                           |
| Provide server, port, user, and password information to pass through an Internet Proxy or Firewall required to connect to the upstream host. Also, enable SSL protocol.                                                                                                                                                   | -dynamic-proxy-host<br><br>-dynamic-proxy-port<br><br>-dynamic-proxy-user<br><br>-dynamic-proxy-pass<br><br>-dynamic-ssl |
| Override the default prefix.domain, and class OID for obtaining resources from the upstream URL. Not normally needed.                                                                                                                                                                                                     | -dynamic-prefix<br><br>-dynamic-defdmn<br><br>-dynamic-defcls                                                            |

## Preloading Deployment Options

A Proxy Server's preload resolution is always performed on the upstream Core Server or full-service Satellite Server. However, there are several ways to configure the actual deployment of the static cache resources in an Sentient environment. These deployment options are set the same way that Sentient agents are configured to obtain their resources from the Proxy Server. For example, for multicast, this means associating a pre-configured MULTICAST instance with a Proxy Server User or Workgroup instance in the CSDB.

The configuration options for preloading the Proxy Server Static Cache include:

- Preloading using HTTP: By default, the preloading happens through HTTP on Satellite servers and Linux Proxy servers. The following parameters are set in the rps.cfg file by default:

-static 1

-static-root "<InstallDir>/Data/ProxyServer/static"

-static-trace 0

-static-type agent

-static-host {upstream\_server IP address}

```
-static-port 3464
-static-http-port 3466
-static-user <RPS_Satellite_User>
-static-use-datauri 1
• Preloading Using SSL (TCPS or HTTPS):
-static 1
-static-root "<InstallDir>/Data/ProxyServer/static"
-static-trace 0
-static-type agent
-static-host {upstream_server IP address}
-static-port 3464
-static-http-port 3466
-static-user <RPS_Satellite_User>
-static-use-datauri 1
• "Preloading Using Multicast" below
```

## Preloading Using Multicast

A set of Proxy Servers can obtain their static cache preloads from an RCA Multicast Server. In this configuration, the Proxy Servers receiving their preload resources are configured the same way that Sentient agents are configured to receive resources from a multicast server.

The Proxy Server preloads can also receive their resources using the dynamic windows feature of a multicast server, as discussed in the topic "[Preloading using Multicast with Dynamic Windows](#)" on page 1418.

To preload Proxy Servers using Multicast:

**Caution:** It is recommended that administrators become thoroughly familiar with the operation of the Multicast Server before using it for a Proxy Server preload.

1. For each Proxy Server to participate in the preload, use the CSDB Editor to create a POLICY.USER instance in the CSDB that matches the `-static-user` parameter in the `ProxyServer\etc\rps.cfg` file. By default the `-static-user` is set to: *RPS\_satellite\_hostname*.
2. Using the CSDB Editor, create a POLICY.WORKGROUP instance for the Proxy Servers to specify their desired state for the preload of the Static Cache. The procedure *Configuring a Proxy Server Workgroup for Preload* in the *Sentient Enterprise System Administrators Guide*, shows an example using the friendly name Proxy Preload Apps.
3. Configure a Multicast Server. The Multicast Server can be co-located with the Core Server or co-located with another Satellite Server.

4. Using the CSDB Editor, also create a multicast policy instance just for Proxy preloads. For example, follow these steps to create an instance named PRIMARY.POLICY.MULTICAST.MCPRELOAD.
5. Edit the MCPRELOAD instance attribute values.
  - Set ALTADDRM to the IP address of where the Multicast Server is installed.
  - Set CGMTDATE and CGMTTIME to specify a multicast session start date and time, in Greenwich Mean Time (GMT), so that the session is active when the proxy preload command is executed.

**Note:** The hours for the collection start time, CGMTTIME, are specified using a 24-hour (military) format.

The following figure illustrates a MCPRELOAD instance with CGMTDATE and CGMTTIME entries that initiate a multicast session for the Proxy preload on November 4, 2005 at 2 p.m. GMT.

**Caution:** The Proxy Preload command must be issued at the same time or shortly after the multicast collection session starts.

Database Tree View:

The Database Tree View shows a hierarchical structure of database objects. The 'MULTICAST' folder is expanded, showing its sub-objects. The 'MCPRELOAD' object is highlighted in blue.

- CLIENT
- MACHINE
- NOVADIGM
- OS
- PATCH
- PATCHMGR
- POLICY
  - Country / Region (COUN)
  - Departments (DEPT)
  - Machine Manufacturer (M)
  - Machine Models (MODEL)
  - Machine Roles (ROLE)
  - Machine Subnets (SUBNI)
  - Mobile Device Config (MB)
  - Multicast (MULTICAST)
    - \_BASE\_INSTANCE\_
    - MCDWPRELOAD
    - MCPRELOAD**
    - NJMAH
  - PDACONFIG (PDACONF)
  - Server Stagers (STAGER)
  - Users (USER)
  - Workgroups (WORKGRP)
- PRDMAINT
- SOFTWARE
- SYSTEM
- USAGE

Multicast class MCPRELOAD Instance Attributes:

| Name     | Attribute Description               | Value                    |
|----------|-------------------------------------|--------------------------|
| DOMAIN   | Domain Name                         | &{ZOBJDOMN}              |
| CLASS    | Class Name                          | &{ZOBJCLAS}              |
| INSTANCE | Instance Name                       | &{ZOBJNAME}              |
| MCAST    | Enable Multicast [Y/N]              | Y                        |
| MODE     | Broadcast or Multicast [M/B]        | M                        |
| MORDER   | Service Installation Order [B/A/S]  | S                        |
| ADDRESS  | Broadcast or Multicast Address      | 229.0.0.0                |
| PORT     | Broadcast or Multicast UDP port     | 9512                     |
| DELAYFP  | Delay After First Packet (mil sec)  | 40                       |
| DELAYBP  | Delay Between Packets (mil sec)     | 40                       |
| RESENDS  | Number of Resends                   | 1                        |
| STORE    | Backup Packets to Store for Resend  | 20                       |
| CGMTDATE | Collection Start Date (YYYYMMDD)    | 20051104                 |
| CGMTTIME | Collection Start Time GMT(HH:MM:SS) | 14:00:00                 |
| CWINDOW  | Duration of Collection Window (min) | 45                       |
| MDELAY   | Delay Before Multicast Start (min)  | 2                        |
| MWINDOW  | Duration of Multicast Window (min)  | 60                       |
| TTL      | Number of Router Hops               | 3                        |
| BYPASCON | User to Prime Multicast Server      |                          |
| ALTADDRM | Multicast Server IP                 | 208.119.233.106          |
| ALTPORTM | Multicast Server Port               | 3463                     |
| MINREF   | Minimum Num Clients Requesting File | 1                        |
| MINSIZE  | Minimum File Size for Multicast     | 1024                     |
| _ALWAYS_ | Manager REXX Method                 | SYSTEM.ZMETHOD.MULTICAST |

24 Multicast CLASS MCPRELOAD attribute(s) displayed

11/4/2005

7:19 PM

As specified in the previous figure, the collection phase (CWINDOW) will last 45 minutes, at which time the Multicast Server stops the collection process.

There will be a 2-minute delay (MDELAY) before the transmission begins. At the designated multicast start-time (CGMTTIME + CWINDOW + MDELAY = 14:47:00 GMT), the Multicast

Server (specified by ALTADDRM) starts multicasting the files that are on the list compiled from the various eligible Sentient agents.

6. Connect the Workgroup Policy instance for the Proxy preload Application to the POLICY.MULTICAST.MCPRELOAD instance.
7. Issue the Proxy preload command when the CGMTTIME and CGMTDATE apply.

**Note:** For information on viewing the results of the multicast session, see [Session Logs for a Preload Using Multicast](#).

## Preloading using Multicast with Dynamic Windows

The Multicast Server offers the ability to configure dynamic (collection and transmission) windows. This means for a group of Proxy Servers being preloaded, multiple multicast distributions can be active concurrently.

This topic gives three sample configurations for using the multicast dynamic windows feature to preload Proxy Servers. Only three MULTICAST instance attributes are used to implement dynamic windows: ADDRESS, CGMTDATE, and CGMTTIME.

**Caution:** Before using this feature, please review the *Multicast Dynamic Windows* topic in the *Sentient Enterprise Console Online Help*.

### Example

The following MULTICAST instance values can be used to initiate a multicast session (using up to 22 dynamic windows) that starts at 05:01:00 GMT *each day*. The empty value of CGMTDATE is used to specify a multicast session that runs once a day.

**Caution:** A Proxy Server preload must also be scheduled for this same time each day for it to participate in the multicast session.

```
ADDRESS = 235.0.0.000-021
CGMTDATE = empty
CGMTTIME = 05:01:00
CWINDOW = 45
MDELAY = 2
ALTADDRM = <multicast_server_IP>
```

During this daily multicast session, the multicast server can issue IP addresses from 235.0.0.000 through 235.0.0.021. The addresses in this range are used sequentially to accommodate overlapping windows for a group. When a given multicast session completes its transmission, the IP address is removed from the in-use list and is available once again.

The CWINDOW value of 45 specifies a collection window of 45 minutes.

## Session Logs for a Preload Using Multicast

On a Proxy Server receiving a preload from a Multicast Server, the preload session logs are found in the directory:

```
ProxyServer\logs\rps
```

The following logs trace the activity of the collection, requests, and received items in a multicast session:

```
connect.log
radreqst.log
radcrecv.log
```

**Note:** The `connect.log` for a Proxy Server preload includes the collection information found in the `radcllect.log` of an Sentient agent multicast session.

For details on the contents of these multicast session logs, see the *Sentient Enterprise Troubleshooting Guide*.

## Preloading Windows Installer Enabled Applications

There are two packages associated with each Windows Installer enabled application, the MSI package that includes the list of MSI Features, and the Administrator Control Point (ACP) package.

**Note:** An Administrative Control Point (ACP) is also known as a Windows Installer Administrative Installation Point (AIP).

The ACP package instance contains all the file resources that are required by a Windows Installer product. If you are using Proxy Servers or Staging Servers, the ACP package needs to be deployed to these servers but not installed on the Sentient agent computers. Only the MSI package needs to be installed on the Sentient agent computers. There are two models for doing this.

### Model 1: Create one Application (ZSERVICE) that includes a ZSTOP expression on the ACP Package

Create one Application (ZSERVICE) instance for both the MSI and ACP packages. Include a ZSTOP expression on the ACP package to prevent the Sentient agent computers from downloading it. Staging Servers and Proxy Servers ignore expressions, and download the ACP package.

**Note:** This model is considered the best practice and is applied automatically when you publish Windows Installer Applications using the Advanced publishing mode of the

RCA Administrator Publisher and apply the default Admin Install Point (AIP) options. For more information on publishing Windows Installer Enabled applications, see the *Sentient Enterprise Console Online Help*.

Use the following procedure if you need to apply the ZSTOP expression to an ACP package manually.

To enable an ACP package for preloading only:

1. Use the CSDB Editor to navigate to the ACP package that you want to preload.
2. Right-click on the ACP package and choose **Edit Instance** from the shortcut menu.
3. Click **ZSTOP000 Stop Resolution (000)** and type a value of 1.  
Typing a value of 1 prevents Sentient agent computers from resolving the ACP package, but allows the Proxy Server or Staging Server to preload the ACP package.
4. Click **OK**.
5. Click **Yes** to confirm the change.
6. Connect the ACP package and the MSI package to the same service.

## Model 2: Create separate Applications (ZSERVICES) for MSI Packages and ACP Packages

An alternative model is to create separate Application (ZSERVICE) instances for the MSI Package and the ACP Package. Deploy the MSI Package to the Sentient agent computers and the ACP Package to the Staging Server and Proxy Servers.

**Caution:** It is *not* recommended naming ACP packages with a common suffix for the purpose of connecting *all* of the ACP packages to a single service and deploying this service to your Staging Servers and Proxy Servers. This method presents unnecessary overload on your infrastructure.

## Configuring HTTPS data download from Linux Proxy

Make sure that your Linux Proxy server is SSL enabled and HTTPS is configured for Sentient agent requests. To download HTTPS data from Linux proxy to the Sentient agent, perform the following steps:

1. On a Proxy Server machine, use a text editor to modify the `rps.cfg` file located in the `ProxyServer\etc` directory. Set the `-static-ssl` parameter to 1, and `-stager` parameter to 0.  

```
rps::init {

 -stager 0
 -static-ssl 1
 -static-port <SSL port>
```



Where *<SSL port>* is the SSL port number on the secured Proxy Server established in Step 1. The default SSL port number is 444.

2. To apply the new `rps.cfg` file settings, restart the RCA Proxy Server service.
3. Add the following lines of code to the `httpd.rc` file located in the `/opt/PSL/RCA/IntegrationServer/etc` directory:

```
Overrides Config {
 PORT 3466
 HTTPS_PORT 443
 SSL_CERTFILE ""
 SSL_KEYFILE ""
 HTTPS TRUE
}
```

where

- `PORT` is the port on which the web service listens, normally 3466.
- `HTTPS_PORT` is the HTTPS port on which the data is downloaded. The only supported port is 443.
- `SSL_CERTFILE` is the SSL certificate file.
- `SSL_KEYFILE` is the SSL certificate key file.
- `HTTPS` is set to `TRUE` to enable HTTPS data download from the Linux proxy. The default value is `AUTO`.

The other parameters and their default values in the `httpd.rc` file are given below:

```
DOCRROOT [file join $Config(HOME) html]
HOST [info hostname]
IPADDR 0.0.0.0
WEBMASTER support.mail@psl.com
UID 50
GID 100
LIMIT 1024
CONFIG $Config(NAME).rc
LIBRARY {}
DEBUG FALSE
LOG_LEVEL 3
LOG_DIR logs
NAME $Config(NAME)
```

|              |                                         |
|--------------|-----------------------------------------|
| HTML_MAIN    | main.tsp                                |
| HTML_BG      | {#FF9900}                               |
| PROTO        | http                                    |
| HTTPS_HOST   | {}                                      |
| HTTPS_IPADDR | 0.0.0.0                                 |
| SSL_CADIR    | \$Config(ETC)/CACertificates            |
| SSL_CAFILE   | \$Config(ETC)/CACertificates/cacert.pem |
| SSL_CIPHERS  | {}                                      |
| SSL_REQUEST  | 0                                       |
| SSL_REQUIRE  | 0                                       |
| USE_SSL2     | 1                                       |
| USE_SSL3     | 1                                       |
| USE_TLS1     | 0                                       |
| SHOW_CONFIG  | 1                                       |

**where**

- **DOCROOT** is the folder name where the documents are stored.
- **HOST** is the host name of the web server.
- **IPADDR** is the host IP address where the web service listens.
- **WEBMASTER** is the support mail address.
- **UID** is the user ID.
- **GID** is the group ID.
- **LIMIT** is the buffer limit.
- **CONFIG** is the configuration file name. The default value is `$Config(NAME) . rc`.
- **LIBRARY** is the list of additional tcl libraries to be added, if required.
- **DEBUG** is to debug trace changes. The default value is `FALSE`.
- **LOG\_LEVEL** is the log level. The default log level is 3.
- **LOG\_DIR** is the directory where the log files are stored. The default folder is `logs`.
- **NAME** is the name of the web service.
- **HTML\_MAIN** is the default home page.
- **HTML\_BG** is the background color. The default background color is `#FF9900`.
- **PROTO** is the protocol. The default protocol is `HTTP`.
- **HTTPS\_HOST** is the host name of the secured web server.
- **HTTPS\_IPADDR** is the IP address of the secured web server.

- `SSL_CADIR` is the directory where the SSL certificates are stored.
- `SSL_CAFILE` is the SSL Certificate Authority file name.
- `SSL_CIPHERS` is the cipher specification name.
- `SSL_REQUEST` is to perform the certificate signing request to check if the certificate is from the trusted certificate authority. The default value is 0.
- `SSL_REQUIRE` is the expression `%{SSL_CIPHER_USEKEYSIZE} >= 128`. By default, the value is 0.
- `USE_SSL2` , `USE_SSL3` , `USE_TLS1` are the types of SSL connects. By default, SSL2 and SSL3 connects are enabled.
- `SHOW_CONFIG` is to display the configuration details. The default value is 1.

## Enhancing Proxy Server Performance

Input Output Completion Port (IOCP) can be used to enhance the Proxy Server network throughput on Windows platform.

To enable IOCP, add the following lines of code to the `RCA-PS.rc` file:

```
Overrides Httpd {
 iocp 1
 buffersize 64100
 backlog 200
 sendcap 25
 recvmode "zero-byte"
}
```

where,

- `iocp` is set as 1 to enable IOCP.
- `buffersize` is the buffer size used for sending and receiving data. Use the following formula to determine buffersize based on your network environment:  
$$\text{Buffersize} = \text{Multiple of } (\text{MTU} - \text{IP Header Size} - \text{TCP Header Size})$$
where, *MTU* is the Maximum Transmission Unit.

Consult your network administrator to calculate exact buffersize for your environment. For example, with an MTU of 1500, IPv4 header size of 20 bytes, and TCP header size of 20 bytes, and without TCP timestamps, the buffersize is a multiple of  $(1500-20-20)=1460$ , such as  $44*1460=64240$ .

- `backlog` is the maximum number of client connections that are accepted in the queue before the request is passed to httpd. Each client connection uses 500 bytes of non-paged pool in the Operating System. *Recommended* value is 200. You can increase the backlog value if you observe an increase in the number of dropped connections.
- `sendcap` is the number of concurrent send operations allowed per client connection. *Recommended* value is 25.
- `recvmode` is the mode used to receive data. The *only* supported mode is "zero-byte". "zero-byte" mode is suitable for proxies that send data.

## Proxy Server Administration

There may be special circumstances involved in your Proxy Server implementation. This chapter explores these possible situations:

- Starting and stopping the Proxy Server.
- Populating the Proxy Server before any Sentient agents try to install software (preloading).
- Purging the Proxy Server dynamic cache

## Starting and Stopping the Proxy Server service

To start and stop the Proxy Server in Sentient Satellite server, start and stop the RCA Proxy Server service on your computer.

On Linux platforms, the RCA Proxy Server service does not start automatically after installation. you *must* start the service manually after the installation. For more information, see *Sentient Enterprise Installation and Upgrade Guide*.

For this example, Windows 2003 Server is used.

To start the Proxy Server service:

1. From your Windows Desktop, click **Start > Control Panel > Administrative Tools > Services**.  
The Service Control Manager window opens.
2. Right-click on the service **RCA Proxy Server** service and select **Start** from the shortcut menu.

The Proxy Server is now running on your computer.

To stop the Proxy Server service:

1. From your Windows desktop, click **Start > Programs > Control Panel > Administrative Tools > Services**.  
The Service Control Manager window opens.
2. Right-click on the service **RCA Proxy Server** and then select **Stop** from the shortcut menu.

The Proxy Server has been stopped.

## Purging the Proxy Server Dynamic Cache

The following topics explain how various configuration parameters can be set to purge the Proxy Server dynamic cache:

- ["The Date-Based Purge of the Dynamic Cache" on the next page](#)
- ["Saving the Index File" on the next page](#)
- ["Specifying a Size-Based Purge of the Dynamic Cache" on page 1426](#)

## The Date-Based Purge of the Dynamic Cache

Every time a file is requested from the Dynamic Cache, the request date is recorded and maintained. If a file has not been requested in a pre-defined maximum number of days, it is deleted whenever a date-based purge occurs. Use the `-dynamic-maxdays` parameter in `rps.cfg` to specify the maximum number of days to keep the files that have not been requested in the dynamic cache. The default is 7 days.

A date-based purge can be triggered whenever the dynamic cache index file is saved. For more information, see ["Saving the Index File" below](#).

## Saving the Index File

An index file is maintained to keep track of when files in the Dynamic Cache were last used. This index file is saved frequently to update the "last-used" date in the dynamic cache files. Whenever the index file is saved, a date-based purge of the dynamic cache also takes place to cleanup "aged" files.

The schedule for saving the index file is established using either the `-dynamic-savetod` or `-dynamic-savefreq` parameters. Use the `-dynamic-savefreq` parameter to schedule the index file save process every *nn* minutes.

**Caution:** For a date-based purge to occur, the index file save process must run. The index file save process is skipped when there are no changes to be applied to it. Therefore, if a dynamic cache has had no resources requested of it or added to it since a previous save, the index file save is skipped and the date-based purge is also skipped.

To use either of these options, first ensure the following dynamic cache parameters are specified:

```
-dynamic 1
-dynamic-makeidx 1
-dynamic-maxdays <max days to hold unrequested files>
```

- Use the `-dynamic-savetod` parameter to set a daily schedule for saving the dynamic index and purging the dynamic cache. A `-dynamic-savetod` entry overrides a `-dynamic-savefreq` entry. Valid values are `-1` (disabled) or `hh:mm:ss`. Seconds may be omitted. Example: To set a time-of-day for the purge to occur daily, use the `-dynamic-savetod` parameter with the time specified in the format `hh:mm:ss`. For example, the following set of parameters will automatically run a purge of the dynamic cache each day at 3:00 AM to delete files that have not been requested for seven days.

```
-dynamic 1
-dynamic-makeidx 1
-dynamic-maxdays 7
-dynamic-savetod 3:00
```

- If `-dynamic-savetod` is not specified, use the `-dynamic-savefreq` parameter to specify how often (in minutes) to save the index file and purge the dynamic cache. Example: The following set of parameters automatically runs a purge of the dynamic cache every 120 minutes for files that have not been requested for seven days.

```
-dynamic 1
-dynamic-makeidx 1
-dynamic-maxdays 7
-dynamic-savefreq 120
```

## Specifying a Size-Based Purge of the Dynamic Cache

Use the following options to automatically have files deleted from the dynamic cache to keep its size in check.

- Automatically run a size-based purge when the maximum size is exceeded.  
The maximum size of the dynamic cache is specified in the `-dynamic-maxsizeMB` parameter in `rps.cfg`. When this maximum size is exceeded, a size-based purge of the dynamic cache runs automatically, purging the least recently used files until the target size is reached.

For example, the following entries specify an automatic purge of the least recently used files if the dynamic cache exceeds 300 MB. The purge process deletes the least recently used files until the cache is below the maximum size, or, to the size required by the new `-dynamic-freespace` parameter (next feature).

```
-dynamic 1
-dynamic-maxsizeMB 300
```

- Define a "freespace" amount to be available after a size-based dynamic cache purge  
When a size-based purge is triggered, you may specify a `-dynamic-freespace` option to purge the dynamic cache down to a low-water mark (as a percentage of the maximum file size). This allows you to bring the cache size down to a predefined manageable size whenever it reaches the maximum file size. Use this option to eliminate repetitive calls to the purge process if upstream downloads occur frequently.

To specify the `-dynamic-freespace` parameter, define a percentage of free space required upon completion of a size-based purge. For example: the following options will trigger a purge of the dynamic cache when it reaches 300 MB, and will purge the least recently used files until the dynamic cache size is more than 10% below 300, or under 270 MB.

```
-dynamic 1
-dynamic-maxsizeMB 300
-dynamic-freespace 10 <percentage below the maxsizeMB after
the purge>
```

- Exempt large files from the first-pass of a size-based purge  
Normally, when a size-based purge takes place, the least recently used files are purged from the dynamic cache first. If large files are purged and then later requested, an undue load could be placed on the network. Use the following two "large file" parameters to allow the purge process to skip over the files that meet these large file criteria during an initial purge.

```
-dynamic-largefile-size <defines minimum large file size, in
bytes>
-dynamic-largefile-mindays <defines minimum days large files stay
in cache>
```

Example: The following entries define large files as "25 MB" and the minimum number of days to remain in the cache as 15. So, the first pass of the purge will exempt files 25 MB or larger

that have remained in the cache less than 15 days, even if they meet the normal "least recently used" criteria.

|                            |         |
|----------------------------|---------|
| -dynamic                   | 1       |
| -dynamic-maxsizeMB         | 300     |
| -dynamic-freespace         | 10      |
| -dynamic-largefile-size    | "25 MB" |
| -dynamic-largefile-mindays | 15      |

If, after purging the non-large files, the desired (reduced) cache size is not achieved, then the purge process will be repeated without regard to the large file settings.





# Chapter 16

---

## SSL Configuration

### Configuring SSL

**Note:** If you are already creating certificates in your environment with existing tools, you need not follow instructions in this chapter.

### The Certificate Generation Utility

For testing, a **Certificate Generation Utility** has been provided. This utility makes it easy to create self-signed certificates for testing. It will be used in this chapter to demonstrate the process for setting up SSL for Sentient.

**Caution:** This utility is intended for testing purposes *only* and should **not** be used in a production environment.

Before using this utility, consider the following:

- The Certificate Generation Utility is **not** a supported Sentient Enterprise product.
- The Certificate Generation Utility is provided *free of charge*.
- The Certificate Generation Utility is used at *your own discretion*; Technical Support **will not** address any issues regarding its use or functionality.

**Note:** **Certificate Generation Utility** can generate certificates on Windows platforms only. However, after the certificates are generated on a Windows system, the same can be copied over to and used on Linux platforms.

### Locating the Certificate Generation Utility

The Certificate Generation Utility can be found at the following location:

`<InstallDir>\Tools`

### Setting up a Certificate

The first step required to set up SSL is to make sure that each system that will be authenticated has a private key and a signed certificate. If you already have a private key for this system, you can use it to generate a certificate.

## Using an Existing Private Key

If you already have a private key in PEM file format, follow these steps:

1. In the `Tools\servers` directory, create a new directory named `hostname`. In this case, `hostname` is the name of the server for which a signed certificate is to be created. For example:

```
Tools\servers\cmserver1
```

2. Copy your private key PEM file into the directory that you just created.
3. Rename the PEM file that you just copied as follows: `hostname-prvkey.pem`. For example:

```
Tools\servers\cmserver1\cmserver1-prvkey.pem
```

## Generating a Signed Certificate

This section provides instructions for creating signed certificates that will be used for SSL configuration. There are three ways that you can generate a signed certificate:

- *Self-signing*  
This is the most convenient but least secure of the three options. Use this strictly for testing.
- *Via a generated CA*  
This option is more secure than self-signing, but not secure for a production environment. It creates a new CA with the parameters that you specify, and this new CA signs the certificate.
- *Via a trusted CA*  
This is the most secure option of the three. In a production environment, be sure to use certificates that have been signed by a trusted CA.

This task will use the Certificate Generation Utility to demonstrate each of these three options.

## Server Names

When you generate a certificate or a certificate request, you must specify the server name. You can use the simple host name (for example, `cmserver1`) or the fully qualified host name (for example, `cmserver1.mycorp.com`).

For MDM feature to work as expected, you must specify the fully qualified host name in small case (for example, `cmserver1.mycorp.com`).

The name that you specify should match the name that will be used in the URL when this server is accessed.

## Option 1: Generating a Self-Signed Certificate

1. From the `certificate_mgmt` directory, run the following command:  

```
cert_mgr create self
```

2. Provide the following information at the prompts:

| Parameter                    | Example              |
|------------------------------|----------------------|
| Server name (becomes the CN) | cmserver1.mycorp.com |
| Country name                 | US                   |
| State or province name       | California           |
| Locality Name                | Sacramento           |
| Organization name            | Mycompany            |
| Organizational unit name     | IT                   |

This information is used to create the **Distinguished Name (DN)** for the certificate. The DN is a unique identifier that is used to provide a name that is unique to the certificate. The DN is derived from the **Common Name (CN)** of the server and the other parameters that you specify.

The components of the DN, including the CN, are visible in the `cert.txt` file, as shown here:

```
Subject: C=US, ST=CA, L=Sacramento, O=MyCompany, OU=IT,
CN=cmserver1
```

**Note:** It is important that the CN part of the certificate's DN be the same as the server's host name. This is vital to the client trusting that it is communicating with the expected host.

After the utility finishes, the server certificate, private key, and related files are located in the `certificate_mgmt\servers\hostname` directory. For example, if the server name entered is `cmserver1`, the following files are generated:

```
certificate_mgmt\servers\cmserver1\cmserver1-cert.pem
certificate_mgmt\servers\cmserver1\cmserver1-cert.txt
certificate_mgmt\servers\cmserver1\cmserver1-prvkey.pem
certificate_mgmt\servers\cmserver1\cmserver1-signer.pem
certificate_mgmt\servers\cmserver1\cmserver1-signer.txt
certificate_mgmt\servers\cmserver1\cmserver1-cert.rnd
certificate_mgmt\servers\cmserver1\cmserver1-keystore.txt
certificate_mgmt\servers\cmserver1\cmserver1-keystore.jks
certificate_mgmt\servers\cmserver1\cmserver1-truststore.txt
certificate_mgmt\servers\cmserver1\cmserver1-truststore.jks
```

**Note:** The `keystore` and `truststore` files are generated only when the `JAVA_HOME` environment variable points to a Java runtime environment (JRE).

After you have verified that your files were correctly generated, see "Enabling SSL" in the *Sentient Enterprise Console Online Help*.

**Caution:** Self-signed certificates are adequate for testing purposes *only*; they should **not** be used in a production environment.

## Option 2: Generating a Re-usable Certificate Signed by a Generated CA

1. From the `certificate_mgmt` directory, run the following command.  
`cert_mgr create signed`
2. Provide the following information at the prompts.

| Parameter                    | Example              |
|------------------------------|----------------------|
| Server name (becomes the CN) | cmserver1.mycorp.com |
| Country name                 | US                   |
| State or province name       | California           |
| Locality Name                | Sacramento           |
| Organization name            | Mycompany            |
| Organizational unit name     | IT                   |

**Note:** The first time you run this command, you will also be prompted for information about the certificate authority (CA) that you are generating. On subsequent runs, it will not prompt you.

The utility generates two sets of files in this case.

- The first set consists of the server certificate and related files, which are located in the `certificate_mgmt\servers\hostname` directory (see the description of the files under "Option 1: Generating a Self-Signed Certificate" on page 1430).
- The second set consists of the CA files. These are located in the `certificate_mgmt\ca` directory.

For example, if the server name is specified as `cmserver1`, the following files are generated:

```
certificate_mgmt\servers\cmserver1\cmserver1-cert.pem
certificate_mgmt\servers\cmserver1\cmserver1-cert.txt
certificate_mgmt\servers\cmserver1\cmserver1-prvkey.pem
certificate_mgmt\servers\cmserver1\cmserver1-signer.pem
certificate_mgmt\servers\cmserver1\cmserver1-signer.txt
certificate_mgmt\servers\cmserver1\cmserver1-cert.rnd
```

```
certificate_mgmt\servers\cmserver1\cmserver1-keystore.txt
certificate_mgmt\servers\cmserver1\cmserver1-keystore.jks
certificate_mgmt\servers\cmserver1\cmserver1-truststore.txt
certificate_mgmt\servers\cmserver1\cmserver1-truststore.jks
certificate_mgmt\ca\ca.rnd
certificate_mgmt\ca\ca-cert.pem
certificate_mgmt\ca\ca-prvkey.pem
certificate_mgmt\ca\ca-index.txt
certificate_mgmt\ca\ca-index.txt.attr
certificate_mgmt\ca\ca-index.txt.old
certificate_mgmt\ca\ca-serial
certificate_mgmt\ca\ca-serial.old
```

**Note:** When you use the signed option, the Signing Authority Certificate is copied from the `certificate_mgmt\ca` directory. If the `certificate_mgmt\ca\ca-cert.pem` file already exists, that file will be used. Otherwise, it will be created on the first run and used for generating subsequent certificates.

**Note:** The `keystore` and `truststore` files are generated only when the `JAVA_HOME` environment variable points to a Java runtime environment (JRE).

After you have verified that your files were correctly generated, see "Enabling SSL" in the *Sentient Enterprise Console Online Help*

**Caution:** This method of generating signed certificates is adequate for testing purposes only and should not be used in a production environment.

## Option 3: Generating a Certificate Signed by a Trusted CA

These steps show you how to use the Certificate Generator Utility to generate a private key and certificate request that you can then send to a trusted CA. This might be an external CA, such as Verisign or Thawte, or a CA that your company or institution owns and administers.

1. From the `certificate_mgmt` directory, run the command  
`cert_mgr create request`
2. Provide the following information at the prompts:

| Parameter                    | Example              |
|------------------------------|----------------------|
| Server name (becomes the CN) | cmserver1.mycorp.com |

| Parameter                | Example    |
|--------------------------|------------|
| Country name             | US         |
| State or province name   | California |
| Locality Name            | Sacramento |
| Organization name        | Mycompany  |
| Organizational unit name | IT         |

After the utility finishes, the certificate request, private key, and related files are located in the `certificate_mgmt\servers\hostname` directory. For example, if the server name is specified as `cmserver1`, the following files are generated:

```
certificate_mgmt\servers\cmserver1\cmserver1.rnd
certificate_mgmt\servers\cmserver1\cmserver1-prvkey.pem
certificate_mgmt\servers\cmserver1\cmserver1-request.pem
certificate_mgmt\servers\cmserver1\cmserver1-request.txt
```

3. Request a signed certificate by sending the `hostname-request.pem` to your signing authority.

**Note:** Be sure that the server certificate that is purchased is a **base-64 encoded x.509** certificate. This is typical for certificates that are generated for the Apache Freeware (ModSSL or OpenSSL) Server.

4. When you receive this signed certificate from your signing authority, paste it into the `servers\hostname\hostname-cert.pem` file.
5. Paste the Signing Authority Certificate (must be in PEM format) into the `servers\hostname\hostname-signer.pem` file.

You now have a private key, a signed certificate, and the signing authority certificate files that are needed for product configuration.

## Additional Information about Certificates and Keys

This section provides information about installing the private key file.

### Installing the Private Key File

The Certificate Generation Utility also generates a private key in the form of the following PEM file:

```
hostname-prvkey.pem.
```

To install the private key, place this file in the appropriate directory on the server. See "Enabling SSL" in the *Sentient Enterprise Console Online Help* for the specific location of this directory for each type of Sentient server.

If you open the private key file with a text editor, the contents will look similar to the following:

```
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4, ENCRYPTED
DEK-Info: DES-CBC, 6EC0947550541AAB

1MV8Y4rkywlYn30yUB5ULtKLfj0YSzX+KZvxCeuw+9x95x1Ikvej4b8iBDuEOaTR
MIIDZTCCAs6gAwIBAgIBADANBgkqhkiG9w0BADQFADCBhDELMAkGA1UEBhMCVVMx
MzMxNDJaMIGEMQswC7YDVQQGEwJVUzELMAkGA1UECBMCQ0ExEDAQBgNVBACTB0Zy
ZW1vbnQxDzANBgNVBAoTB1N5Z2F0ZTEQMA4GA1UECjxMHQWx0dm1ldzEQMA4GA1UE
H10kihMe0Ny94uj8a6ccMJ+1kRj2grVmaw8tJi+6G76NXhvZvwumfHZMtnhKUKth
Mf3XLtUkz1z5LqoVJzuDoLQVcm7Ddx0iff+FLwRhsj153KQqoRYucL0opirXYc6R
8T+XMo3tkd4q=
-----END RSA PRIVATE KEY-----
```

In order to maintain compatibility with industry standards, The RSA crypto-system method of obtaining certificate requests has been adopted. The RSA crypto-system is a public key crypto-system that offers encryption and digital signatures (authentication). In the private key shown above, the key type (**RSA**) is indicated at the beginning and end of the file.

## Command Line Options for the Certificate Generation Utility

The tables [Options for cert\\_mgr create](#) and [Options for cert\\_mgr import](#) describe the options that can be used with the `cert_mgr create` and `cert_mgr import` commands that are described in Chapter 2, "Configuring SSL" on page 1429.

### Options for cert\_mgr create

| Option      | Description                                                                                                                    | Default                                                                         |
|-------------|--------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------|
| -hostname   | Host name of the server for which you will create the certificates.                                                            | Simple host name of the system on which you are running <code>cert_mgr</code> . |
| -trustpass  | The password for the truststore.                                                                                               | changeit                                                                        |
| -rndbytee   | Size of the random bytes when creating the random file that will be used to create the private key for the server certificate. | 2048 bytes                                                                      |
| -keysize    | Size of the server's private key in bits.                                                                                      | 1024 bits                                                                       |
| -keypass    | The password for the server's certificate when it is added to the keystore.                                                    | secret                                                                          |
| -days       | The number of days the server's certificate will be valid.                                                                     | 9999 days                                                                       |
| -carndbytes | Same as <code>rndbytes</code> , but for the CA.                                                                                | 2048 bytes                                                                      |
| -cakeysize  | Same as <code>keysize</code> , but for the CA.                                                                                 | 1024 bytes                                                                      |
| -cadays     | Same as <code>days</code> , but for the CA.                                                                                    | 9999 days                                                                       |

### Options for cert\_mgr import

| Option      | Description                                                                                                                                                   | Default                                                                         |
|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------|
| -hostname   | Host name of the server for which you will create the certificates.                                                                                           | Simple host name of the system on which you are running <code>cert_mgr</code> . |
| -signedcert | The fully qualified path and file name of the signed certificate that was returned by the CA.                                                                 |                                                                                 |
| -signercert | The fully qualified path and file name to the certificate of the signing CA.<br><br>Used when importing a certificate via the Certificate Generation Utility. |                                                                                 |



# Chapter 17

## Multicast Transmission

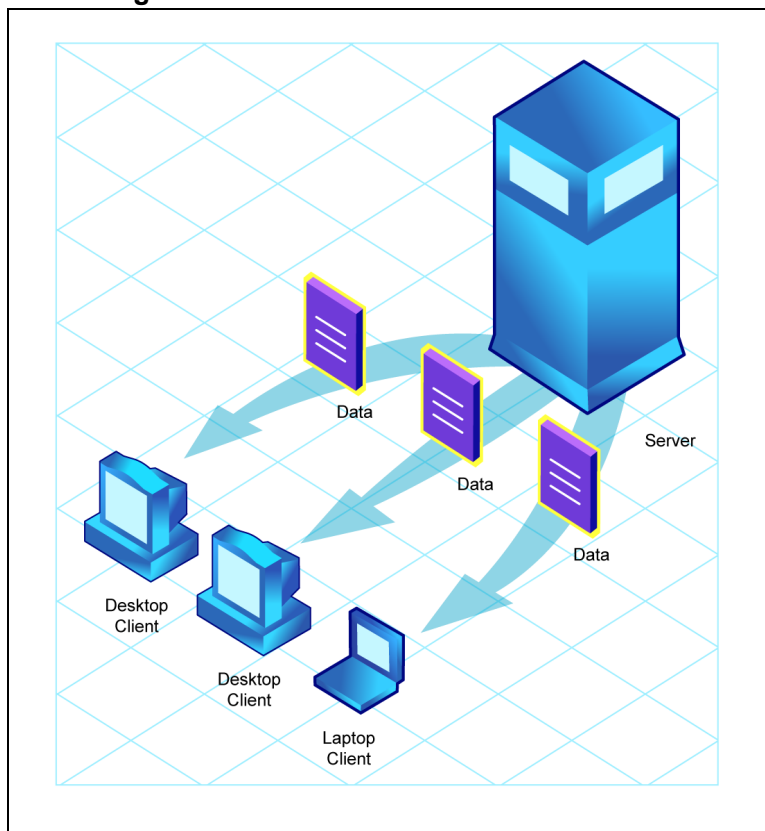
Most multicast utilities are designed to provide the simplest delivery of a payload that has been statically composed of all files and components for all possible recipients. In this model, every receiver is forced to take all resources that are bundled into that payload.

The Sentient multicast option allows the collection of the sets of resources that are needed by only those receivers that are eligible to participate in a specific multicast transmission. This means that only data required that is by the participants is sent, and the participants retrieve only the data that they have requested.

## Unicasting vs. Multicasting

The standard method of data transmission is **unicasting** (see figure [Unicasting Scenario](#)). In a unicast scenario, a server communicates with multiple agents individually and at different times, and transmits data to each during its communications session.

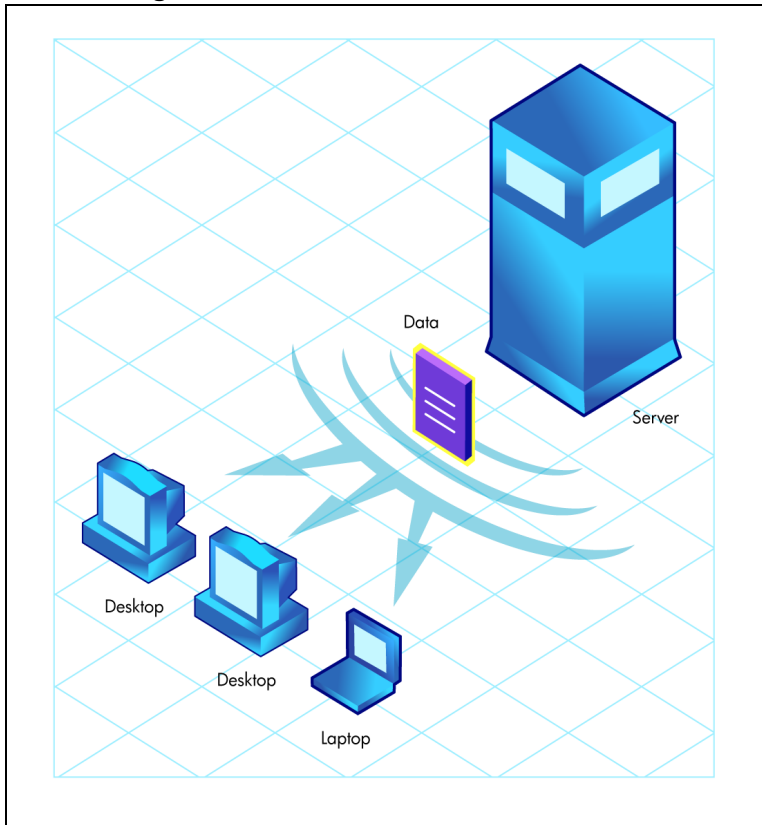
### Unicasting Scenario



Since the server must repeatedly transmit identical data, the unicast method is very time-consuming.

**Multicasting** is a technique that allows the simultaneous transmission of a data stream to many receivers (see figure [Multicasting Scenario](#)). The receivers identify themselves as “interested parties” by joining a logical group, using the **Internet Group Membership Protocol (IGMP)**.

### Multicasting Scenario



## Benefits of Multicasting

Multicasting provides the following benefits:

- It maximizes the use of network bandwidth by transmitting a single data-stream to multiple agents, simultaneously.
- It saves resources on the server by not having to set up separate client sessions and then repetitively transmit data individually to each client.

## Flexible, Centralized Configuration

In many multicast implementations, the logistics of configuring receivers in order to synchronize them with data transmitters becomes very labor-intensive or, for a large number of receivers, extremely restrictive. The administrator must ensure that the receivers are listening at the time the multicast transmission takes place and that the appropriate multicast address has been configured for all listeners. Altering these parameters on a large number of machines, and orchestrating the synchronization of sender and receivers on short notice, can be extremely difficult, if not impossible.

The configuration that is required in order to setup Sentient multicast distribution is performed centrally by the Sentient administrator. The Sentient agents obtain all required multicast configuration parameters in the framework of the standard Sentient agent connect. The parameters can be changed regularly or as needed, and the Sentient agent connect process can be started using any of the standard options that allow end user scheduled, or remotely invoked, initiation.

To further optimize its multicast capabilities, Sentient provides the ability (using the standard administration tools that are provided with the product) to associate users that have similar requirements into separate multicast groups (for example, file servers and desktops).

## Increasing Network Efficiency

The Sentient multicasting utility enables you to maximize the functionality of your existing network bandwidth. This is done by configuring the Multicast Server to concurrently transmit a single data-stream to multiple Sentient agents.

Each Sentient agent can distinguish between the files and components that are being delivered in the multicast stream, and retrieve only those that it needs. As a result, transfer time is minimized, and the Sentient agent's storage and processing resources are conserved because they are not forced to process and store superfluous data.

## Guaranteed Delivery

The Multicast Server uses existing Sentient functionality to determine whether all of the required resources have been delivered in the multicast phase. It then uses the standard protocol—the Sentient agent connect—to deliver any unsatisfied requirements. This protects against the “thrashing” that can occur in broadcast/multicast type protocols when the quality of the communications channel is poor. The Multicast Server can be tuned to the network characteristics using a buffering/transmit scheme, and a programmable time delay between packets.

**Note:** The guaranteed delivery feature is specific to Windows operating systems.

The results of a multicast session are reported to the Sentient Configuration Server Database (CSDB). For more information on how this is accomplished and the benefits of this feature, see “Multicast Results for Reporting” in the *Sentient Enterprise Console Online Help*.

## Multicast Process

During the multicast process, the Sentient agent interacts with the Configuration Server and the Multicast Server. The multicast process is defined by two time windows: the **preparation window** and the **distribution window**. (See the figure “[Multicast Time Windows and Phases](#)” on the next page.) The time windows are then subdivided into four **phases**, during which the Sentient agent interacts with the Configuration Server and the Multicast Server in order to:

- Determine which files are needed.
- Collect the requirements for the necessary files.

- Transmit the files.
- Report the results of the transmission.

### Multicast Phases

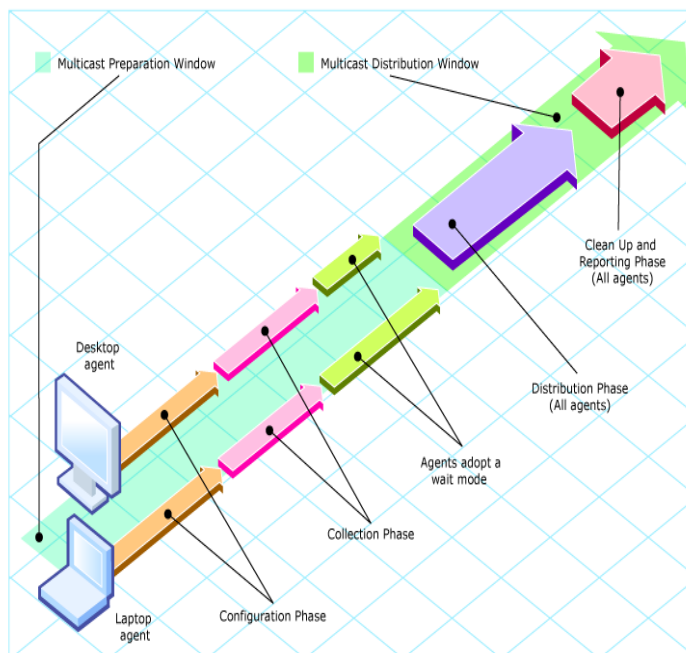
The multicast phases involve interaction (at various times) between the Sentient agent, the Configuration Server, and the Multicast Server. The Sentient agent is involved in all four phases of the process, whereas the Multicast Server and the Configuration Server are active only during some phases. This section details these phases, as well as each component's role in the process. The four phases of the multicast process are:

The four phases of the multicast process are:

- Configuration Phase
- Collection Phase
- Multicast Distribution Phase
- Clean up and Reporting Phase

**Note:** The parameters of these phases are configured in multicast instances in the Configuration Server Database.

### Multicast Time Windows and Phases



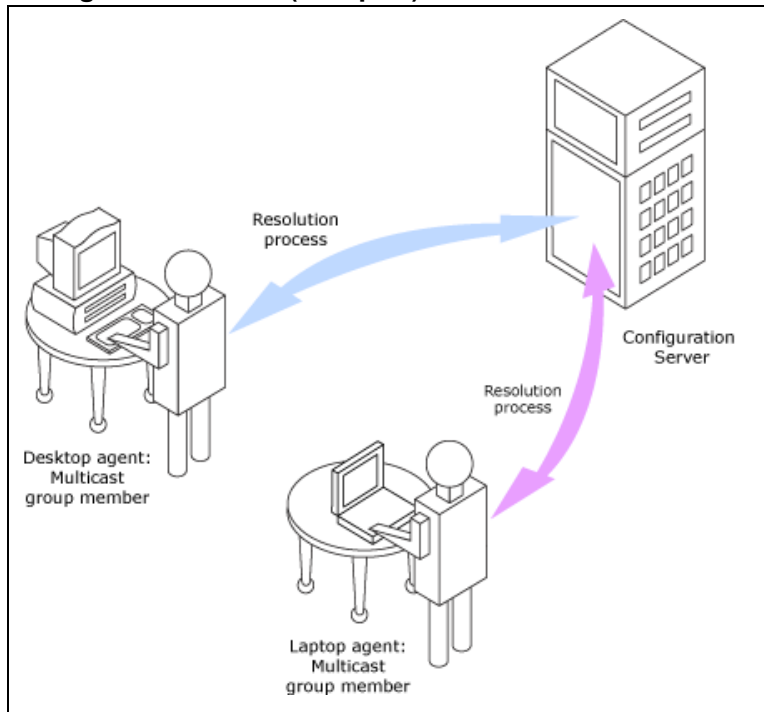
### Configuration Phase

(Sentient agents –Configuration Server) and (Sentient agents –Multicast Server)

The configuration phase (shown in figure [Configuration Phase \(first part\)](#) and figure [Configuration Phase \(second part\)](#)) occurs during a standard Sentient agent connect with the Configuration Server. An Sentient agent goes through a normal resolution to determine the resources it

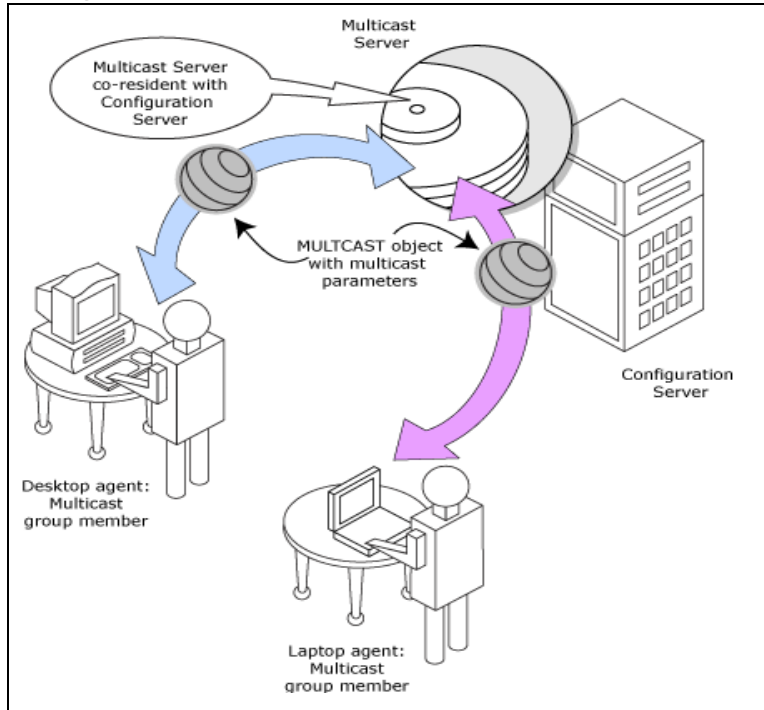
requires, and whether it is attached to a multicast group, as established by the system administrator.

### Configuration Phase (first part)



If a Sentient agent is eligible, it then connects to the Multicast Server and exchanges the MULTICAST object, which contains the multicast parameters. If the connection fails in this phase, the Sentient agent will perform normal, point-to-point retrieval (with the Configuration Server, using TCP/IP without multicast) to obtain all resources that are required in order to achieve the desired state.

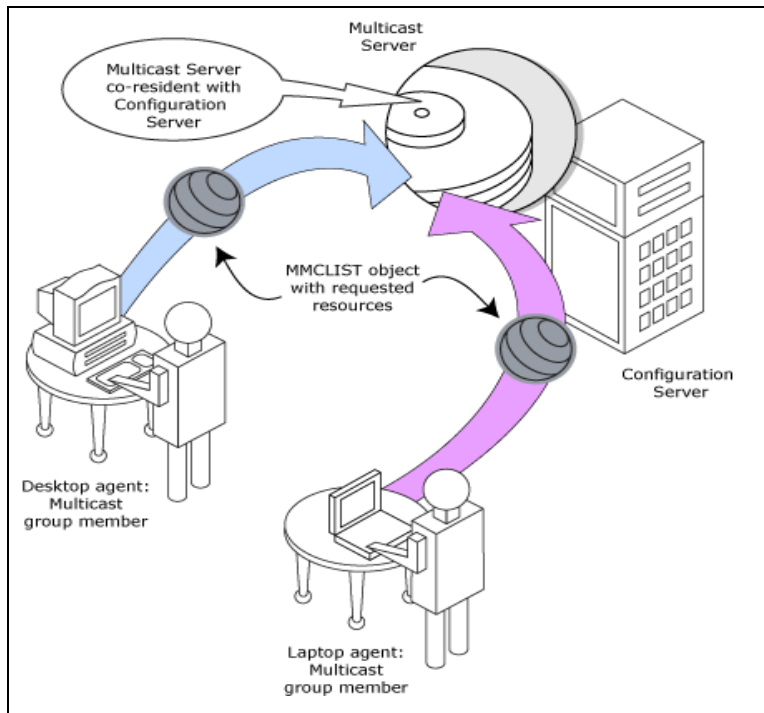
Thus far, all communications have been performed using Persistent Systems standard TCP/IP communications.

**Configuration Phase (second part)****Collection Phase**

(Sentient agents – Multicast Server)

In this phase (assuming the criteria to be eligible for multicast distribution, as described in the first phase, have been met), the Sentient agent will send, to the Multicast Server, the MMCLIST object (the list of requested resources that was compiled by the Configuration Server during the configuration phase). The Multicast Server stockpiles all Sentient agent requests and prepares to transmit the files at the scheduled time. The Sentient agent adopts a wait mode in anticipation of the multicast transmission (See figure [Collection Phase](#)).

### Collection Phase

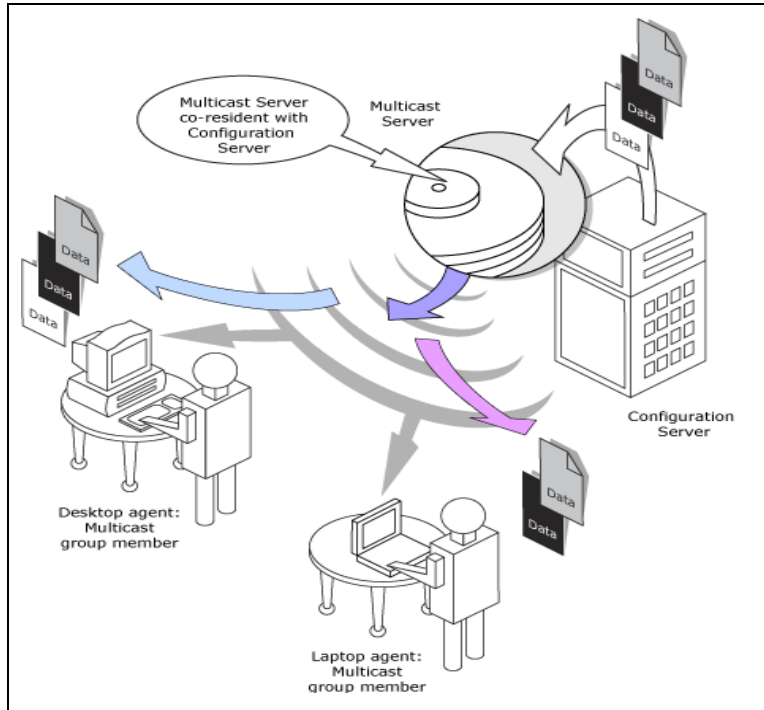


### Distribution Phase

(Sentient agents –Multicast Server –Configuration Server)

The third phase is the actual multicast transmission of the files (see the figure [Distribution Phase](#)). The Multicast Server retrieves required files from the Configuration Server, and then transmits a single data-stream that contains only the files and components that have been defined by the Sentient agents involved. As files arrive, they are checked (in the MMCLIST object) by the Sentient agent to determine if they are required. If required, they are read from the multicast stream and marked to indicate that they have been received.

### Distribution Phase



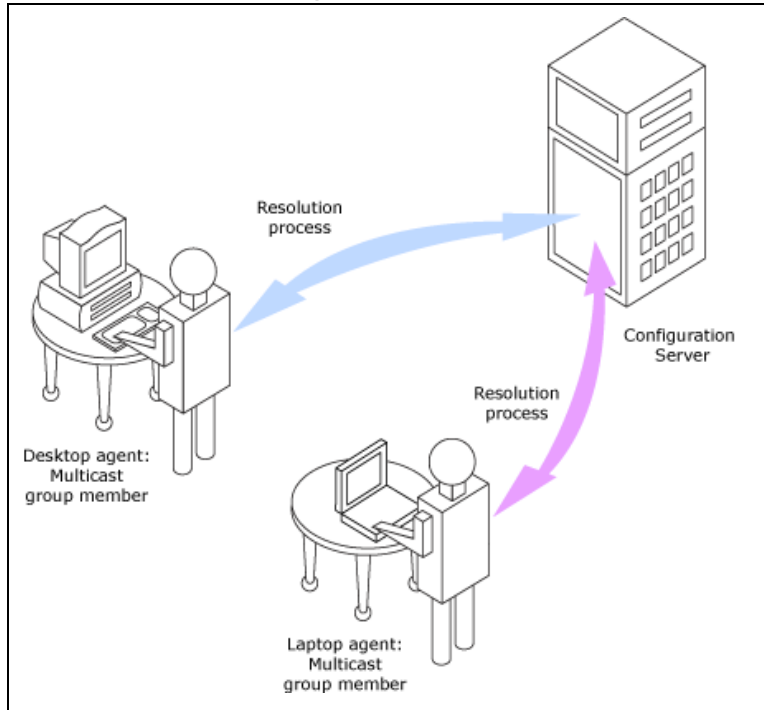
### Clean up-and-Reporting Phase

(Sentient agents – Configuration Server)

The final phase of the process runs when all required files have been received by the Sentient agent, or the distribution time window has expired. At this time, the Sentient agent checks for the presence of all required resources, and uses standard Sentient communications techniques to obtain, from the CSDB, any that are missing. Installation activities are then run for the resources; the completion status (including error information and multicast file-transfer statistics) is reported to the CSDB (see figure [Clean up-and-Reporting Phase](#)).



### Clean up-and-Reporting Phase



### Completion Status Information

Unlike most multicast implementations, which are fundamentally file transfer utilities, the Multicast Server provides full status information on the completion and installation of the distribution for each connecting Sentient agent.

For more information on how multicast session results can be used for auditing and reporting, see "Multicast Results for Reporting" in the *Sentient Enterprise Console Online Help*.

## Advanced Multicast Configuration Options

Three multicast-specific variables (MCORDER, MWINDOW, and BYPASCON) are in the MULTICAST Class. A fourth, MCELIGBL, is in the ZSERVICE Class. This section describes these variables, and how they can be customized using the CSDB Editor.

### MCELIGBL

In order to participate in a multicast session, services must be individually defined as multicast-eligible. This is done with the MCELIGBL variable, found in PRIMARY.SOFTWARE.ZSERVICE. All services that are instances of the ZSERVICE Class will, by default, be multicast-eligible because the Configuration Server installation sets MCELIGBL=Y.

If a service is not to be distributed via multicast, edit only that service by specifying MCELIGBL=N.

**Note:** Do not edit the `_BASE_INSTANCE_`.

## MCORDER

If a service is mandatory but not eligible for multicast, it will be downloaded to and installed on the Sentient agent directly from the CSDB. MCORDER enables an administrator to specify when to download and when to install mandatory services that are not multicast eligible.

Services can be configured as mandatory with the on/off switch, **ZSVCMO**, an instance attribute of all services.

The MCORDER attribute is located in the PRIMARY.POLICY.MULTICAST Class. It has three values, as described in the table "[MCORDER Values](#)".

### MCORDER Values

| Value | Result                                                                                                                                                                    |
|-------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| B     | Download the service and install it on the Sentient agent BEFORE the multicast session.                                                                                   |
| A     | Download the service and install it on the Sentient agent AFTER the multicast session.                                                                                    |
| S     | SPLIT the process so that it downloads the service to the Sentient agent before the multicast, and installs the service after the multicast session. This is the default. |

## MWINDOW

This variable controls the amount of time (in minutes) for an Sentient agent to run before terminating its multicast "receive" activity. The default is 0, which allows an Sentient agent to maintain the receive mode for as long as is necessary to receive all the requested data. This will not affect the duration of the multicast session; it will ensure only that the Sentient agent process does not run longer than is necessary to receive the data it requested.

**Note:** For more information on calculating a value for MWINDOW, see Appendix D, "[Implementation and Diagnostics](#)" on page 1454.

## BYPASCON

When multicast processing occurs, each Sentient agent connects to the Configuration Server and the Multicast Server. The Sentient agents then send a **control object** and the list of required resource files (a needs list) to the Multicast Server.

However, if all the Sentient agents' data payloads are identical, Sentient multicasting can be configured to save on network bandwidth usage by limiting which Sentient agents send which data to the Multicast Server. This is accomplished with the BYPASCON variable (of the PRIMARY.POLICY.MULTICAST Class). BYPASCON has three values.

### BYPASCON=

All Sentient agents will exchange a control object and a needs list object with the Multicast Server.

**BYPASCON=UserJoe**

The Sentient agent with this user ID will send both objects to the Multicast Server, and the other Sentient agents will send only the control object.

**BYPASCON=UserJoe 2** (user ID followed by a space and the numeral 2)

The Sentient agent with this user ID will send both objects to the Multicast Server; no other Sentient agents will be part of the object exchange.

**Note:** This option requires that the local time on the clock of the machine that houses the Multicast Server be in synchronization with all multicast-eligible Sentient agents.

## Burst Mode

**Note:** This feature is for multicast software distribution only. It is not supported for use with Sentient OS Manager.

The *Guaranteed delivery* option (see "[Guaranteed Delivery](#)" on page 1439) provides higher data-transmission rates and more reliable transmission of large files, but is designed for OS image downloads only—not software downloads.

The **Burst Mode** option improves the reliability and speed of software downloads.

## Using Burst Mode

The Multicast Server has the basic mode of operation of sending one packet of data, then waiting the specified number of milliseconds (the value of `DELAYBP`) before sending the next packet.

For more information on `DELAYBP`, see:

- "DELAYBP" in the table "MULTICAST Class Instance Attributes" in the *Sentient Enterprise Console Online Help*, and
- "[Calculate DELAYBP and MWINDOW](#)" on page 1455.

After every 20 packets (this number is configurable), each packet is re-sent one or more times (the value of `RESENDS` [see "RESENDS" in the table "MULTICAST Class Instance Attributes" in the *Sentient Enterprise Console Online Help*]). Although this increases reliability, it does not guarantee that large files will be received.

The **burst mode** feature works differently. Instead of sending 1 packet and delaying for `DELAYBP`, Multicast Server sends multiple packets with no delay and then waits `DELAYBP` before re-sending them. So, the packets are still resent one or more times (the value of `RESENDS`), but in *bursts*, which gives a throughput of  $n$  multiplied by the basic mode, which is equivalent to  $n=1$ .

**Note:** This version of Multicast Server includes a version of `nvdkit` that is configured with a lower threshold for the delay—down to at least 2 milliseconds—which provides additional throughput that is independent of and in addition to burst mode.

## Parameter Values Selection

Assuming the typical values of `DELAYBP = 10` (ms) and `-burst = 1`, the following changes will affect the throughput as described.

- Setting `DELAYBP = 5` will roughly double the throughput.
- Setting `-burst = 5` will increase throughput by a factor of five.

When used in combination, the improvements in throughput, although not linear, will increase. For example:

- Setting `DELAYBP = 5` and `-burst = 5`, the throughput will increase by a factor of approximately **9**.

**Note:** These figures were obtained on a lightly loaded 1 GBps network. Actual results will vary according to network speed, reliability, and load.

A reasonable maximum value for `-burst` is 10, and the lowest value for `DELAYBP` would be 2.

## Setting DELAYBP and Burst

To set `DELAYBP` and Burst, follow these steps:

1. To open the CSDB Editor click **Start > Programs > Sentient Enterprise Administrator > Sentient Enterprise Administrator CSDB Editor**
2. In the tree-view, double-click **PRIMARY, POLICY, and MULTICAST**.
3. Specify a value for the `DELAYBP` variable in the **MULTICAST** instance.
  - Save the changes and close the CSDB Editor.
4. Open Windows Explorer and navigate to `SystemDrive:\Program Files\PSL\RCA\MulticastServer\etc.`
5. Specify a value for the `-burst` parameter in the Multicast Server's `mcast.cfg` file.
  - Save the changes and close Windows Explorer.
6. Stop the Multicast Server.
7. Restart the Multicast Server.

## Testing the Multicast Server

Now that the Sentient Multicast Server has been installed and configured, it is important to test it before putting it into production to ensure that it will execute as expected. There are three ways to perform a multicast test: one from an Sentient agent perspective, and two from an administrator perspective.

The Sentient agent-based method is not efficient in production because it is specific only for that Sentient agent. However, it is a good test to check the installation and configuration of the Multicast Server. The two remaining methods are more suited to a production environment

because they offer the ability to include multiple Sentient agents (as part of a group, as discussed in the previous chapter).

In order to conduct either of these tests, do the following.

- Delete the service that is going to be used for the test from the Sentient agent machine.
- Connect the multicast instance to the Sentient agent via a group.
- Set up a multicast instance with an active collection window.

## Sentient Agent vs. Administrator Testing

This section details the testing of Multicast Server from an Sentient agent perspective (starting "Sentient Agent Testing" below) and from an administrator perspective (starting "Administrator Testing" below).

### Sentient Agent Testing

To test the operation of the Multicast Server and the associated Sentient agent software:

**Note:** Make sure that the service that is being used for the test has been deleted from the Application Manager agent machine on which the test is being conducted.

1. On the Sentient agent machine, open a command window.
2. Change the directory to the database location where the Sentient agent is stored.
3. Type:

```
Radskman.exe mname=serverid,dname=SOFTWARE
,ipaddr=serverid,port=3464,cat=Y
```

where *serverid* is the network name of the Configuration Server.

This method of multicasting files is not suitable in production because these steps would have to be executed on each Sentient agent machine. The methods that are covered in the next section are more appropriate for production because the Configuration Server Database can be accessed and then used to include Sentient agents in a group, which can then be associated with a multicast instance.

This means that at a central location, multiple Sentient agents can be configured for a multicast session. Also, the eligibility of a service (MCELIGBL), the parameters for delivering the service (MCORDER), and the transmission of the request files (BYPASCON) can be determined.

### Administrator Testing

This section details testing Multicast Server from an administrator perspective.

### Notify and TIMER

In addition to the Sentient agent test presented above, the Application Manager **Notify** and **Scheduler (TIMER)** features are convenient ways to transmit data to Application Manager agent

computers.

To determine if the installation (multicast session) was successful, look at `connect.log` and `radrecv.log` in the Sentient agent's `log` directory.

## Network Test Modules

Sentient provides Multicast test modules to help you tune the multicast and broadcast parameters for a network configuration.

**Note:** The Multicast test modules are not included in the Sentient media. Contact Persistent Systems Technical support to obtain these modules.

The `multicast_test_modules` directory contains:

- the broadcast send and receive files (`BRECV.CMD` and `BSEND.CMD`),
- the multicast send and receive files (`MRECV.CMD` and `MSEND.CMD`),
- the receive executable (`radcrecv.exe`),
- the send executable (`radcsend.exe`),
- the image-deployment send files (`gdmcsend.exe` and `gdmsend.cmd`),
- two application `.dll` files, and
- four test files (`TESTDATA000n`) of varying sizes.

**Note:** The parameters that are specified in earlier sections of this document might not be suited to all network configurations. Specify values in accordance with the requirements of the environment.

## RADCSEND and RADCRECV

Two programs (RADCSEND and RADCRECV) perform broadcast and multicast file transfers, based on command line parameters. On Windows, these can be run from the command line using the `.cmd` files that are provided.

1. Copy the contents of `multicast_test_modules` to a temporary directory on the destination computer.
2. Run `mrecv.cmd`.
3. Copy the contents of `multicast_test_modules` to the source computer.
4. Run `msend.cmd`. `TESTDATA0001` – `TESTDATA0003` will be sent to the destination computer via multicast.  
RADCRECV will create the log and data files in the current directory.

**Note:** If Sentient is not installed, logs are created in the respective, current directories of each program. Otherwise, look for the `radcrecv.log` in the `log` subdirectory. If either `IDMSYS` or `IDMLOG` (in `win.ini` or `nvd.ini`) is defined, it will make use of directories for

the data and log files.

## Syntax

The syntax for the RADCSEND and RADCRECV programs is shown below. Following the syntax, the table "Syntax Parameters Defined" defines the parameters.

### RADCRECV

```
RADCRECV M|B [address] port buffers timeout (min) [Object with file names]
```

### RADCSEND

```
RADCSEND M|B address port filename file number delay after first packet delay between packets time-to-live buffers resends last-fileflag[1] packet data size [n-to-drop[0]]
```

#### Syntax Parameters Defined

| Parameter                       | Definition                                                                                                                                                                                     |
|---------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| M B                             | M = Multicast, B = Broadcast                                                                                                                                                                   |
| <i>address</i>                  | The target network IP address (xxx.xxx.xxx.xxx).<br><br><b>Note:</b> This parameter must be omitted for RADCRECV in Broadcast mode, and always used for RADCSEND.                              |
| <i>port</i>                     | The IP port that is associated with the (broadcast or multicast) transmission.                                                                                                                 |
| <i>filename</i>                 | The name of the file to be sent.                                                                                                                                                               |
| <i>file number</i>              | The relative file number (if more than one file is to be sent, increment this by 1 for each successive file). RADCSEND expects to find these files in the directory from which it is executed. |
| <i>delay after first packet</i> | The delay (in milliseconds) after the first packet before sending the remaining packet. The delay gives the receiver time to open and compare the file.                                        |
| <i>delay between packets</i>    | The delay (in milliseconds) between data packets (packet size approximately 1 KB).                                                                                                             |
| <i>time-to-live</i>             | The IP TTL value (maximum number of routers to traverse).                                                                                                                                      |
| <i>buffers</i>                  | The number of packets to buffer when packet-retransmits (resends) are specified.                                                                                                               |

| Parameter                     | Definition                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|-------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>resends</i>                | The number of retransmits for each grouping of <i>buffers</i> packets.                                                                                                                                                                                                                                                                                                                                                                                             |
| <i>last-file-flag</i>         | Indicates the last file to be transmitted. This will cause the receiver to terminate execution at the end of this file. The default is 1 (true). Set to zero for all but the last file, if multiple files are sent.                                                                                                                                                                                                                                                |
| <i>packet size</i>            | The maximum number of data bytes transmitted per packet.                                                                                                                                                                                                                                                                                                                                                                                                           |
| <i>n-to-drop</i>              | Set this to a non-zero value to simulate packets being dropped by the transmission network. RADCSEND will randomly drop packets so that (100 / n-to-drop) percent of the packets will not be sent, and therefore, not received by RADCRECV.<br><br>This value is optional and the default is 0.                                                                                                                                                                    |
| <i>Object with file names</i> | Multicast only. This value is optional.<br><br>RADCRECV will store files in IDMDATA as named in this object. If not specified, the received file is stored in the same directory as RADCRECV. The object must have the same format as MMCLIST.<br><br>RADCRECV also looks in IDMLIB for the object, MULTICAST, which is also optional. To set the parameters log level (default = <b>40</b> ) and resends (default = <b>0</b> ), ZTRACEL and RESENDS are set here. |
| <i>timeout (min)</i>          | Maximum time that RADCRECV will accept data before stopping (failsafe mechanism for live run).                                                                                                                                                                                                                                                                                                                                                                     |

**Note:** If MULTICAST.ZTRACEL is set to 55 or greater, RADCRECV will log header information for every packet that gets to the GetControlInfo routine. (RADCRECV bypasses packets that are “malformed” or not intended for the multicast/broadcast session it is looking for.)

## BROADCAST Tests

The contents of the BSEND.CMD file are:

```
radcsend B 231.1.222.6 9511 TESTDATA000n 1 50 50 2 20 1 1 1020
```

where:

### RADCSEND Parameters for BROADCAST

|              |                                                 |
|--------------|-------------------------------------------------|
| B            | Indicates Broadcast                             |
| 231.1.222.6  | <i>address</i> (specified on the command line)  |
| 9511         | port                                            |
| TESTDATA000n | <i>filename</i> (specified on the command line) |



|               |                                                            |
|---------------|------------------------------------------------------------|
| 1 (2, 3, ...) | <i>file number</i>                                         |
| 50            | <i>delay after first packet</i>                            |
| 50            | <i>delay between packets</i>                               |
| 2             | <i>time-to-live</i>                                        |
| 20            | <i>buffers</i>                                             |
| 1             | <i>resends</i>                                             |
| 1             | <i>last-file-flag</i>                                      |
| 1020          | <i>packet data size</i>                                    |
|               | <i>n-to-drop (optional, not used in the above example)</i> |

To run, type:

BSEND dest\_IP\_address filename

**Note:** The receive program must be started first, in order to receive the file.

The contents of the BRECV.CMD file are:

```
radcrecv B 9511 10 45
```

where:

#### RADCRECV Parameters for BROADCAST

|      |                          |
|------|--------------------------|
| B    | Indicates Broadcast      |
| 9511 | <i>port</i>              |
| 10   | <i>buffers</i>           |
| 45   | <i>timeout (minutes)</i> |

To run, type:

BRECV

## MULTICAST Tests

The contents of the MSEND.CMD file are:

```
radcsend M 231.1.222.6 9511 TESTDATA0001 1 50 50 2 20 1 0 1020radcsend
M 231.1.222.6 9511 TESTDATA0002 2 50 50 2 20 1 0 1020radcsend M
231.1.222.6 9511 TESTDATA0003 3 50 50 2 20 1 1 1020
```

where:

#### RADCSEND Parameters for MULTICAST

|   |                     |
|---|---------------------|
| M | Indicates Multicast |
|---|---------------------|

|               |                                                            |
|---------------|------------------------------------------------------------|
| 231.1.222.6   | <i>address</i> (specified on the command line)             |
| 9511          | <i>port</i>                                                |
| TESTDATA000n  | <i>filename</i> (specified on the command line)            |
| 1 (2, 3, ...) | <i>file number</i>                                         |
| 50            | <i>delay after first packet</i>                            |
| 50            | <i>delay between packets</i>                               |
| 2             | <i>time-to-live</i>                                        |
| 20            | <i>buffers</i>                                             |
| 1             | <i>resends</i>                                             |
| 1             | <i>last-file-flag</i>                                      |
| 1020          | <i>packet data size</i>                                    |
|               | <i>n-to-drop</i> (optional, not used in the above example) |

To run, type:

MSEND

**Note:** The receive program must be started first, in order to receive the file.

The contents of the MRECV.CMD file are:

```
radcrecv M 231.1.222.6 9511 20 45
```

where:

#### **RADCRECV Parameters for MULTICAST**

|             |                                                                  |
|-------------|------------------------------------------------------------------|
| B           | Indicates Multicast                                              |
| 231.1.222.6 | <i>address</i>                                                   |
| 9511        | <i>port</i>                                                      |
| 20          | <i>buffers</i>                                                   |
| 45          | <i>timeout (minutes)</i>                                         |
|             | Object with file names (optional, not used in the above example) |

To run, type:

MRECV

## Implementation and Diagnostics

This section offers information on:

- Implementing the multicast process
  - ["Preliminary Parameter Calculation"](#) below,
  - ["Pre-Multicast Session Checklist"](#) on page 1457
  - ["Multicast Program-Call Schematic"](#) on page 1457

## Preliminary Parameter Calculation

Before attempting to run a multicast session, it is necessary to:

1. Calculate the DELAYBP and MWINDOW values.
2. Test the network (using the multicast test modules, RADCSEND and RADCRECV) to ensure that it is properly configured for multicast.
3. Examine the test logs.

## Calculate DELAYBP and MWINDOW

### DELAYBP

$$\text{DELAYBP} = (8 * P / N)$$

- P is the packet size (typically, **1020** bytes) and
- N is the network bandwidth (in this example, **16** kilobits/second). Therefore, **8** multiplied by **1020** (P) divided by **16384** ( $16 * 1024$ ) equals **0.498** seconds, which rounds to **0.5** second.

So, **DELAYBP = .5 second**.

### MWINDOW

$$\text{MWINDOW} = D * (T / P) * (R + 1)$$

- D is the DELAYBP (as computed above),
- IT is the total bytes (to be transmitted),
- P is packet size (from DELAYBP example), and
- R is the number of resends. Assuming values of: R = **1** and T = **10MB** (10,000,000 bytes), the value of MWINDOW computes as follows: **0.5** (D) multiplied by **10,000,000** divided by **1020** (T/P) multiplied by **2** (R + 1) equals **9803.9215** seconds, which rounds to **9804** seconds.

So, **MWINDOW = 2 hours, 44 minutes**.

## Test the Network

Run the network test modules as discussed in Appendix C, ["Network Test Modules"](#) on [page 1450](#) on page 46 but substitute the value of DELAYBP with that which pertains to your network.

**Note:** Considering the MWINDOW value previously calculated, it is recommended that, if testing on a 16-Kbps network, a smaller test file (such as 10 KB) be used. This should take approximately 10 seconds.

**Note:** MWINDOW is only a fail-safe to ensure termination of the receive program (RADCRECV). Use it in production—not for testing, where you are observing the test. Typically, it is used as an estimate of the end of the transmission. In practice, add time for a margin of safety.

## Examine the Test Logs

**Note:** The following examples contain explanatory comments (*italicized test within parentheses*) that will not be in the log.

### RADCRECV

The `radcrecv.log` is in the same directory as the program. If the Sentient agent is on the same machine, the log is in the `log` subdirectory of the Sentient agent installation directory, `IDMSYS`. At the end of the log is the summary information (shown below with irrelevant text removed):

```
08:50:57 Number of times Each Packet is Sent [1] (RESENDS + 1, no
bearing on receiver)08:50:57 Multicast Packet Inactivity Timeout [5]
(minutes)08:50:57 Multicast IP address [231.1.222.6] mode [M]08:50:57
Port [9511]08:50:57 Number of buffers: [20]08:50:57 MWINDOW value in
(sec): [2700]09:30:20 Packets received: 39216 dropped(est): 0 (The
number ofunique packets received, not counting resends. The number
dropped isan estimate, based on the gaps in the packet number
sequence.)09:30:20 Done
```

### RADCSEND

The `radcsend.log` (in earlier versions, this was `edmcSEND.log`), is always in the program's directory. The log has this information before the start:

*(Test with: 20ms delay, 2 resends, 40 MB file, MWINDOW of 2353 seconds = 39.2 minutes)*

```
EDMCSEND started on [Wed Nov 20 08:44:58 2002]Mode selected is:
[M]Destination address is: [231.1.222.6]Destination port is:
[9511]Processing file: [TESTDATA0004]Relative file number: [1]Delay
after the first packet: [20]Delay between packets: [20]Number of
packets to re-send: [20]Number of resends: [2]Time to live: [3]Last
file flag: [1]Packet Data Size: [1020]File Size: [40000000]
```

*(And at the end:)*

```
Total Packets: 117648Exiting on [Wed Nov 20 09:24:11 2002]: [Done]
```

*(From this, the following can be computed: Effective delay between packets: 2353 seconds divided by 117648 packets = 20 ms/packet(?). Thruput: 40 MB divided by 2353 sec = 0.017 MB/sec [remember, there were 2 resends])*

## Pre-Multicast Session Checklist

Before starting a multicast session, do the following:

1. Be sure that each SERVICE to be multicast is defined as MANDATORY.
2. Specify MCELIGBL = Y for SERVICES to be multicast.
3. Connect the SERVICE instances and the MULTICAST instance to the USER or WORKGROUP instance.
4. Set the MULTICAST instance parameters, as outlined below: The most common problems are determining the start time and duration of the Collection window (CGMTTIME and CWINDOW). The Collection window should be long enough to allow Sentient agents to:
  - Connect to the Configuration Server to determine the needed resources, and
  - Connect to the Multicast Server to exchange the request information.

**Note:** Be advised that the internal clocks on the machines (Sentient agents and servers) should be synchronized to within a few minutes of each other, with respect to GMT. This means that CGMTTIME should be a few minutes earlier than the time that the Sentient agents are expected to connect, and the CWINDOW time should be a few minutes later than the time that the last Sentient agent is expected to complete the collection/request phase of multicast.

5. Specify identical values for the parameters DELAYBP and DELAYFP.

**Note:** The value of DELAYBP is determined by the calculations shown in "Calculate DELAYBP and MWINDOW" on page 1455 and modified by the results of the tests shown in "Examine the Test Logs" on the previous page

The value of MWINDOW (calculated in "Calculate DELAYBP and MWINDOW" on page 1455) should be padded 20–50% to allow for unforeseen influences, such as network congestion.

**Note:** Remember to base these calculations on the total number of bytes of all the files that are expected to be transmitted.

## Multicast Program-Call Schematic

The figure "Call Sequence of the Multicast Session Programs" on the next page illustrates the sequence in which the various programs are called during a multicast session.

**Note:** The execution flows from *left to right* and *top to bottom*.

**LEGEND:**  
**A** = Asynchronous execution  
**S** = Synchronous execution

The flowchart illustrates the RADSKMAN process, which is divided into three main phases:

- Collection Phase:** Initiated by a "Notify or Scheduler Start Collection Phase" event, leading to the "RADSKMAN (Collection Phase)" process. This process then branches into two paths:
  - Left Path (Collection Phase):** A sequence of synchronous steps (S) involving "RADPINIT", "RADCONCT (Collection Phase: verify, difference, and transfer)", and "RADCLECT".
  - Right Path (Distribution/Installation Phases):** An asynchronous step (A) leading to the "RADREQST" process.
- Distribution/Installation Phases:** The "RADREQST" process branches into two main paths:
  - Distribution Phase:** A synchronous step (S) leading to "Communicate with the Multicast Server", which then leads to "RADCRECV".
  - Installation Phase:** A synchronous step (S) leading to "RADSKMAN (Installation Phase)".
- Central Components and Data Flow:**
  - The "Multicast Server" receives "RESOURCE files" from the "Configuration Server File System" and sends "RESOURCE files sent via multicast" to "RADCRECV".
  - The "Multicast Server" also interacts with the "MULTICAST MMCLIST" and the "Configuration Server".
  - The "Configuration Server" receives "ZMASTER MULTICAST" from the "RADCLECT" process and sends "ZMASTER" to the "RADCONCT (Installation Phase)" process.
  - The "RADCONCT (Installation Phase)" process sends "ZSTATS (via RADCONCT)" back to the "Configuration Server".
- Annotations:**
  - A note indicates that the sequence from "RADSKMAN (Collection Phase)" through "RADPINIT", "RADCONCT", and "RADCLECT" is called for each mandatory service.
  - Another note indicates that the sequence from "RADSKMAN (Installation Phase)" through "RADPINIT" and "RADCONCT" is called for each mandatory service.

# Appendix A

---

## Smart Card Authentication

Sentient Enterprise supports two-way authentication using smart cards. SSL must be enabled for smart card authentication.

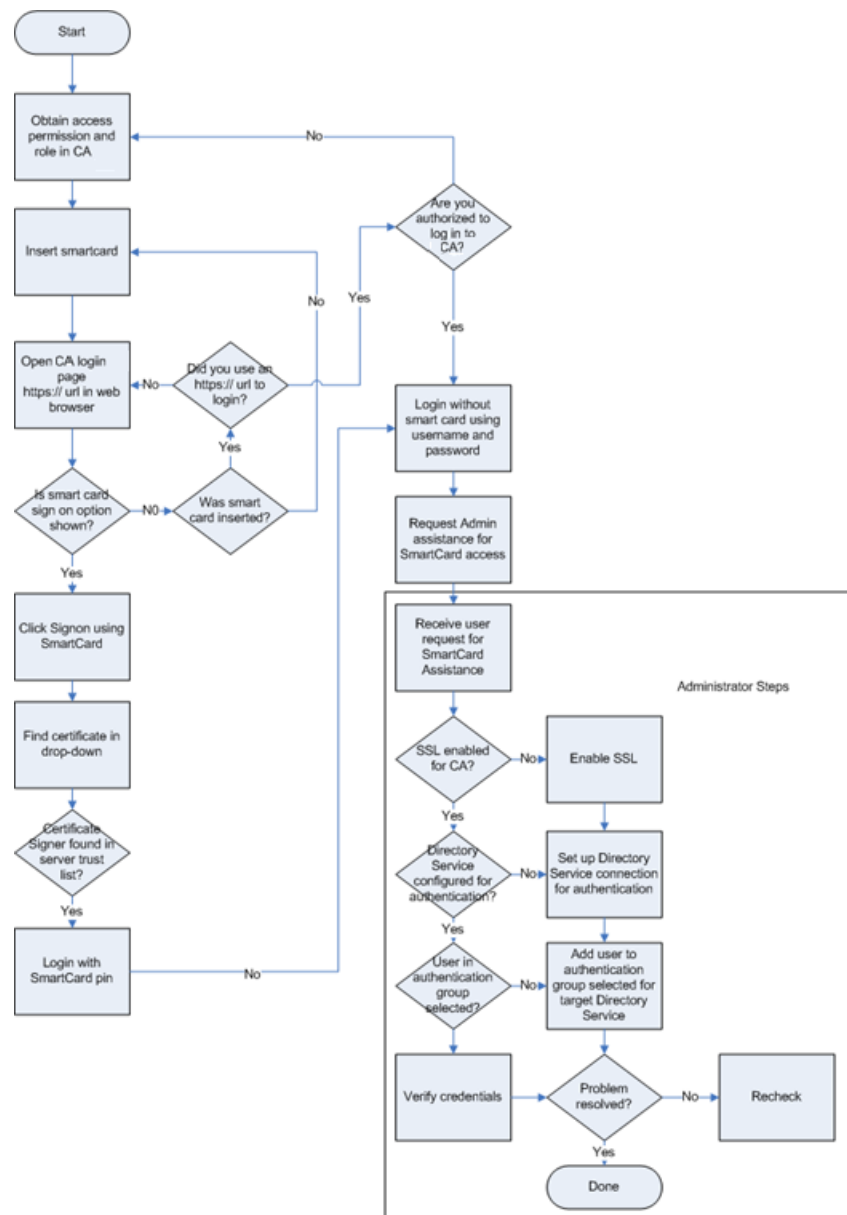
As part of the smart card login process, the user must select a certificate that matches a trusted certificate in the Core Server truststore. The process of validating this certificate against the user in the directory consists of the following checks:

- **subjectdn**  
The domain name (subjectdn) value of the certificate is obtained. A check is performed to determine if the subjectdn matches the equivalent userdn in one of the mounted directories where authentication is enabled. If so, the user is eligible to login. If not, the altsubjectname check is performed.
- **altsubjectname**  
The alternate subject name (altsubjectname) value of the certificate is obtained. A check is performed to determine if the altsubjectname matches the AD userprincipal name in one of the mounted directories where authentication is enabled. If so, the user is eligible to login. If not, the email address check is performed.
- **email address**  
It is determined if the certificate has an email address value in the subjectdn. If available, a check is performed to determine if it matches the mail attribute in one of the mounted directories where authentication is enabled. If so, the user is eligible to login. If not, the altSecurityIdentities match is performed.
- **altSecurityIdentities**  
It is determined if the certificate has a subject name and an issuer name. If available, a check is performed to determine if it matches the attribute in altSecurityIdentities. If so, the user is eligible to login. If not, the usercertificate match is performed.
- **usercertificate match**  
A check is performed to determine if the usercertificate matches the usercertificate attribute in one of the mounted directories where authentication is enabled. If so, the user is eligible to login. If not, login fails.

## Smart Card Login Process

The following diagram depicts the steps involved in the Smart Card login process. It provides alternative actions and questions to consider if a step in the normal process fails.

### Smart Card Login Process





# Appendix B

---

## Microsoft Products Supported

Sentient Patch Manager supports the following Microsoft Products:

- .NET Framework
- .NET Framework 1.1
- .NET Framework 2.0
- .NET Framework 2.0
- .NET Framework 3.5
- .NET Framework 4.0
- Antigen for Exchange/SMTP (MU)
- ASP .NET MVC
- BackOffice Resource Kit SE
- BackOffice Resource Kit SE
- BackOffice Server 4.0
- BizTalk Server 2000
- BizTalk Server 2002
- BizTalk Server 2002 (MU)
- BizTalk Server 2004
- Commerce Server 2000
- Commerce Server 2002
- Communicator 2007
- Compatibility Pack for the 2007 Office system
- Compute Cluster Pack
- Data Protection Manager 2006 (MU)
- Exchange 2000 Enterprise Server
- Exchange 2000 Server
- Exchange Server 2003
- Exchange Server 5.0
- Exchange Server 5.5
- Expression Media V1

- Expression Media V2
- FrontPage 2000
- FrontPage 2000 Server Extensions
- FrontPage 2002
- FrontPage 2003
- FrontPage 97 Server Extensions
- FrontPage Server Extensions 2002
- Host Integration Server 2000 (HIS)
- Host Integration Server 2004 (HIS)
- Host Integration Server 2006 (HIS)
- Host Integration Server 2009(HIS)
- Host Integration Server 2010(HIS)
- Index Server 2.0
- Indexing Services for Windows 2000
- InfoPath 2003
- Internet Explorer 10
- Internet Explorer 11
- Internet Explorer 4.0
- Internet Explorer 4.01
- Internet Explorer 5
- Internet Explorer 5.01
- Internet Explorer 5.5
- Internet Explorer 6
- Internet Explorer 6.0 for Windows Server 2003
- Internet Explorer 7
- Internet Explorer 8
- Internet Explorer 9
- Internet Information Server 3.0
- Internet Information Server 4.0
- Internet Information Server 5.0
- Internet Information Server 5.1
- Internet Information Server 6.0
- Internet Security and Acceleration Server 2004

- Internet Security and Acceleration Server 2006
- ISA Server 2000
- Lync 2010
- Lync Attendant 2010
- Lync Attendee 2010
- MDAC 2.1
- MDAC 2.5
- MDAC 2.6
- MDAC 2.7
- MDAC 2.8
- Microsoft Data Engine (MSDE)
- Microsoft Exchange Server 2000
- Microsoft Exchange Server 2003
- Microsoft Exchange Server 2007
- Microsoft Exchange Server 2010
- Microsoft Exchange Server 2013
- Microsoft Exchange Server 2016
- Microsoft Expression Design 1
- Microsoft Expression Design 2
- Microsoft Expression Design 3
- Microsoft Expression Design 4
- Microsoft Forefront Client Security
- Microsoft Forefront TMG Client 2010 (MU)
- Microsoft Office 2002
- Microsoft Office 2003
- Microsoft Products
- Microsoft Report Viewer Redistributable 2005
- Microsoft Report Viewer Redistributable 2008
- Microsoft Research AutoCollage 2008
- Microsoft SQL Server 2000 Desktop Engine (MSDE 2000)
- Microsoft SQL Server 2000 Desktop Engine (WMSDE)
- Microsoft System Center Virtual Machine Manager 2008
- MS Office 2007

- MS Office 2010
- MS Office 2013
- MS Office 2016
- MS SQL
- MS SQL 2000
- MS SQL 2005
- MS SQL 2008
- MS SQL 2008 R2
- MS SQL 2012
- MSN Messenger 6.0
- MSN Messenger 6.1
- MSN Messenger 6.2
- Office 2000
- Office 2003
- Office 2007 Com
- Photo Draw 2000 Version 2
- Silver Light
- SMS
- SQL Server 2000
- SQL Server 7.0
- SQL Server Desktop Engine (MSDE) 1.0
- SQL Server Desktop Engine (MSDE) 2000
- System Center Configuration Management 2007 (SCCM 2007)
- Systems Management Server 1.2
- Systems Management Server 2.0
- VBA 6.0
- Virtual PC
- Virtual Server
- Visual Studio 2005
- Visual Studio 2008
- Visual Studio 2010
- Windows 10
- Windows 2000 (MU)

- Windows 2000 Advanced Server
- Windows 2000 Professional
- Windows 2000 Server
- Windows 2008
- Windows 7
- Windows 7 Embedded
- Windows 7 Server
- Windows 8
- Windows 8 Embedded
- Windows 8 Server
- Windows 8.1
- Windows 8.1 Server
- Windows Data Center 2003
- Windows Internet Explorer Dynamic
- Windows Media Player 10.0
- Windows Media Player 11.0
- Windows Media Player 6.4
- Windows Media Player 6.4 for Windows 2000
- Windows Media Player 6.4 for Windows NT 4.0
- Windows Media Player 6.4 for Windows Server 2003
- Windows Media Player 6.4 for Windows XP
- Windows Media Player 7.0
- Windows Media Player 7.1
- Windows Media Player 9 Series
- Windows Media Player 9 Series for Windows Server 2003
- Windows Media Player 9 Series for Windows XP
- Windows Media Player 9.0
- Windows Media Player for Windows XP
- Windows Messenger 4.7
- Windows Messenger 5.0
- Windows Messenger 5.1
- Windows NT Server 4.0
- Windows NT Server 4.0, Enterprise Edition

## Reference Guide

### Appendix B: Microsoft Products Supported

---

- Windows NT Server 4.0, Terminal Server Edition
- Windows NT Workstation 4.0
- Windows Script 5.1
- Windows Script 5.5
- Windows Server 2003 for Small Business Server
- Windows Server 2003, Datacenter Edition
- Windows Server 2003, Enterprise Edition
- Windows Server 2003, Standard Edition
- Windows Server 2003, Web Edition
- Windows Vista
- Windows XP
- Windows XP Home Edition
- Windows XP x64
- Windows XP x64 2003
- Works Converter

## We appreciate your feedback!

If an email client is configured on this system, by default an email window opens when you click [here](#).

If no email client is available, copy the information below to a new message in a web mail client, and then send this message to [radiadocfeedback@accelerite.com](mailto:radiadocfeedback@accelerite.com).

**Product name and version:** Sentient Enterprise, 11.00

**Document title:** Reference Guide

**Feedback:**

