Persistent

SASVA IDE Plugin 2.1.2

For Visual Studio Code

User Guide

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 in this document should be considered as a promise of an additional warranty of any kind, implied, statutory, or in any communication between them, including without limitation, the implied warranties of merchantability, non-infringement, title, and fitness for a particular purpose. Persistent Systems shall not be liable for technical or editorial errors or omissions contained herein.
The information herein is subject to change anytime without notice.

Restricted rights legend

It is confidential computer software. A valid license from Persistent Systems or its licensors is required for possessing, using, 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 notices

© Copyright 2025 Persistent Systems, its affiliates, and licensors

Trademark notices

Persistent Systems or Persistent are trademarks or trade names or service marks or logos of Persistent. All other brands or products are trademarks, trade names, service marks, logos or registered trademarks of their respective holders or owners thereof.

Revision history

Version	Date	Description of change
1.0	29 July 2025	Initial release

For the latest version of this document, refer to the SASVA Help Articles page on:
https://support.accelerite.com/hc/en-us/sections/37500881870477

Contents

© 2025 Persistent Systems Ltd. All rights reserved.

© 2025 Persistent Systems Ltd. All rights reserved.

Overview

SASVA Plugin is used to enhance security posture, performance, and maintainability by utilizing AI-based recommendations. The goal of SASVA Plugin is to improve the developer's productivity, streamline workflows, and enhance the overall development experience. It can reduce the time and effort required for various development tasks allowing developers to extend and customize their development environments to better meet their needs.

Use Cases

SASVA Plugin enables developers to work more efficiently and effectively by enhancing functionality and improving code quality. The use cases reflect how SASVA Plugin can significantly boost developers' productivity, streamline workflows, and enhance project management capabilities.

Here is the breakdown of the use cases for SASVA Plugin covering specified functionalities.

1. Conduct a comprehensive project scan for quality, security, and compliance.
2. Generates visual representation of project architecture and dependencies.
3. Automate repetitive tasks in the project development process.
4. Identifies and fixes coding errors and issues automatically or with minimal intervention.
5. Provides AI-powered assistance for coding queries and problem-solving.
6. Speeds up the process of identifying and fixing bugs.

© 2025 Persistent Systems Ltd. All rights reserved.

Getting Started

This section will guide you through the essential setup steps including verifying system requirements, downloading the Plugin, meeting prerequisites, and completing the installation.

System requirements

Hardware
RAM – 32 GB
V CPU – 8

Software
Python (Version 3.11.0 or above)
Pip
VS Code (Version 1.97 or later)

Compatibility matrix

This matrix outlines the supported and tested third-party tools and platforms for the current release of the SASVA Plugin. Only the versions and configurations listed below are officially supported.

Category	Supported Tools/Platforms
Project Management Tools	\ Jira (Cloud & Server) \ Azure DevOps Boards
Code Repositories	\ Git \ Gitlab \ Bitbucket \ Azure DevOps (only supports public repositories)
Operating System	\ Windows 10/11

© 2025 Persistent Systems Ltd. All rights reserved.

Downloading SASVA IDE Plugin

1. Visit https://planner.accelerite.com.
2. Create your SASVA account and log in using your credentials.
3. After activating your account, visit https://planner.accelerite.com/download.

4. Select the desired **IDE type** from the drop-down menu.
5. Click **Download** to obtain the plugin suitable for your operating system.

   Once the download is complete, follow the instructions provided in the SASVA Plugin Installation section.

Prerequisites

1. Ensure that VS Code (Version >= 1.97) is installed on your system.
   If installation is needed, follow the given instructions:
   - Download the latest version from the official VS Code website.
   - Once installed, open a terminal or command prompt and run the following command to check the version:

     ```
     code --version
     ```
   - Confirm that the first figure in the output is 1.97 or higher.

2. Ensure that Python (Version >= 3.11) is installed on your system.
   If installation is needed, follow the given instructions:
   - Download Python from the official Python website.
   - Once installed, run the following command to check the version:

     ```
     python --version
     or (on some systems)
     ```

© 2025 Persistent Systems Ltd. All rights reserved.

```
python3 --version
```

\ Confirm the output shows version 3.11 or above.

3. Ensure that Git is installed on your system.
   If installation is needed, follow the given instructions:
   \ Download Git from the official Git website.
   \ Confirm the installation by running the following command:

   ```
   git --version
   ```

   \ The command should output the Git version installed on your system.

SASVA Plugin Installation

Follow the instructions to install the SASVA Plugin:

1. Open the Command Prompt.
2. Set the Environment Variable and use the `setx command` to define the SASVA_SERVER variable.
   Note: Before executing this command, ensure you have received the correct server URL from the support team.

   ```
   setx SASVA_SERVER <url from support>
   ```

3. After setting the variable, you can verify it by echoing its value:

   ```
   echo %SASVA_SERVER%
   ```

4. Open the VS Code Editor.
5. Click the **Extensions** icon in the Activity Bar on the left side of VS Code. This will open the Extensions view, displaying installed and recommended Plugins.
6. Click the 3-dot (ellipsis) button.
7. Click on **Install** from VSIX.
8. Browse and select the VSIX package.
9. Click **Install** to start the installation. This will install the SASVA Plugin on your IDE. You can see the installation status at the bottom status bar of the editor.
10. Run the command before using the Plugin:

    ```
    pip install gitpython --trusted-host pypi.org --trusted-host
    files.pythonhosted.org
    ```

11. Once it's installed the SASVA Plugin logo  appears on the activity bar. Proceed to the next section for configuration.

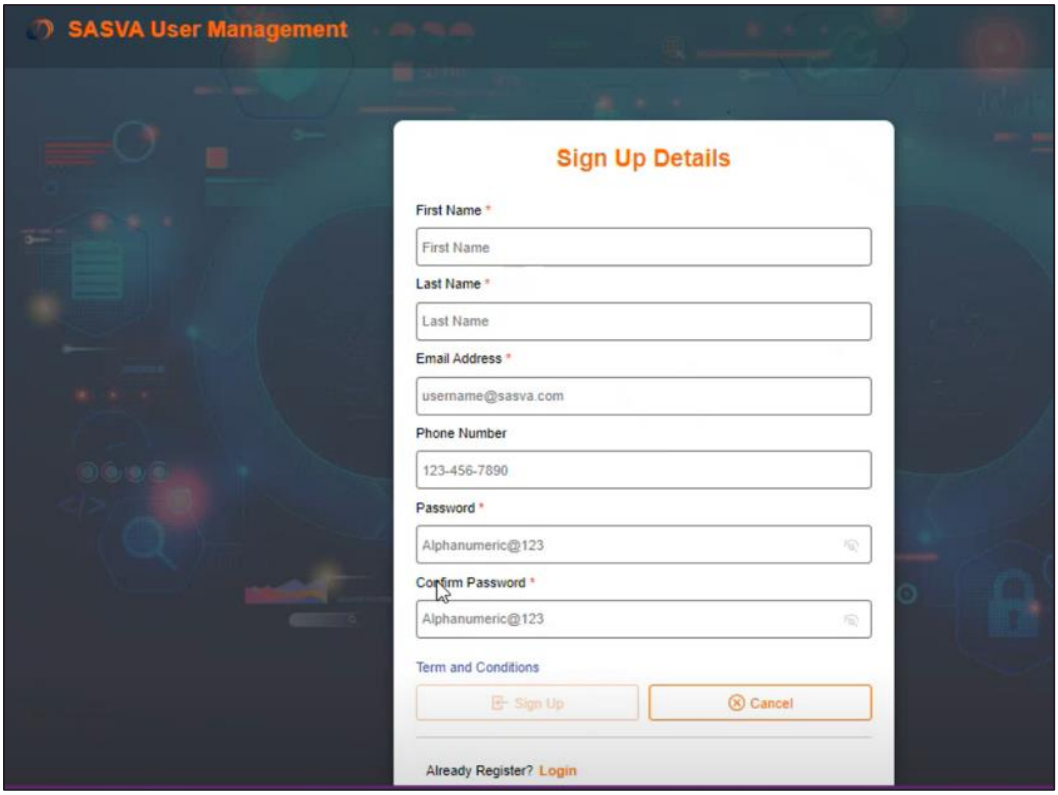
Signing In to SASVA Plugin

To access the SASVA Plugin, you can choose from the following sign-in options based on your account type and preferences.

© 2025 Persistent Systems Ltd. All rights reserved.

New Account Creation

1. Click the **Sign Up** option within the Login page.
   Sign Up Details page will appear.



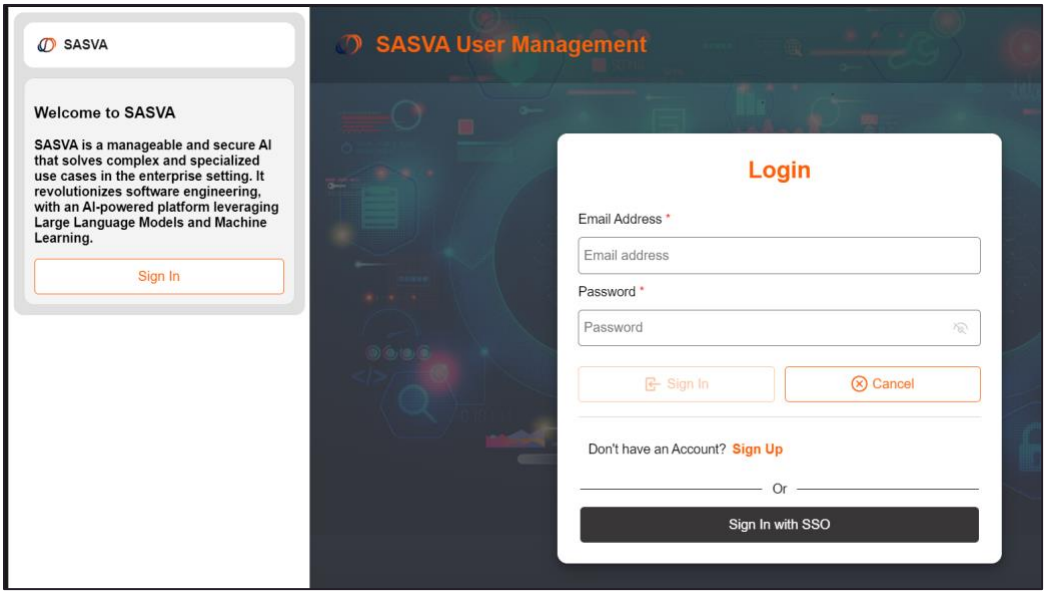
2. Enter Registration Information. The mandatory fields are marked with an asterisk (*) sign.
   a. **First Name**- Enter your first name.
   b. **Last Name**- Enter your last name.
   c. **Email**- Provide a valid email address.
   d. **Phone Number**- Enter a valid contact number for any communication.
   e. **Password**- Create a strong password for your account.
   f. **Confirm Password**- Re-enter the password to confirm.
3. Click the **Sign Up** button.
4. Send an email to sasva_support_in@persistent.com to activate the account.
5. Once your email is verified, Click the **Login**.
6. Sign in using your new account credentials.

Existing Account Holders

To sign in, follow the given instructions:

1. Click the **Sign in** button displayed on the left side of the homepage.
   The SASVA User Management Login page will appear.

© 2025 Persistent Systems Ltd. All rights reserved.

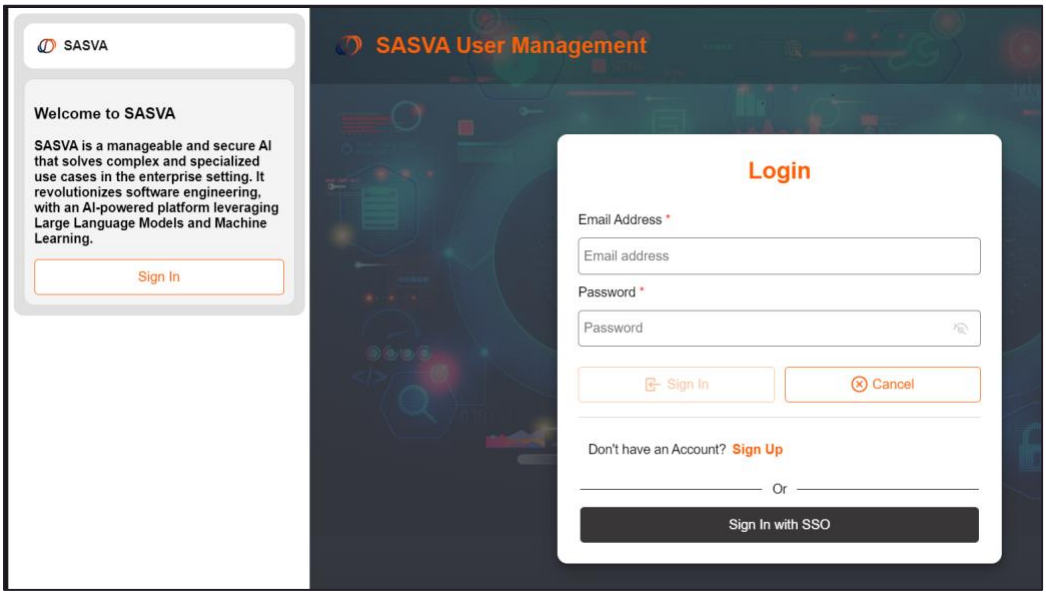


2. Input your registered email address and password.
3. Click the **Sign in** button to complete the Login process.

Sign in with SSO

To sign in with SSO (Single Sign-On), follow the given instructions:

1. Click the **Sign in** button displayed on the left side of the homepage.
   The SASVA User Management **Login page** will appear.



2. Click the **Sign In with SSO** button.
3. Authenticate using your credentials.
4. Once authentication is complete, you will see a confirmation message.
5. Switch back to SASVA IDE

© 2025 Persistent Systems Ltd. All rights reserved.

Features

Reload

When you click on reload, it refreshes the entire development environment and updates the User Interface.

Source Connect

Source Connect is a feature that allows you to connect to both Remote Repositories and Local Folders (enabling you to work with projects stored on your local machine without requiring a repository connection). It also supports multiple repository connections simultaneously.
After successfully connecting, you can initiate a project scan to get a comprehensive project overview.
The plugin scans your project in three progressive stages: **Basic**, **Intermediate**, and **Advanced**.
As each stage completes, related features are automatically enabled based on the scan results. You can begin using these features immediately, even while the scan continues in the background. This staged approach ensures you don't have to wait for the full scan to finish before accessing key functionalities.
Refer to the table below for the features enabled at each stage:

Stages	Features						
	SASVA Ask Lite Search	SASVA Ask Search	SASVA Ask Deep Search	Project Diagram	Assisted Defect Fix	Automated Defect Fix	Migration
Basic	✓	✗	✗	✗	✗	✓	✓
Intermediate	✓	✓	✗	✗	✗	✓	✓
Advance	✓	✓	✓	✓	✓	✓	✓

Additionally, the following features are independent of the scanning stages and are always available:

- SASVA Ask Generic
- Builder
- SASVA QE

You can add multiple repositories from remote sources or local directories. Once added, project folders can be directly selected for scanning using the **Current Source** dropdown.

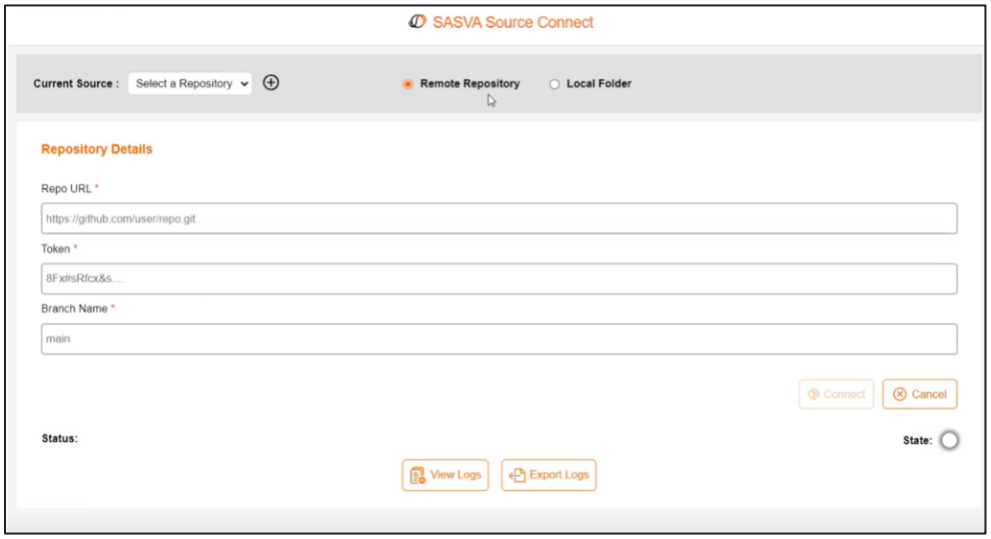
© 2025 Persistent Systems Ltd. All rights reserved.

Adding Remote Repositories

Workflow steps:

1. Click the **Source Connect** label on the left side panel of the UI.
2. The **SASVA Source Connect** page will appear.
3. Go to the **Current Source** label and click the + icon.
4. Select the Remote Repository radio button and enter the following required information:
    - Repo URL
    - Access Token
    - Branch name

Note: For instructions on generating **Access Token**, see the Reference section.



5. Click the **Connect** button to initiate the connection process.
6. A **Message Center** window will appear, displaying the details of the tasks performed.
   You can also open the Message Center by clicking the icon.
7. Monitor the displayed **Status** and **State** to track the progress of the connection and ensure successful completion.
8. Select the **View Logs** button to review the connection details and activity logs, if needed.
9. Select the **Export Logs** button to download and save the connection and activity logs for troubleshooting purposes.
10. Once the repository connection is successful, the interface will display:
    - Source
    - Branch
    - Path

Perform project scan

Workflow steps:

1. Once the connection is successfully established, the repository will appear in the **Current Source** dropdown.

© 2025 Persistent Systems Ltd. All rights reserved.

2. Select the repository you want to scan from the dropdown.
3. Click the **Connect to SASVA** button to start the project scan.

4. Click the **Show Inclusion List** button to specify which file formats should be included in the scan. If no file formats are added, the scan will include all files by default.
5. Click the **View Pdf Report** button to open the scan report.
6. To download the report, click the **Download Pdf Report** button.

Adding Local Repository

1. Click the **Source Connect** label on the left side panel of the UI.
2. The **SASVA Source Connect** page will appear.
3. Go to the **Current Source** label and click the + icon.
4. Select the **Local Folder** radio button.

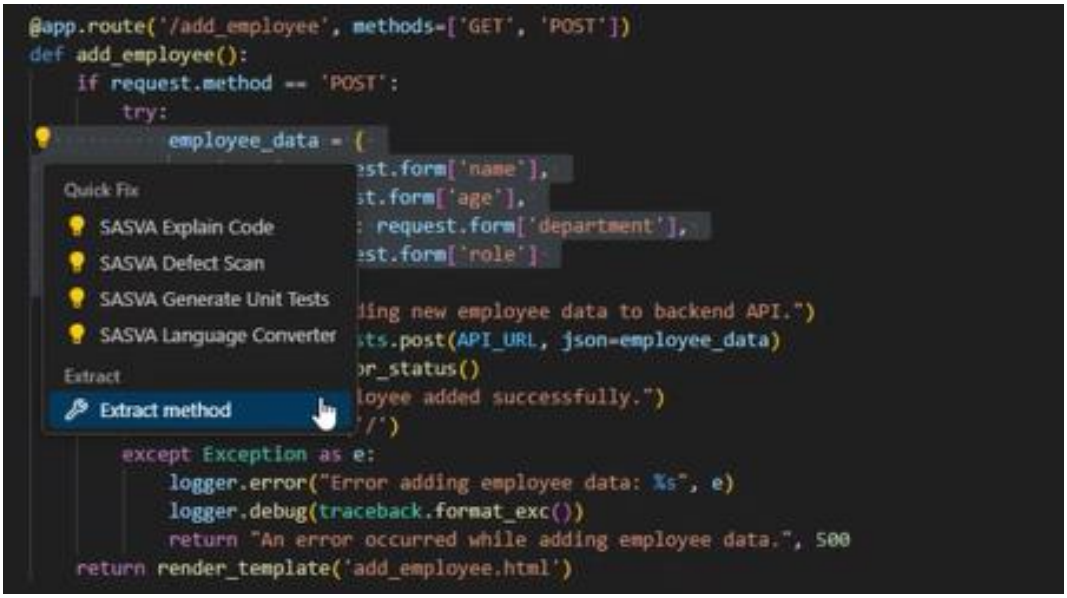
5. Click the **Browse Folder** button.
6. Select the project folder you want to import into IDE.
7. Click the **Add** button. The project will be added to the **Current Source** drop-down.
8. You can select the project to be scanned from the **Current Source** dropdown.
9. Click the **Connect to SASVA** button to begin the project scan.
10. Click the **Show Inclusion List** button to specify which file formats should be included in the scan. If no file formats are added, the scan will include all files by default.
11. Click the **View Pdf Report** button to open the scan report.
12. To download the report, click the **Download Pdf Report** button.

SASVA Toolkit

SASVA Toolkit provides quick fix actions which is typically indicated by the light bulb icon in the SASVA Plugin. It offers developers contextual recommendations to enhance code quality, fix issues efficiently, and boost productivity for the selected piece of code with minimal effort. Perform the following steps to use the SASVA Toolkit option in SASVA Plugin:

© 2025 Persistent Systems Ltd. All rights reserved.

1. In the editor, select the piece of code where you want to check issues or improvements.
2. After selecting the code, click the light bulb icon to open the QuickFix menu.



3. The QuickFix menu will display a list of actions for your selected code, such as:
    - SASVA Explain Code
    - SASVA Defect Scan
    - SASVA Generate Unit Tests
    - SASVA Language Converter

SASVA Explain Code

The SASVA Explain Code option provides insights into what a specific code segment does. This option is useful for understanding complex code or unfamiliar code segments.
Steps to use SASVA Explain Code in QuickFix:

1. Select the specific piece of code.
2. Click the QuickFix icon to open the QuickFix menu.
3. Select **SASVA Explain Code** from the QuickFix menu.
4. SASVA Plugin will generate an explanation for the selected code, which will appear in a Code Explanation window.
5. If needed, you can save the explanation by clicking the **Save** button and selecting a location to store it for future reference.

SASVA Defect Scan

The SASVA Defect Scan option performs a deeper scan for defects in the selected code. It helps in quickly identifying and addressing code issues, enhancing code quality and maintainability.
Steps to use SASVA Defect Scan in QuickFix:

1. Select the specific piece of code.

© 2025 Persistent Systems Ltd. All rights reserved.

2. Click the QuickFix icon to open the QuickFix menu.
3. Select **SASVA Defect Scan** from the QuickFix menu. It will initiate a scan to detect issues, errors or vulnerabilities.
4. Once the scan is complete, a defect scan report will display, listing all the detected issues along with the suggested fixes.
5. Click the **Save** button and choose a folder to save the defect scan report, if needed.
6. If you are not satisfied with the defect scan report or have any questions. You select the code, and it will appear in the SASVA Ask window. There you can enter your query to receive a response.

SASVA Generate Unit Tests

SASVA Generate Unit Tests option simplifies the process of creating unit tests. It ensures that your code is thoroughly tested with minimal manual effort.
Steps to use SASVA Generate Unit Tests in QuickFix:

1. Select the specific piece of code.
2. Click the QuickFix icon to open the QuickFix menu.
3. Select **SASVA Generate Unit Tests** from the QuickFix menu.
4. SASVA Plugin will automatically generate unit test report for the selected code.
5. Click the **Save** button and select a folder to save the unit test report, if needed.

SASVA Language Converter

The SASVA Language Converter option is used for quickly translating code into different programming languages, making it easier to work across multiple languages or platforms.
Steps to use SASVA Language Converter in QuickFix:

1. Select the specific piece of code, you want to convert to a different programming language.
2. Click the QuickFix icon to open the QuickFix menu.
3. Select the **SASVA Language Converter** from the QuickFix menu. This will open a panel with language conversion options.
4. Choose the programming language you wish to convert the code to, from the available list of options and submit. The converted code will appear in a new panel.
5. Click the **Save** button and choose a folder to store the code for future reference.

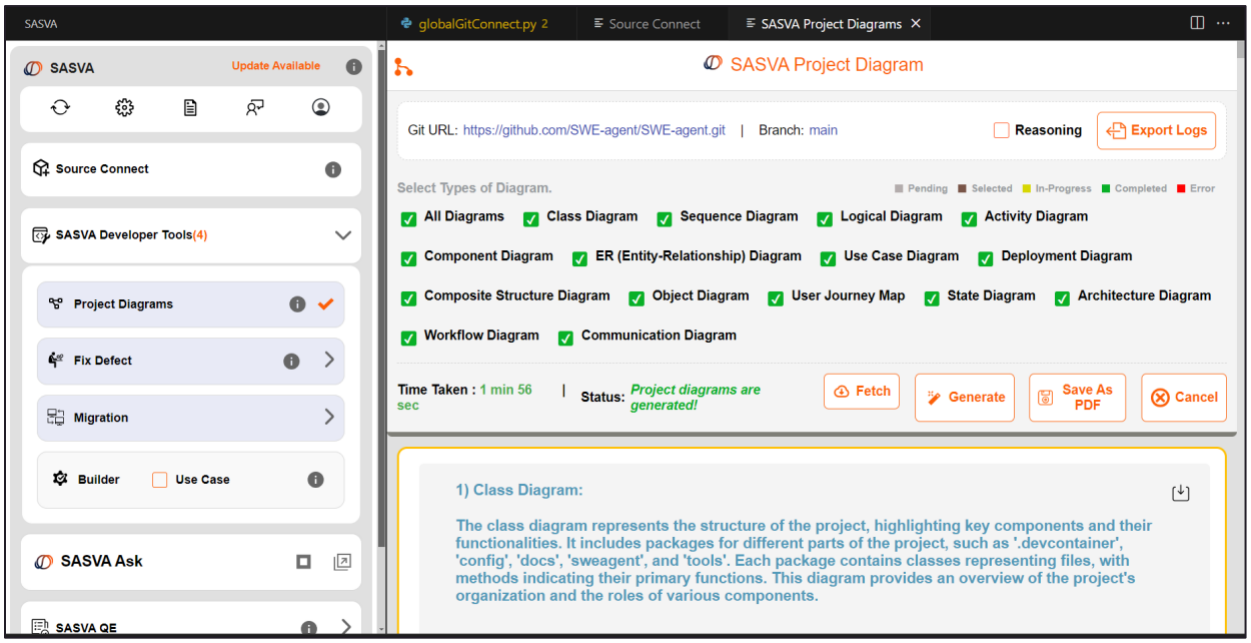
SASVA Developer Tools

Developer tools in SASVA Plugin refer to additional functionalities that extend the capabilities of SASVA Plugin to make the developer's tasks more efficient. To access the developer tool feature, you must establish a Repository connection.
Here is an overview and workflow of the Developer Tools feature available in the SASVA Plugin.

© 2025 Persistent Systems Ltd. All rights reserved.

Project Diagrams

The Project Diagram will provide a visual overview of the project's structure, components, and relationships, making it easier for developers to understand complex systems. A Project Diagram also helps in identifying project phases and dependencies which aids in better planning and resource allocation.



Here is the list of diagrams and their significance:

Diagram Name	Description
Class Diagram	A diagram that shows the classes and their relationships in your project, helping you understand the object-oriented design
Sequence Diagram	A diagram that shows the sequence of events and interactions between objects in your project, helping you understand the flow of your application
Logical Diagram	A diagram that shows the logical structure of your project, including the relationships between components and the data flow.
Activity Diagram	A diagram that shows the activities and tasks involved in your project, helping you understand the workflow and business processes.
Component Diagram	A diagram that shows the components and their relationships in your project, helping you understand the architecture and design.
ER (Entity-Relationship) Diagram	A diagram that shows the entities and their relationships in your project, helping you understand the data modeling and database design.
Use Case Diagram	A diagram that shows the use cases and their relationships in your project, helping you understand the functional requirements and user interactions.

© 2025 Persistent Systems Ltd. All rights reserved.

Deployment Diagram	A diagram that shows the deployment of your project, including the hardware and software components, and their relationships.
Composite Structure Diagram	A diagram that shows the composite structure of your project, including the components and their relationships.
Object Diagram	A diagram that shows the objects and their relationships in your project, helping you understand the object-oriented design and implementation.
User Journey Map	A diagram that shows the user's journey and interactions with your project, helping you understand the user experience and usability.
State Diagram	A diagram that shows the states and transitions of your project, helping you understand the behavior and functionality.
Architecture Diagram	A diagram that shows the overall architecture of your project, including the components, relationships, and interactions.
Workflow Diagram	A diagram that shows the workflow and business processes involved in your project, helping you understand the operations and management.
Communication Diagram	A diagram that shows the communication and interactions between components and objects in your project, helping you understand the data flow and integration.

Follow the given steps to generate, manage, and save project diagrams:

1. Click the **Project Diagrams** label.
2. The SASVA Project Diagrams window will appear.
3. (optional) Select the Reasoning checkbox to get a local response. If unchecked (default), responses will come from the SASVA Model.
4. In the displayed window you will see multiple types of diagram checkboxes.
5. Select the checkboxes corresponding to the diagrams you wish to generate.
6. From the **Project Structure** list, you can select relevant folders based on your project requirements to generate diagrams.
7. Click the **Generate** button.
8. For large repositories, you can select the **Fetch** option to quickly retrieve pre-saved diagrams.
9. During the generation process the color of the selected diagram checkboxes will change to indicate progress.
10. The time taken for generation and the status will be displayed.
11. Once completed, the generated diagrams will appear on the screen along with a brief description of each diagram.
12. Click the regenerate icon to regenerate the diagram, if the generated diagrams do not meet your expectations.
13. Click the save icon to save an individual diagram.
14. Click the **Save as PDF** button to save all the generated diagrams to a specified location.
15. Click the **Cancel** button to cancel the ongoing generation process.

© 2025 Persistent Systems Ltd. All rights reserved.

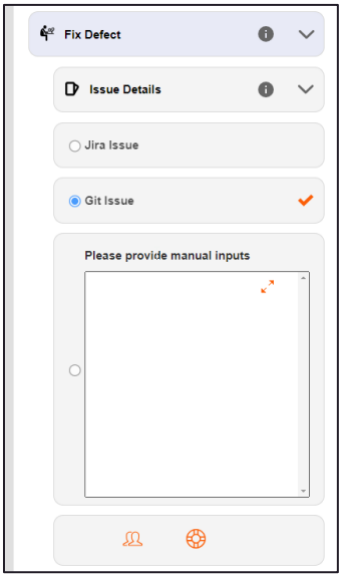
16. Select the **Export Logs** button to download and save activity logs for troubleshooting purposes.

Fix Defect

The Fix Defect feature assists developers in identifying, tracking, and resolving defects or issues in their code.
This feature provides seamless integration with **Jira**, **Git**, and **Custom** Issue-tracking systems.
It offers two options for resolving defects: **SASVA Assisted** and **SASVA Automated** .
It supports two response models: the SASVA Model for standard answers and the SASVA Reasoning Model for local responses.
(optional) Select the Reasoning checkbox to get a local response. If unchecked (default), responses will come from the SASVA Model.

Workflow Steps:

1. Click the **Fix Defect** label, and the **Issue Details** drop-down will appear.
2. Click the **Issue Details** to open the drop-down menu.



3. Choose one of the options given in the drop-down menu based on your defect source.
   - **Jira Issue**- Select to view or resolve issues tracked in Jira.
   - **Git Issue**- Select to fix issues related to code repositories in Git.
   - **Custom Issue**- Select to provide manual inputs.

4. Select the resolution method as per your requirement, after selecting the issue source.
   Resolution methods:
   SASVA Assisted - This method provides step-by-step guidance to resolve the defect manually.

© 2025 Persistent Systems Ltd. All rights reserved.

SASVA Automated - This method generates fixes for the issues and automatically applies them.

Assisted Fix

Git Issue

1. Click the **Git Issue** radio button and enter the required information including:
    - Git URL
    - Access Token

Note: For instructions on generating a Git Personal Access Token (PAT), see the Reference section.

2. Click **OK** and wait for the orange color tick mark which validates the issue from Git.
3. Click the **SASVA Assisted** button to open the **SASVA Issue Details** page, where you can view the issue details that need to be fixed.
4. Observe the state color. If the state color is **Red**, the project is not ready for the defect fix process.
5. Click the **Project Size** check button to reduce the file size, and adjust the slider as required to exclude the unnecessary files.
6. Once the slider is adjusted, click the **Continue** button.
7. If the state color changes to **Green**, the project is ready for the defect fix process.
8. You can view the number of files selected for the defect fix process, as well as those that were excluded, in the **Message Center** window.
9. Click the **Continue to Fix** button to open the **SASVA Defect Fix Execution** page to begin the defect resolution process.
   Note:
    - The execution page will provide one defect fix execution step at a time.
    - Some steps may require manual actions, while others may be automated.
10. Follow the step-by-step instructions provided.
11. Click the **Step 1 Completed** button to move to the next step.
12. Continue this process for all remaining steps in the defect fix plan.
13. If you encounter any issue during execution, click the **Ask Help** button for assistance.
    Steps for Using the **Ask Help** option during defect fix execution.
    a. During the defect fix execution, locate and click the **Ask Help** button.
    b. Upon clicking, a text box will appear, where you can type in any queries you have regarding the execution steps.
    c. For questions or instructions related to a specific step:
        i. Select the radio button corresponding to that step.
        ii. Enter your instructions or questions in the provided field.
        iii. Wait for the response in the same interface.
        iv. By clicking the copy icon , you can copy the response.

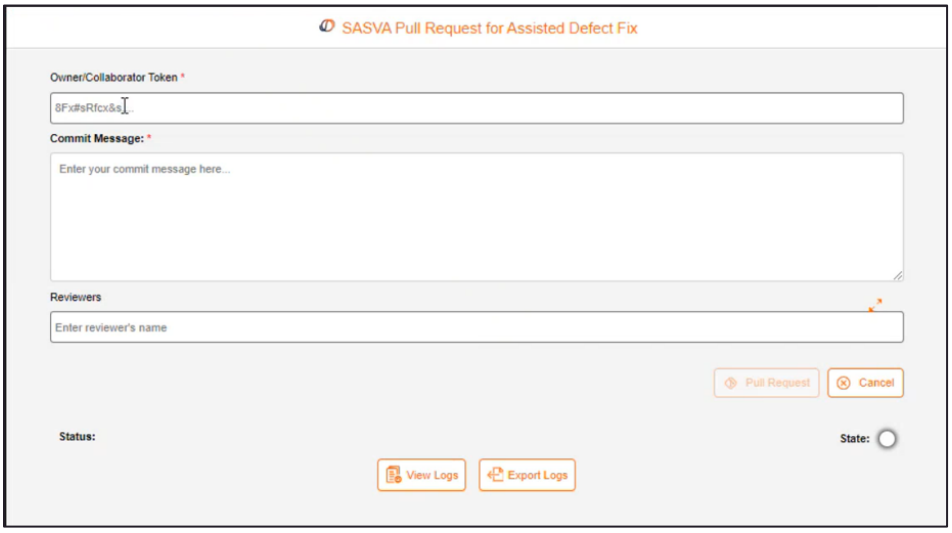
© 2025 Persistent Systems Ltd. All rights reserved.

d. If your question is regarding a specific piece of code:
    i. Click the **Ask Code** checkbox.
    ii. Upon clicking, you will be prompted to choose the file.
    iii. Navigate through the files and select the relevant code file.
    iv. Enter your question for the chosen piece of code.
    You will get the response which will help you proceed with the defect fix execution.
    v. Click the 

button to insert the response directly into the selected file.

14. Once all steps are completed and verified, the issue will be fixed.
15. A **SASVA Status message** pop-up will appear, prompting you to confirm whether you want to create a Pull Request (PR).

16. Click **Yes** to continue or click **No** to close.
17. Once you click **Yes**, you will be redirected to the Pull Request page. Enter the required information in the provided fields.



18. Click the **Pull Request** button to create the pull request.
19. Monitor the displayed **Status** and **State** to track the progress and ensure successful completion.
20. Select the **View Logs** button to review the details and activity logs.
21. Select the **Export Logs** button to download and save the activity logs for troubleshooting purposes.
22. A tick mark will appear on **SASVA Assisted**, indicating the issue is resolved.

JIRA Issue

1. Click the **JIRA Issue** and enter the required information including:

© 2025 Persistent Systems Ltd. All rights reserved.

- JIRA Issue URL
- Email id
- JIRA Token

Note: For instructions on generating a **Jira Token** (API Token), see the Reference section.

2. Click **OK**, and the issue will be captured.
3. Click the **SASVA Assisted** button to open the **SASVA Issue Details** page, where you can view the issue to be fixed, based on the associated JIRA ticket and its description.
4. Observe the state color. If the state color is **Red**, the project is not ready for the defect fix process.
5. Click the **Project Size** check button to reduce the file size, and adjust the slider as required to exclude the unnecessary files.
6. Once the slider is adjusted, click the **Continue** button.
7. If the state color changes to **Green**, the project is ready for the defect fix process.
8. You can view the number of files selected for the defect fix process, as well as those that were excluded, in the **Message Center** window.
9. Click the **Continue to Fix** button to open the **SASVA Defect Fix Execution** page to begin the defect resolution process.
   Note:
   - The execution page will provide one defect fix execution step at a time.
   - Some steps may require manual actions, while others may be automated.

10. Follow the step-by-step instructions provided.
11. Click the **Step 1 Completed** button to move to the next step.
12. Continue this process for all remaining steps in the defect fix plan.
13. If you encounter any issue during execution, click the **Ask Help** button for assistance.
    Steps for Using the **Ask Help** option during defect fix execution.
    a. During the defect fix execution, locate and click the **Ask Help** button.
    b. Upon clicking, a text box will appear, where you can type in any queries you have regarding the execution steps.
    c. For questions or instructions related to a specific step:
       i. Select the radio button corresponding to that step.
       ii. Enter your instructions or questions in the provided field.
       iii. Wait for the response in the same interface.
       iv. By clicking the copy icon, you can copy the response.
    d. If your question is regarding a specific piece of code:
       i. Click the **Ask Code** checkbox.
       ii. Upon clicking, you will be prompted to choose the file.
       iii. Navigate through the files and select the relevant code file.
       iv. Enter your question for the chosen piece of code.

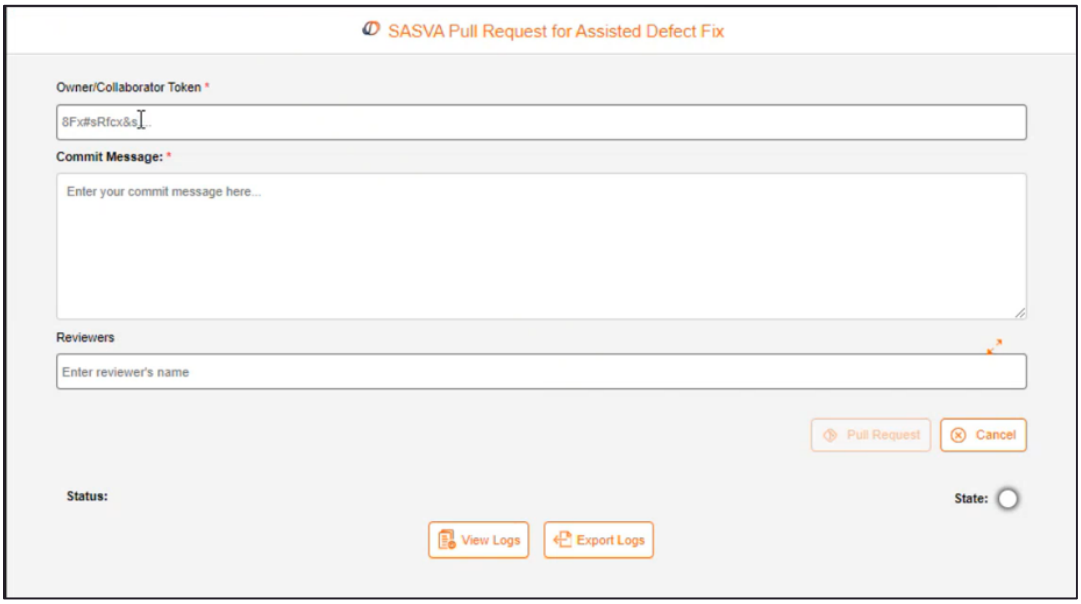
© 2025 Persistent Systems Ltd. All rights reserved.

You will get the response which will help you proceed with the defect fix execution.

v. Click the  button to insert the response directly into the selected file.

14. Once all steps are completed and verified, the issue will be fixed.
15. A SASVA Status message pop-up will appear, prompting you to confirm whether you want to create a Pull Request (PR).

16. Click **Yes** to continue or click **No** to close.
17. Once you click Yes, you will be redirected to the Pull Request page. Enter the required information in the fields provided.



18. Click the **Pull Request** button to create the pull request.
19. Monitor the displayed **Status** and **State** to track progress and ensure successful completion.
20. Select the **View Logs** button to review the details and activity logs.
21. Select the **Export Logs** button to download and save the activity logs for troubleshooting purposes
22. A tick mark will appear on **SASVA Assisted**, indicating the issue is resolved.

Custom Issue

1. Enter the input for which you need a fix in the provided text box.
2. Click the **SASVA Assisted** button to open the **SASVA Issue Details** page, where you can view the issue details that need to be fixed.
3. Observe the state color. If the state color is **Red**, the project is not ready for the defect fix process.

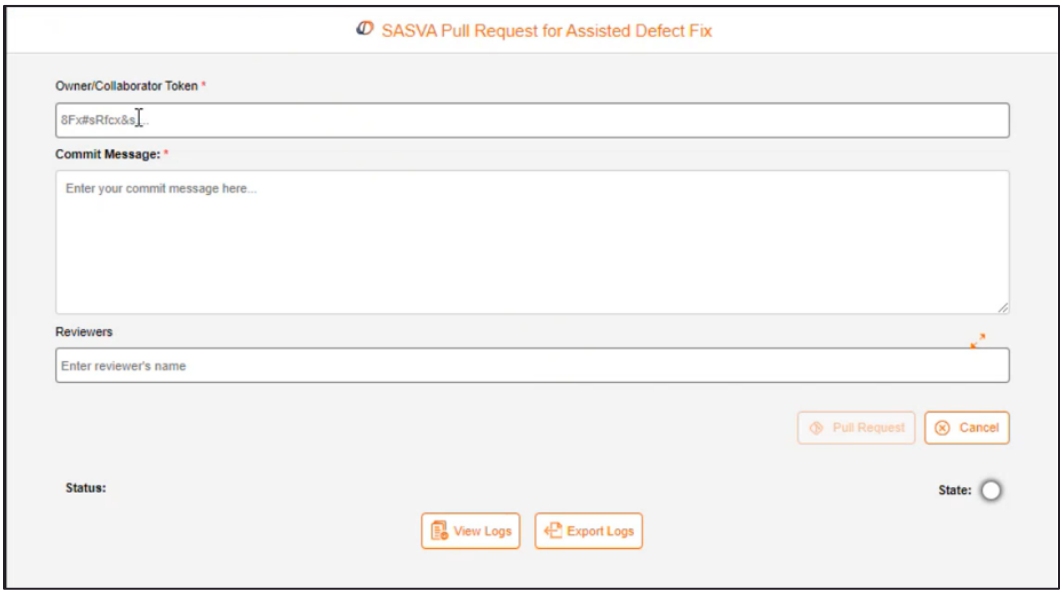
© 2025 Persistent Systems Ltd. All rights reserved.

4. Click the **Project Size** check button to reduce the file size, and adjust the slider as required to exclude the unnecessary files.
5. Once the slider is adjusted, click the **Continue** button.
6. If the state color changes to **Green**, the project is ready for the defect fix process.
7. You can view the number of files selected for the defect fix process, as well as those that were excluded, in the **Message Center** window.
8. Click the **Continue to Fix** button to open the **SASVA Defect Fix Execution** page to begin the defect resolution process.
   Note:
   - The execution page will provide one defect fix execution step at a time.
   - Some steps may require manual action, while others may be automated.
9. Follow the step-by-step instructions provided.
10. Click the **Step 1 Completed** button to move to the next step.
11. Continue this process for all remaining steps in the defect fix plan.
12. If you encounter any issue during execution, click the **Ask Help** button for assistance.
    Steps for Using the **Ask Help** option during defect fix execution.
    a. During the defect fix execution, locate and click the **Ask Help** button.
    b. Upon clicking, a text box will appear, where you can type in any queries you have regarding the execution steps.
    c. For questions or instructions related to a specific step:
       i. Select the radio button corresponding to that step.
       ii. Enter your instructions or questions in the provided field.
       iii. Wait for the response in the same interface.
       iv. By clicking the copy icon, you can copy the response.
    d. If your question is regarding a specific piece of code:
       i. Click the **Ask Code** checkbox.
       ii. Upon clicking, you will be prompted to choose the file.
       iii. Navigate through the files and select the relevant code file.
       iv. Enter your question for the chosen piece of code.
       v. You will get the response which will help you proceed with the defect fix execution.
       vi. Click the Apply button to insert the response directly into the selected file.
13. Once all steps are completed and verified, the issue will be fixed.
14. A **SASVA Status message** pop-up will appear, prompting you to confirm whether you want to create a Pull Request (PR).

15. Click **Yes** to continue or click **No** to close.

© 2025 Persistent Systems Ltd. All rights reserved.

16. Once you click **Yes**, you will be redirected to the Pull Request page. Enter the required information in the fields provided.



17. Click the **Pull Request** button to create the pull request.
18. Monitor the displayed **Status** and **State** to track the progress and ensure successful completion.
19. Select the **View Logs** button to review the details and activity logs.
20. Select the **Export Logs** button to download and save the activity logs for troubleshooting purposes
21. A tick mark will appear on SASVA Assisted, indicating the issue is resolved.

Automated Fix

Git Issue

1. Click the **Git issue** and enter the required information including:
    - Git URL
    - Access Token

Note: For instructions on generating a Git Personal AccessToken (PAT), see the Reference section.

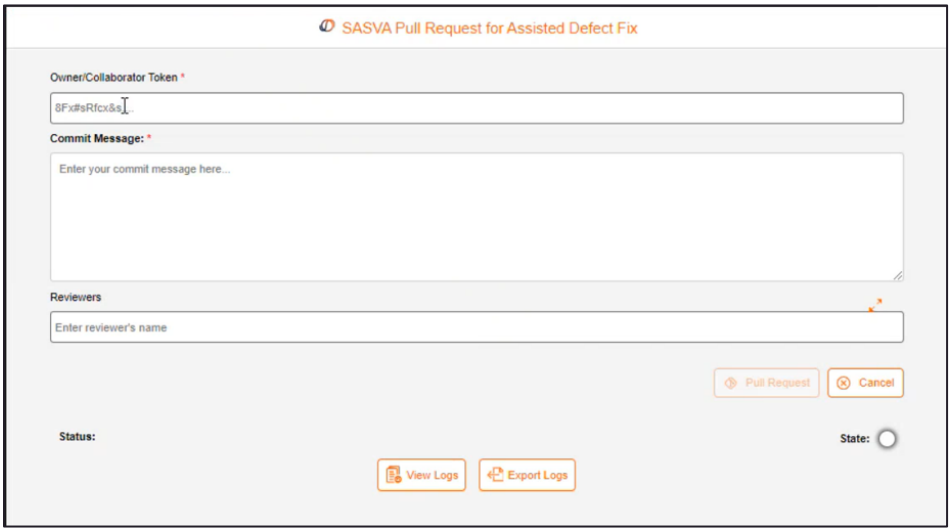
2. Click **OK**, and the issue will be captured.
3. Click the **SASVA Automated** button, and the **SASVA Automated Defect Fix** panel will appear on the screen. Where you can see the Git URL and branch.
4. Click the **Start** button to initiate the process.

© 2025 Persistent Systems Ltd. All rights reserved.

5. You can also check the **Status**, **State**, and **Time Required** to track the fix's progress.
6. (optional) Select the **Reasoning** checkbox to get a local response. If unchecked (default), responses will come from the SASVA Model.
7. Once the defect fix process is complete, a pop-up will appear displaying SASVA Automated Defect Fix Report, showing the updated files, newly created files, and the time taken to complete the fix.

8. Click the **Copy** button to copy the report if needed.
9. Once all steps are completed and verified, the issue will be fixed.
10. A **SASVA Status message** pop-up will appear, prompting you to confirm whether you want to create a Pull Request (PR).

11. Click **Yes** to continue or click **No** to close.
12. Once you click **Yes**, you will be redirected to the Pull Request page. Enter the required information in the fields provided.



13. Click the **Pull Request** button to create the pull request.
14. Monitor the displayed **Status** and **State** to track the progress and ensure successful completion.
15. Select the **View Logs** button to review the details and activity logs.
16. Select the **Export Logs** button to download and save the activity logs for troubleshooting purposes

© 2025 Persistent Systems Ltd. All rights reserved.

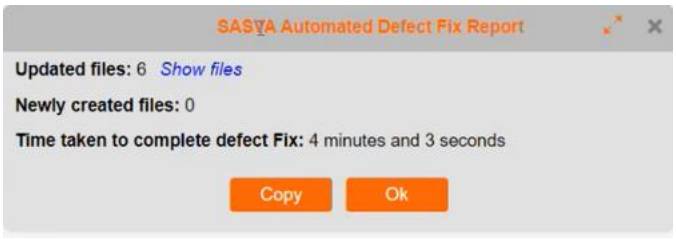
JIRA Issue

1. Click the **JIRA Issue** and enter the required information including:
    - JIRA Issue URL
    - Email id
    - JIRA Token

Note: For instructions on generating the **Jira Token** (API Token), see the Reference section.

2. Click **OK**, and the issue will be captured.
3. Click the **SASVA Automated** button, and the **SASVA Automated Defect Fix** panel will appear on the screen.
4. Click the **Start** button to initiate the process.

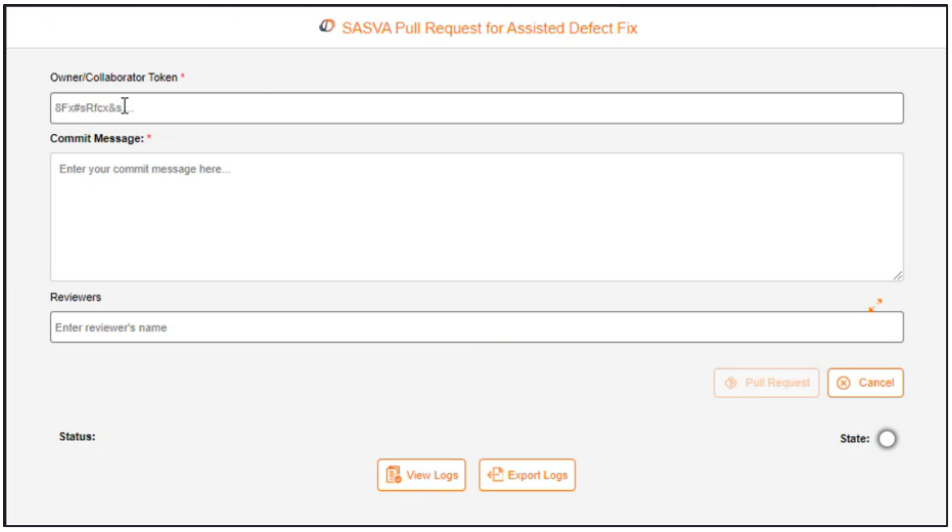
5. You can also check the **Status**, **State**, and **Time Required** to track the fix's progress.
6. (optional) Select the **Reasoning** checkbox to get a local response. If unchecked (default), responses will come from the SASVA Model.
7. Once the defect fix process is complete, a pop-up will appear displaying SASVA Automated Defect Fix Report, showing the updated files, newly created files, and the time taken to complete the fix.



8. Click the **Copy** button to copy the report if needed.
9. Once all steps are completed and verified, the issue will be fixed.
10. A **SASVA Status message** pop-up will appear, prompting you to confirm whether you want to create a Pull Request (PR).

11. Click **Yes** to continue or click **No** to close.
12. Once you click **Yes**, you will be redirected to the Pull Request page. Enter the required information in the provided fields.

© 2025 Persistent Systems Ltd. All rights reserved.

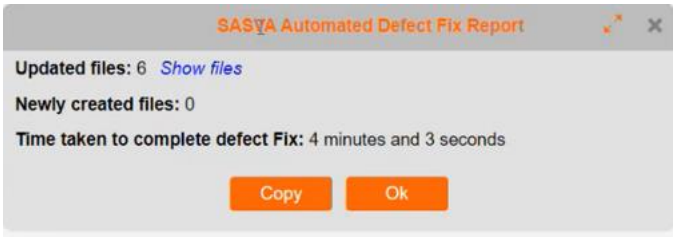


13. Click the **Pull Request** button to create the pull request.
14. Monitor the displayed **Status** and **State** to track the progress and ensure successful completion.
15. Select the **View Logs** button to review the details and activity logs.
16. Select the **Export Logs** button to download and save the activity logs for troubleshooting purposes

Custom Issue

1. Enter the input for which you need a fix in the provided text box.
2. Click the **SASVA Automated** button, and the **SASVA Automated Defect Fix** panel will appear on the screen. Where you can see the Git URL and branch.
3. Click the **Start** button to initiate the process.

4. You can also check the **Status**, **State**, and **Time Required** to track the fix's progress.
5. (optional) Select the **Reasoning** checkbox to get a local response. If unchecked (default), responses will come from the SASVA Model.
6. Once the defect fix process is complete, a pop-up will appear displaying the SASVA Automated Defect Fix Report, showing the updated files, newly created files, and the time taken to complete the fix.

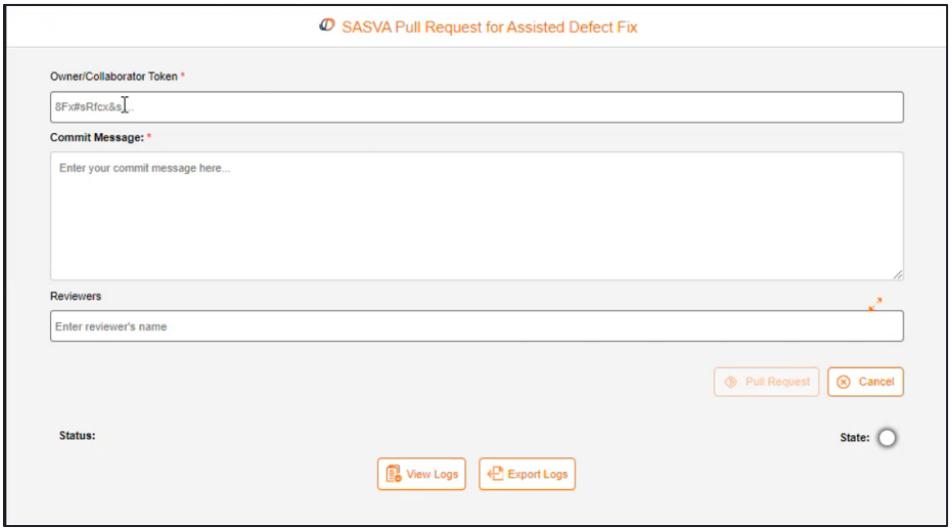


7. Click the **Copy** button to copy the report if needed.

© 2025 Persistent Systems Ltd. All rights reserved.

8. Once all steps are completed and verified, the issue will be fixed.
9. A **SASVA Status message** pop-up will appear, prompting you to confirm whether you want to create a Pull Request (PR).

10. Click **Yes** to continue or click **No** to close.
11. Once you click **Yes**, you will be redirected to the Pull Request page. Enter the required information in the fields provided.



12. Click the **Pull Request** button to create the pull request.
13. Monitor the displayed **Status** and **State** to track progress and ensure successful completion.
14. Select the **View Logs** button to review the details and activity logs.
15. Select the **Export Logs** button to download and save the activity logs for troubleshooting purposes

Migration

Prerequisite: Connect to the Repo, before moving forward with migration. Refer to the section Reference for instructions.

The Migration feature allows developers to transition their projects and data from one programming language to another. Using this feature developers can ensure a smooth transition while maintaining code quality and functionality.
It supports two response models: the SASVA Model for standard answers and the SASVA Reasoning Model for local responses.
(optional) Select the Reasoning checkbox to get a local response. If unchecked (default), responses will come from the SASVA Model.
This content outlines the migration process and provides options to customize the migration for optimal results.

Options to customize the project migration

© 2025 Persistent Systems Ltd. All rights reserved.

- Generate Inline: This option transforms code directly within the same file, ensuring that the migrated code creates a new file in the current folder. This approach avoids creating separate folders or files for the target language.

  If the checkbox is unchecked, it will create an additional folder with the original folder name in the format “foldername_target”.
- Interactive: This option is checked by default. In this option, the user can make decisions during the migration process. The user is prompted to review and approve changes or provide guidance during migration.
- Generate Test Cases: This option automatically creates test scripts to validate the functionality and correctness of the migrated code.
- All Files: This option ensures, the migration includes all files within the project including source code files, configuration files, resource files, documentation, and hidden files.

  The “All Files” option is checked by default, but if you uncheck it, a text box will appear where you can specify the names of the files you wish to exclude from the migration.
- Enhance Instructions: This option is checked by default. In this option, the user can get more detailed instructions and guidelines to understand the changes made during the migration.
- Latest Target: This option ensures the migrated project is updated to align with the most recent version of the target programming language.
- Read File: This option scans and validates the files in the source project before starting the migration.

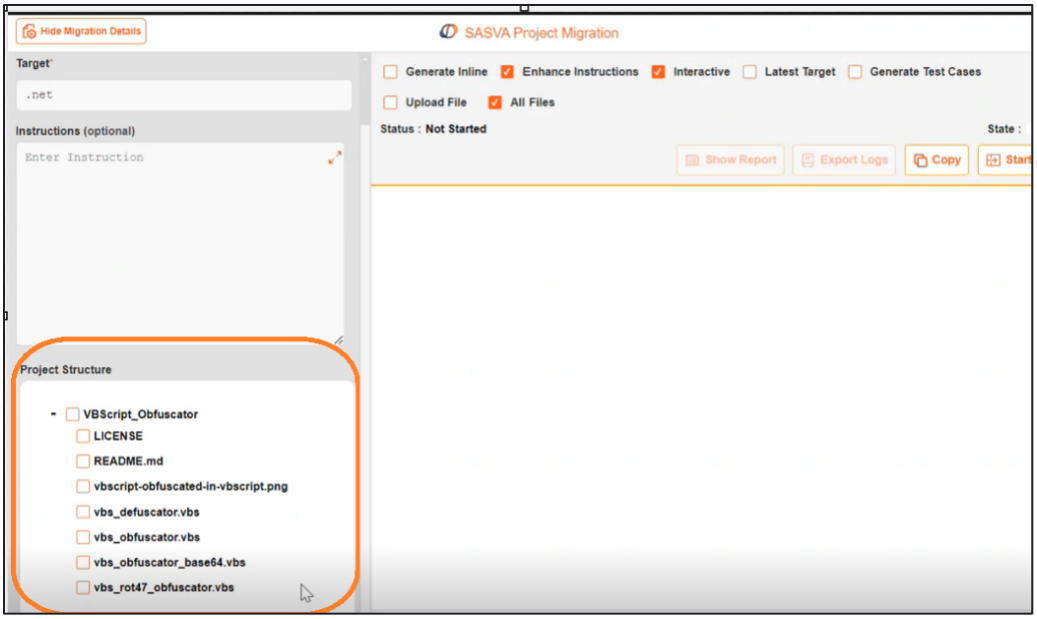
Project Migration

Project migration enables developers to migrate projects from one programming language to another. This process involves migrating the project's codebase, structure, and associated components to the target language while preserving the project's functionality.

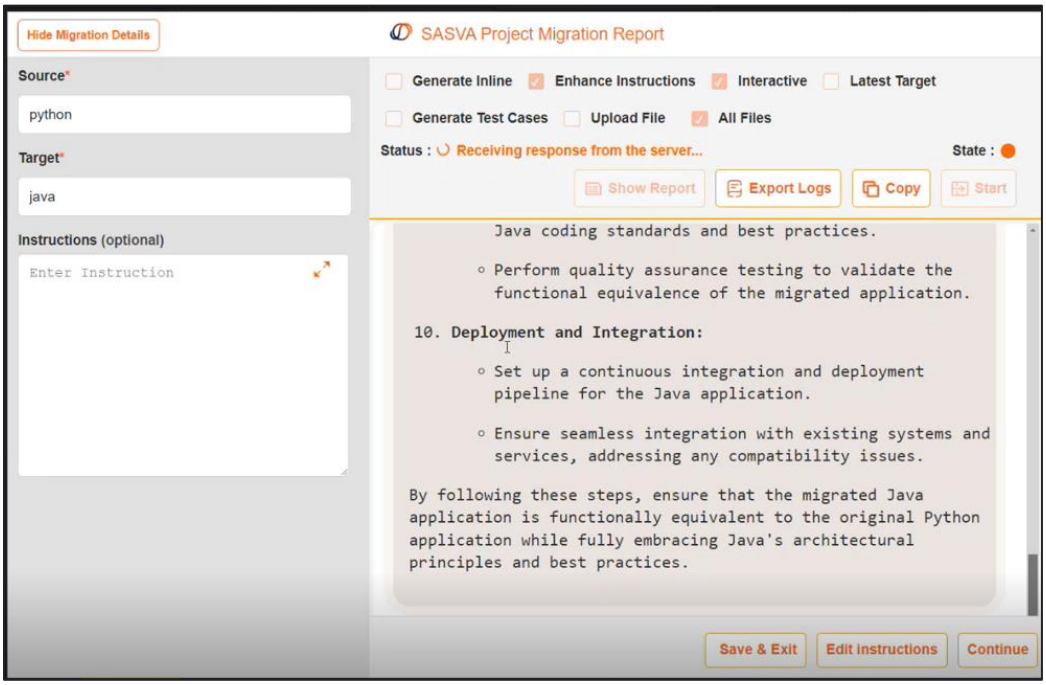
Steps to Execute the Project Migration

1. Click the **Project Migration** label.
2. Enter the source language and target language of the project in the appeared text boxes.
3. (optional)You can also provide instructions for migration based on your requirements.
4. Select the folder for project migration from the options given in the Project Structure list.

© 2025 Persistent Systems Ltd. All rights reserved.



5. Check the relevant options based on project needs.
6. Click the **Start** button to initiate the migration process.
7. The **State** and **Status** indicators will display the progress of the migration process.



8. Click the **Save & Exit** button to save the migration progress and resume later without starting over.
9. Enter the source and target language you specified earlier and click the **Start** button to resume the migration.
10. Click the **Hide Migration Details** button to expand the screen.

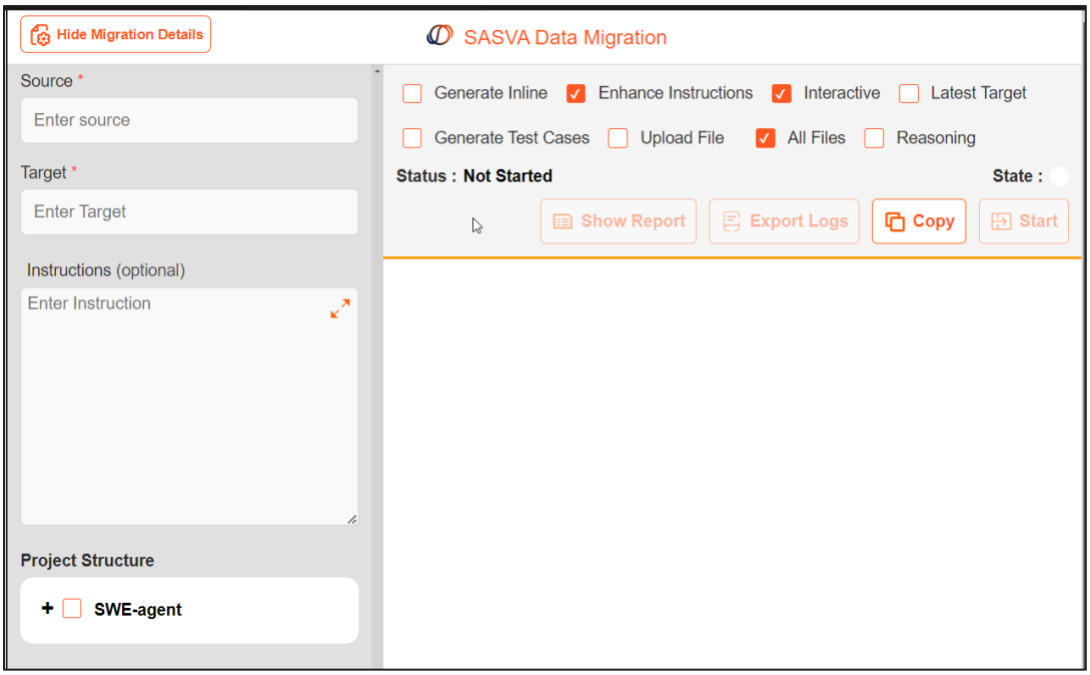
© 2025 Persistent Systems Ltd. All rights reserved.

11. Click the **Edit Instructions** button if you want to provide any extra input or modify the migration process.
12. Click the **Show Report** button to view the detailed migration report.
    (**Show Report** button will become enabled, once the migration process is complete.)
13. Click the **Export Logs** button to download and save the migration logs as a **.txt** file for future reference.

Data Migration

The Data Migration feature facilitates the seamless migration of data structures and pipelines from one programming language to another.
Steps to Execute Data Migration:

1. Click the **Data Migration** label.
2. Enter the source language and target language of the data in the appeared text boxes.
3. (optional)You can also provide instructions for migration based on your requirements.
4. Select the folder for data migration from the options given in the Project Structure list.



5. Check the relevant options based on project needs.
6. Click the **Start** button to initiate the migration process.
7. The **State** and **Status** indicators will display the progress of the migration process.
8. Click the **Hide Migration Details** button to expand the screen.
9. Click the **Edit Instructions** button if you want to provide any extra input or modify the migration process.
10. Click the **Show Report** button to view the detailed migration report.
    (**Show Report** button will become enabled, once the migration process is complete.)
11. Click the **Export Logs** button to download and save the migration logs as a **.txt** file for future reference.

© 2025 Persistent Systems Ltd. All rights reserved.

Builder

This AI-driven feature is designed to understand and generate human-like text, based on a given prompt. It can do end-to-end development based on your prompt. It supports input of any length. It works in multiple stages. Builder assists developers in the complete development process, from gathering requirements to writing code, testing, and deployment.
Follow the steps to use the Builder:

1. Open the **VS Code** editor.
2. Click the **SASVA Plugin** icon on the activity bar.
3. Click the **SASVA Developer Tools** drop-down.
4. Use Case checkbox:
    - If you check the **Use Case** checkbox you can enter the use case for the application, you want to develop in the text box.
    - If you do not check the Use Case checkbox and directly click the Builder label, the system will ask you to provide a description of the application you want to develop.
5. Click the **Builder** label. You will then be prompted to select a location/folder to save your work.
6. After receiving the use case or description, it will start analyzing the use case or description and generate a plan based on it.
7. Click the **Continue** button and enter the name of the app you want to develop.
8. Enter a detailed explanation of the app and click the **Continue** button.
9. Once your detailed explanation is submitted, and if the inferencing server requires further clarification, it will ask additional questions. Otherwise, it will proceed to the next step.
10. It will start planning the project architecture and prompt you to verify the availability of the required programming language, as well as any necessary applications or software before proceeding.
11. If not, install the required programming language on your system and click the **Continue** button to proceed, or click the **Skip** button to skip the question.
12. If you don't want to continue at that moment and prefer to resume later, you can click the **Save And Exit** button.
13. Clicking the **Last User Input** button will highlight the most recent input you entered.
14. Builder will break your idea into multiple tasks and provides description of each.
15. Click the **Yes** button, if you are satisfied with the tasks and their description provided by Builder.
16. If you are not satisfied with the tasks and description, click the **Edit** button to modify the description as needed.
17. After clicking the **Yes** button, Builder will provide you with step-by-step instructions to help you develop the project. It will also display the completion percentage after each stage of execution, provide a new ReadMe for that, and then proceed to the next task. During execution, it will display errors and provide solutions to fix them.

© 2025 Persistent Systems Ltd. All rights reserved.

18. If you click the **No** button, you will be prompted to describe what went wrong and add the necessary details.
19. The **Add Feature** button will be enabled at this stage.
20. To add a feature to the project, enter the desired input and click the **Add Feature** button.
21. To verify if the app is functioning as expected, you will need to manually confirm by following the instructions provided by Builder.
22. Click the **Yes** button, if the app is functioning as expected.
23. If an application is running and taking too long, you can click the **Terminate Command** button to stop the process and move further.

Builder operates in multiple stages, merges them, and generates the final output. You can review the output in the file location you selected to save your work.

SASVA QE

SASVA Quality Engineering (QE) is an AI-driven tool that automates the automation process, eliminating the need for automation experts and making test creation more accessible. It enhances productivity by 35-50% over traditional automation and 60-65% over manual testing. This tool is compatible with multiple frameworks and GenAI models. It seamlessly integrates with both new and existing automation initiatives, optimizing efficiency and reducing maintenance efforts.

SASVA QE tool with multiple capabilities:

1. Code Quality solution
2. Test Generator
3. UI Test Automation
4. API Test Automation
5. Test Data Management
6. ETL Test Automation
7. Test Script Migration

Code Quality Solution

The Code Quality feature enhances code readability, maintainability, and overall quality by analyzing code repositories and providing actionable suggestions. It integrates seamlessly with your development process, leveraging Gen AI for insightful evaluations and enabling quick identification of issues. Users can upload the default template file containing configured rules or add custom rules for different categories, input the path of a GitHub repository, and review and select files for analysis. The tool generates detailed reports summarizing analysis results, offering suggestions to improve code quality, and includes recommendations for each category provided in the input Excel.

Follow the steps to analyze code quality:

1. Click **SASVA QE** drop-down and select **Code Quality Solution**.
2. Download the test template by clicking **Download Template**.

© 2025 Persistent Systems Ltd. All rights reserved.

3. Open the downloaded sample template file which contains pre-configured rules or add custom rules and provide the required details.
4. **Upload** the template file to the tool.
5. Provide the **GitHub Repository Path**.
6. Select the **Repository Type**. For the Private repository provide the valid GitHub token
7. Select the **Testing framework** from the drop-down list.
8. Choose the **LLM Model** for analysis.
9. The **List file** helps you to choose the files from the repository.
10. Click **Generate Report** to generate a detailed report.

The report generated by your code quality tool includes essential details such as the file name and line number where issues are detected. It provides specific improvement suggestions and includes the relevant code snippet that needs to be changed. This detailed information helps developers quickly locate and address issues, ensuring the code remains high-quality and adheres to best practices.

Test Generator

The Test Generator is a powerful feature within the SASVA QE platform that enables automated creation of test cases based on various input formats. It streamlines the testing process by converting business requirements into structured test scenarios, reducing manual effort and improving consistency.

You can generate test cases using two methods:

There are 2 methods to generate test cases:

- **User Stories**: Upload a JSON file containing user stories and provide prompts to generate targeted test cases.
- **Agentic Test Cases Creation**: Upload business requirement documents (BRD) and either use Jira-exported data or manual prompts to generate test cases.

Both methods produce an Excel file containing the generated test scenarios, ready for review and implementation.

User Stories

Use this method when you have a JSON file containing user stories and want to generate test cases based on specific testing instructions. Follow the steps to generate the test cases:

8. Create the JSON file containing the user story details in the format mentioned below:

© 2025 Persistent Systems Ltd. All rights reserved.

```
{
    "release": {
      "releaseName": "sasvaTesting21",
      "releaseType": "Custom Release",
      "description": "create a gaming page",
      "subReleaseType": "",
      "reviewer": [],
      "gitConnectorId": 0,
      "jiraConnectorId": 0,
      "repoURL": "",
      "branchName": "main",
      "state": "new",
      "labels": "Execution",
      "shortDescription": "create a gaming page",
      "useSourceVersionRepo": false,
      "useProjectManagementTool": false,
      "draftId": 0
    },
    "customUseCases": [
      {
        "title": "Implement comprehensive test coverage to validate all
critical functionalities and edge cases of the gaming page.",
        "epics": [
          {
            "title": "Title of the eppic",
            "description": "description of epic",
              "devEfforts": {
              "expert": 1.4,
              "junior": 2.6,
              "medium": 1.92
            },
            "stories": [
              {
                "title": "title of the story",
                "description": "description of story",
                "allStages": false,
                "tasks": [
                  {
                    "title": "Title of the task",
                    "description": " descrption fo the task",
                    "manualEfforts": 0.12,
                    "devEfforts": {
                      "expert": 0.12,
                      "junior": 0.28,
                      "medium": 0.2
                    },
                    "aiEffort": "0.05",
                    "category": "functional"
```


© 2025 Persistent Systems Ltd. All rights reserved.

```
        },
        {
          "title": "Description of the task",
          "description": " descrption fo the task",
          "manualEfforts": 0.2,
          "devEfforts": {
            "expert": 0.12,
            "junior": 0.28,
            "medium": 0.2
          },
          "aiEffort": "0.07",
          "category": "technical"
        }
      ],
      "manualEfforts": 0.52,
      "devEfforts": {
        "expert": 0.4,
        "junior": 0.8,
        "medium": 0.6
      },
      "aiEffort": "0.22",
      "category": "both"
    }
  ]
 }
 ]
 }
 ]
}
```


9. Navigate to **SASVA QE** > **Test Generator**.
10. Choose the **User Stories** radio button.
11. **Browse and upload** your **JSON** file.
12. Enter your prompt in the text box (For Example: Generate positive test cases).
13. Click **Generate Testcase** to initiate the process.
14. Once generation is complete, click **Download Data** to obtain the Excel file containing the test scenarios.

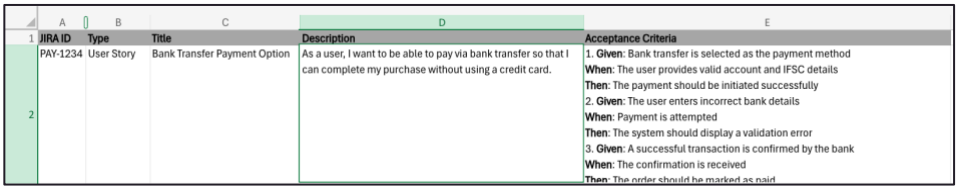
Agentic Test Cases Creation

Use this method when you have business requirement documents and want to generate test cases using either Jira data or manual input. Follow the steps to generate test scenarios using agentic test case creation:

1. Navigate to **SASVA QE** > **Test Generator**.

© 2025 Persistent Systems Ltd. All rights reserved.

2. Choose the **Agentic Test Cases Creation** radio button. The automated Test Case Generator window opens.
3. Browse and upload your knowledge documents such as BRD or requirement document in the pdf, docx or txt format.
4. Choose requirement input method:
   a. **Upload Jira Data**: You can generate the excel file from Jira containing the details of the user stories. Make sure that user stories must have the Title, Description and Acceptance Criteria.
      i. Export user stories from Jira into an Excel file. Ensure each story includes **Jira ID, Type, Title**, **Description**, and **Acceptance Criteria**.



	A	B	C	D	E
1	JIRA ID	Type	Title	Description	Acceptance Criteria
2	PAY-1234	User Story	Bank Transfer Payment Option	As a user, I want to be able to pay via bank transfer so that I can complete my purchase without using a credit card.	1. **Given**: Bank transfer is selected as the payment method **When**: The user provides valid account and IFSC details **Then**: The payment should be initiated successfully 2. **Given**: The user enters incorrect bank details **When**: Payment is attempted **Then**: The system should display a validation error 3. **Given**: A successful transaction is confirmed by the bank **When**: The confirmation is received **Then**: The order should be marked as paid

      I. Browse and **upload** the excel file.
      II. Click **Generate Test Cases for All Rows**.
      III. Once complete, click **Download All Test Cases** to download generated output. The excel file containing test scenarios will be downloaded on your computer.
   b. **Enter test cases requirement manually**:
      I. Select **Enter test cases requirement manually** radio button. The prompt window appears.
      II. Provide the **prompt** and press enter. (For Example: I want to generate testcases related to "Conceptual workflow of a VOP Request and Related Response" along with details for its Step1. Create multiple test cases for the following scenario: A PSU initiates a SEPA instant credit transfer or SEPA credit transfer to another PSU holding a Payment Account in SEPA. Include scenarios where the Payment Account Number and Payment Counterparty Name are provided by either the Requester or a PISP, ensuring accurate information. Consider instances where the Payment Counterparty is a legal entity identified by a fiscal number, VAT number, or LEI instead of a name. The Requesting PSP must verify all details subject to exemptions.)
      III. Review the generated requirement details.
      IV. If the requirement is satisfactory, click **Approve Requirements**.
      V. If not, enter your feedback in the **Enter your feedback on the requirements** text box and click **Send Requirement Feedback**. The system will update the requirements based on your input.
      VI. Review the updated requirement and click **Approve Requirements**.
      VII. Review the generated test cases.
      VIII. If the test cases are satisfactory, click **Approve Test Cases**.

© 2025 Persistent Systems Ltd. All rights reserved.

IX. If not, enter your feedback in the **Enter your feedback or the** text box and click **Send Feedback**. The system will regenerate the test cases accordingly.

X. Once approved, click **Download Updated Test Cases as Excel**. The excel file containing test scenarios will be downloaded on your computer.

UI Test Automation

The UI Test Automation feature automates the creation of test scripts for UI testing across various frameworks and languages. You can input an object repository and test case scenarios in Excel format, and the UI Test Automation will create optimized scripts tailored to your chosen language and framework. For projects with existing scripts, the solution can utilize them to generate new scripts that extend and enhance the current ones, ensuring a thorough and efficient testing process.
There are 2 types of automation:

- Automated GenUI Tool
- Global State Mapping

Supported Frameworks and Languages

Here is the list of supported languages:

- Selenium
  - Java
  - Python
  - .Net
- Selenium with Cucumber BDD
- Robot Framework
- Cypress
- Playwright
- Jest
- WebDriverIO

Automated GenUI Tool

This feature uses AI to generate test scenarios for your project.
Follow the steps to create a UI automation framework:

1. Click the **SASVA QE** drop-down and select **UI Test Automation**.
2. Choose the **Automated GenUI Tool** radio button.
3. Choose and **Upload** the Object Repository file. An object repository is a centralized storage of locators. It contains information about the elements of a web page. You can upload the existing Object Repository file or create one using the Selenium IDE plugin.
4. Upload the **Excel** file containing English language test scenarios.
5. Select **LLM Model** from the list.
6. Select **Framework** for the automation.

© 2025 Persistent Systems Ltd. All rights reserved.

a. Select the **Language** for automation from the drop-down.

b. For **Robot** and **Selenium** frameworks, Select POM option appears. Choose **POM** or **NOPOM** option as per requirement. Make sure to follow below example to create object files and test case excel.

\ For POM:

- Create and upload the POM object repository file.

  For example:

POM_Obj_repo_sample.json

```
{
  "url": "https:<your_test_url.com>",
  "obj_repo": [
    {
      "BasePage": {
        "commands": [
          {
            "name": "profile_dropdown",
            "target": "//*[@id='dropdownToggle']"
          },
          {
            "name": "signout_button",
            "target": "//*[@id='rfxHeaderSignOutButton']"
          }
        ]
      }
    },
    {
      "LoginPage": {
        "commands": [
          {
            "name": "username",
            "target": "//input[@placeholder='Enter username']"
          },
          {
            "name": "password",
            "target": "//input[@placeholder='Enter password']"
          },
```

- Create and upload the POM Excel File:

  For example:

© 2025 Persistent Systems Ltd. All rights reserved.

	Scenarios -	Prompts
1	Scenario 1-	1. Open Application Open this url in chrome browser- "https:<Your_test_url.com>" Maximize the window 2. In LoginPage, Login To The Application as Ess Associate01 Of Store1 Click on username, Enter {username} Click on password, Enter {password} Click on login button to login 3. In DashboardPage, Navigate To ESS Request Page On Web Hover on Down Arrow Click on Request Calendar Menu 4. In ESSPage, Add Day Off Request From Request Calendar Click On Add Request Click on Request Type and Select {Request Type} Select Request Day Off Week Start Date as PW2-D3 Select Request Day Off Week End Date as PW2-D3 Select Reason Type For Day Off as {Reason} Click on Save Button Click on Clear Button 5. Logout Click on Profile Dropdown Click on Sign Out Button

sr no	username	password	Planning Week(MM/DD/YYYY)	Effective Date	Request Type	Reason
1	1001	abc@123	7/6/2025	7/15/2025	Day Off	Paid Vacation
2	1002	abc@123	7/6/2025	7/15/2025		

\ For NO POM:

- Create and upload the NO POM object repository file.

  For example:

NO_POM_Obj_repo.json

```
{
  "url": "https:<your_test_url.com>",
  "tests": [
    {
      "commands": [
        {
          "name": "maximize the window",
          "target": "maximize",
          "targets": []
        },
        {
          "name": "username",
          "target": "xpath=//input[@name='email']"
        },
        {
          "name": "password",
          "target": "xpath=//input[@name='password']"
        },
        {
          "name": "login",
          "target": "xpath=//button[normalize-space()='Sign In']"
        }
      ]
    },
    {
      "commands": [
        {
          "name": "Users Menu",
          "target": "xpath=//*[@id='PSMA-Dashboard-lib']/div[1]/div/di
        },
```

- Create and upload the NO POM Excel File:

© 2025 Persistent Systems Ltd. All rights reserved.

	Scenarios -	Prompts
1	Scenario 1-	1. Chromedriver path is C:\drivers\chromedriver.exe. Open this url in chrome browser- "https://<Your test_URL>.com/" 2. Maximize the window 3. Click on username, Enter {username} by send keys 4. Click on password, Enter {correct_password} by send keys 5. Click on login button 6. Add wait of 5 secs after each step. 8. Exit
2	Scenario 2-	1. Click on Users in toolbar. 2. Add wait of 5 secs. 3. Click on Add Users button and validate the form header is = "Add User". 4. Fill the data as : FIRST_NAME = {first_name} LAST_NAME = {last_name} EMAILID = {email} PHONE_NO = {phone} 5. Select Role = {role} from role field dropdown menu. 6. Click on Submit button

	username	correct_password	wrong_password	first_name	last_name	email	phone	role
1	abc@testdomain.com	Sunday@123						
3	abc@testdomain.com	Sunday@125	abcdefg	Testuser3	adduser	testadduser@	#########	Manager

7. Click **Submit** to start the automation.
8. Click the **Download Scripts** to save the scripts on your computer. You can import the scripts in IDE to validate the automation.
9. After the successful execution of the generated scripts, you can access the test reports generated in the same folder in HTML format.

Global State Automation (GSM)

This feature is designed to generate a JSON file containing key-value pairs, where each function name from your input scripts becomes a key, and its corresponding one-line explanation serves as the value. This functionality allows for the automation of global state mapping by enabling users to upload existing object repository files and update them with new information about the application's functionality. Additionally, GSM facilitates the generation of global state mappings for existing test scripts, allowing users to upload these scripts, check the generated mappings, and extend the scripts for new scenarios. This process enhances the reusability and extensibility of test scripts by aligning them with the current functionality and methods.

Create GSM for new scripts

1. Click the **SASVA QE** drop-down and select **UI Test Automation**.
2. Choose the **Global State Mapping** radio button.
3. Choose and **Upload** the script files.
4. Click S**ubmit** to generate the GSM file for the uploaded scripts.

Update the Existing GSM files

1. Click the **SASVA QE** drop-down and select **UI Test Automation**.
2. Choose the **Global State Mapping** radio button.

© 2025 Persistent Systems Ltd. All rights reserved.

3. Choose and **Upload** the script files.
4. Check the box **Update Global State Mapping** to update the existing global state mapping scripts.
5. **Upload** the global state mapping file in which you wish to make changes.
6. Click **Submit** to enhance your existing scripts by considering the current functionality and reusing methods.

API Test Automation

This feature is developed to streamline API testing processes. Begin by uploading a Swagger document that outlines the API's functionality. The tool will then generate an Excel file with all the necessary tests based on the Swagger document. After reviewing and customizing these tests to fit your needs, upload them back into the tool. The solution will provide you with ready-to-use scripts to automate the testing process. Additionally, it generates a test report to reflect test execution status.

Follow the steps to set up your API test automation:

1. Click the **SASVA QE** drop-down and select **API Test Automation**.
2. Select the **LLM Model** from the drop-down.
3. Upload your **Swagger Open API specification File** in .txt format as input**.**
4. Select the **Type of Test Cases** from the drop-down that you want to generate.
   There are 20 types of functional test cases that can be generated. These include:
   a. **Status Code Validation for Valid Requests**: Verify that the API consistently returns the expected response status code, such as "200 OK" for valid and properly formatted requests.
   b. **Authentication Handling with Invalid Credentials**: Test the API's response when provided with invalid authentication credentials, ensuring it consistently returns a "401 Unauthorized" status code as expected.
   c. **Graceful Handling of Missing or Invalid Parameters**: Verify that the API handles missing or invalid request parameters gracefully and returns clear and user-friendly error messages that aid in troubleshooting.
   d. **Input Data Validation with Malformed Data**: Test the API's input validation by submitting various forms of malformed data, such as invalid email formats, and confirm that it properly rejects and responds to these inputs.
   e. **Timeout Handling under Load**: Confirm that the API correctly handles timeouts by simulating requests that take longer to process, ensuring that it remains responsive and does not hang.
   f. **Pagination Functionality Verification**: Test the API's pagination functionality by requesting specific pages of results and verifying that the responses contain the expected data and pagination information.
   g. **Concurrency Testing without Data Corruption**: Verify that the API handles concurrent requests from multiple users without data corruption or conflicts, ensuring data integrity.

© 2025 Persistent Systems Ltd. All rights reserved.

h. **Response Format Adherence (JSON/XML)**: Ensure that the API consistently returns responses in the specified format (e.g., JSON or XML) and adheres to the defined schema for data structure.
i. **Caching Mechanism Evaluation with Repeated Requests**: Evaluate the API's caching mechanism by making repeated requests and verifying that the cache headers are correctly set and honored.
j. **Rate Limiting Assessment**: Test the API's rate limiting by sending requests at a rate that exceeds the defined limits and checking for the expected rate-limiting responses, ensuring that limits are enforced.
k. **HTTP Method Support for CRUD Operations**: Verify that the API supports a variety of HTTP methods (GET, POST, PUT, DELETE) for Create, Read, Update, and Delete operations, and that it returns appropriate responses for each.
l. **Error Handling Capabilities for Meaningful Messages**: Evaluate the API's error-handling capabilities by intentionally causing errors, such as invalid inputs or unexpected situations, and confirm that it consistently returns meaningful error messages for troubleshooting.
m. **Conditional Request Handling (If-Modified-Since, If-None-Match)**: Test the API's support for conditional requests using headers like If-Modified-Since and If-None-Match, ensuring that responses are handled appropriately.
n. **Sorting and Filtering Validation for Resource Listings**: Verify that the API correctly sorts and filters resource listings based on specified parameters, maintaining data accuracy.
o. **Handling Long or Complex Data without Data Corruption**: Ensure that the API properly handles long or complex strings, such as URLs or text fields, without truncating or corrupting the data.
p. **Content Negotiation Support for Multiple Formats**: Test the API's support for content negotiation by specifying different Accept headers (e.g., JSON, XML) and verifying that the response format matches the requested format.
q. **Resource Not Found Handling (404 Not Found)**: Confirm that the API consistently returns the appropriate "404 Not Found" response when attempting to access a non-existent resource.
r. **Response Time Measurement for Various Requests**: Measure the API's response time for different types of requests to assess its performance and responsiveness.
s. **Handling Large Payloads (File Uploads)**: Verify that the API can handle large payloads, such as file uploads, without encountering errors or significant performance degradation.
t. **Compatibility with Client Libraries and SDKs**: Evaluate the API's compatibility with different client libraries or SDKs to ensure seamless integration with various platforms and programming languages.

5. Click **Generate Testcases** to start the test case creation.

© 2025 Persistent Systems Ltd. All rights reserved.

6. Click **Download** to save the test case file on the computer. Review and revise the generated test cases as needed. You can update the orchestration column in the template to chain multiple API calls.
7. Select the **Testing Framework** from the drop-down.
8. **Upload** the Test Case Template file.
9. Click **Generate Scripts**. Once the script generation is completed the new step appears to download scripts.
10. Click **Download Scripts** to save the scripts.
11. Execute scripts in the IDE environment and refer to the newly generated test report in HTML format.

Test Data Management

The Test Data Management (TDM) solution streamlines data management for precision testing, supporting MySQL and MongoDB databases. It uses an Intelligent GenAI Agent and LLM model to enhance data handling and testing. Key features include automated data mining, subsetting, and masking, reducing manual effort. The solution converts natural language into SQL queries for easier interaction. With robust support for multiple databases and efficient data masking, it ensures high-quality, secure data management. The user-friendly interface simplifies database detail entry and data process management.
This feature offers two main functions that are as follows:

- Synthetic Data Generation – Helps to generate test data for the testing.
- Pi Test Data Management (PiTDM) – Helps to perform data mining, subsetting, and masking operations.

Prerequisite:

\ Ensure that a valid database connection is established with My SQL or Mongo DB

Synthetic Data Generation

Follow the steps for execution:

1. Click the **SASVA QE** drop-down and select **Test Data Management**.
2. Select the **Synthetic Data Generation** radio button.
3. **Download** the **Test Cases Excel** file.
4. Update the Excel file according to the instructions provided in the **readme** tab. Specify the scenarios in the **Scenarios** tab for which the data needs to be generated, then re-upload the updated Excel file.
5. Click **Submit** to generate data.
   In the output Excel file, a new tab will be created for each scenario, containing synthetic data based on the input provided.

PiTDM

Follow the steps to perform test management operations:

1. Click the **SASVA QE** drop-down and select **Test Data Management**.

© 2025 Persistent Systems Ltd. All rights reserved.

2. Select the **PiTDM** radio button.
3. Select the Database from the drop-down menu.
   This PiTDM currently supports two types of databases:
   - My SQL
   - MongoDB.
4. Provide the database details as follows:
   - User Id - Enter your database user ID
   - Password- Enter your database password
   - Host- Enter the host IP address of your database
   - Port-Enter the port number used by your database
   - Database Name- Enter the name of your database.
5. A confirmation message appears upon validation of credentials.
6. Select the operation to be performed from the drop-down list options.

- **Data Quality Check** involves evaluating the accuracy, completeness, consistency, and reliability of data. This process ensures that the data meets the required standards and is fit for its intended use. Common tasks include:
  - Validation: Checking for data entry errors, missing values, and duplicates.
  - Standardization: Ensuring data follows a consistent format.
  - Accuracy Verification: Comparing data against trusted sources to confirm its correctness.
- **Data Mining** is the process of discovering patterns, correlations, and insights from large datasets using statistical and computational techniques. It helps in making informed decisions and predictions. Key activities include:
  - Classification: Assigning data into predefined categories.
  - Clustering: Grouping similar data points together.
  - Association Rule Learning: Identifying relationships between variables in the dataset.
- **Data Generation** involves creating synthetic data that mimics real-world data. This is useful for testing, training machine learning models, and other scenarios where real data is unavailable or sensitive. Techniques include:
  - Random Data Generation: Creating data based on random distributions.
  - Simulation: Generating data through simulated processes.
  - Data Augmentation: Enhancing existing data by adding variations.
- **Data Masking** is the process of hiding original data with modified content to protect sensitive information. This ensures data privacy and security while allowing the data to be used for testing and analysis. Methods include:
  - Substitution: Replacing sensitive data with fictional but realistic values.

© 2025 Persistent Systems Ltd. All rights reserved.

- Shuffling: Randomly rearranging data within the same dataset.
- Encryption: Converting data into a coded format that can only be read with a key.

\ **Data Unmasking** is the reverse process of data masking, where the original data is restored from the masked data. This is typically done when authorized users need to access sensitive information. It involves:

- Decryption: Converting encrypted data back to its original form.
- Reversal of Substitution: Replacing fictional values with the original data.
- Reordering: Restoring the original order of shuffled data.

7. Provide a clear and concise **prompt** that describes the specific action you want to perform on the data.
8. Click **Submit** to start creating the report. Once the report is generated, follow the new step that appears.
9. Click **Download File** to save the generated Excel file report on the computer.

ETL Test Automation

This advanced Gen AI-powered feature enhances ETL (Extract, Transform, Load) testing for SQL-to-SQL or Non-SQL to SQL databases. It streamlines the validation of source and target derivations, swiftly detecting any discrepancies. If inconsistencies are found, it identifies and highlights the inaccurate records, ensuring accurate and complete data transformation.

Prerequisites:

\ Ensure that you have two databases available: one for the Source and one for Target.

Follow the steps to perform the ETL test:

1. Click the **SASVA QE** drop-down and select **ETL Testing Solution**.
2. The **SASVA QE ETL Test Automation** page will appear.
3. Choose the operation type **SQL TO SQL** or **NOSQL to SQL** radio button.
4. In **Step1**, download the Excel template containing sheets for the source database, target database, and ETL testing scenarios.
5. You can update the information in the Excel template as per your requirements.
6. In **Step 2**, upload the template and click the **Submit** button.
7. Once submitted, **Step 3** will appear allowing you to download the generated scripts.

Test Script Migration

The Test Script Migration feature enables the seamless migration of test scripts from one language to another, ensuring compatibility across different frameworks.
It includes two main types:

\ Test Script IE (Intelligence Extraction)- Facilitates the migration of test scripts and object repositories to different frameworks, ensuring a consistent and efficient testing environment.

© 2025 Persistent Systems Ltd. All rights reserved.

- Function Intelligence Extraction- Efficiently extracts specific functions from scripts, simplifying the task of isolating and analyzing individual functions within larger codebases.

Test Script IE

The Test Script IE feature facilitates the seamless migration of test scripts and object repositories to different testing frameworks. This process ensures that your testing environment remains consistent and efficient, regardless of the framework used. Follow the steps to perform test script migration:

1. **Upload** the script files by clicking the box in step 1.
2. Select the appropriate option from the dropdown menu:
   - **Test Scripts**: These are scripts designed to test the functionality of your application. Select this option if your uploaded files are intended for testing purposes.
   - **Object Repo**: This refers to the repository where objects used in the test scripts are stored. Select this option if your uploaded files contain object definitions.
3. Choose the migration **Framework**. There are two frameworks:
   - **Protractor**: A framework for testing Angular and AngularJS applications. Select this if your scripts are designed for Angular applications.
   - **Codecept**: A modern end-to-end testing framework for web applications. Choose this if your scripts are for general web application testing.
4. Click **Submit** to start the migration.
5. Click **Download File** to download the migrated files.
   This feature simplifies adapting your test scripts to different frameworks and helps maintain a robust testing environment.

Function Intelligence Extraction

The Function Intelligence Extraction feature efficiently extracts specific functions from scripts. This process simplifies the task of isolating and analyzing individual functions within larger codebases.

Follow the steps for execution:

1. Click **Test Script Migration** in the SASVA QE tab.
2. Select the **Function Intelligence Extraction** radio button.
3. **Upload the script** from which you wish to extract the function, and then click **Submit**.
4. Once the process is completed, click the **Download File** button to download the function file.

This feature makes it easier to manage and utilize specific functions from scripts.

SASVA Ask

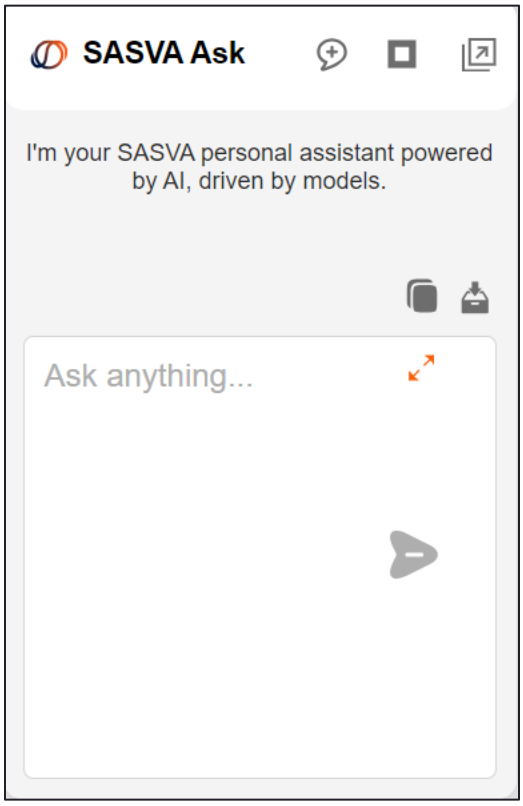
SASVA Ask provides a simple and intuitive interface where you can ask questions and receive responses, whether you need general information or project-specific insights. During

© 2025 Persistent Systems Ltd. All rights reserved.

development, if you face any kind of issue, you can take the help of SASVA Ask. It will deliver accurate results in a matter of seconds.
It supports two response models: the SASVA Model for standard answers and the SASVA Reasoning Model for local responses.

Follow these steps to use the SASVA Ask feature:

1. On the left side of the UI, click the **SASVA Ask** button to open the SASVA Ask (Smaller View) window.



2. Click the **Large Window** Icon to open the SASVA Ask (Larger View) window.

3. In the expanded view of SASVA Ask, two types of search options are available as radio buttons:

© 2025 Persistent Systems Ltd. All rights reserved.

- Generic
- Project Specific

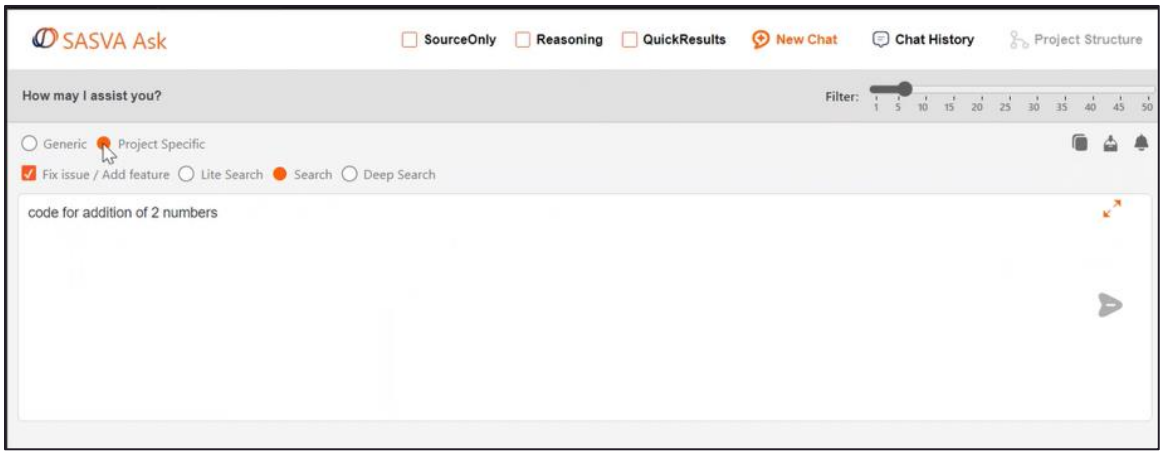
Generic

Generic search is intended for general, non-project-specific queries. It allows users to ask broad questions or explore various topics. The following options are enabled to enhance response quality. You can use it as per your requirements:

- **Reasoning**: Select the Reasoning checkbox to get a local response. If unchecked (default), responses will come from the SASVA Model.
- **New Chat**: The New Chat option allows you to start fresh conversations.
- **Chat History**: The Chat History feature saves your previous conversations, allowing you to revisit, review, and continue them at any time. It helps you keep track of your interactions and access useful information without needing to repeat questions.

Project Specific

Project Specific search is designed for queries related to a particular project context. It allows you to search within the scope of connected project data, ensuring responses are relevant and context aware.



When the Project Specific radio button is selected, several advanced options are enabled to enhance the response based on the selected project context.
The available options include:

- **Source Only**: When selected SASVA will use only the project source code files, ignoring all other project files.
- **Reasoning**: Select the Reasoning checkbox to get a local response. If unchecked (default), responses will come from the SASVA Model.
- **Quick Results**: When selected during Lite or Search, results are generated faster with minimal processing time. If the option is unchecked, the search will take longer but produce higher-quality results.

© 2025 Persistent Systems Ltd. All rights reserved.

- **Filter**: Using this Filter button, you can adjust the slider to select the desired number of files, with a maximum limit of 50 files.
- **Project Structure**: To get more precise responses, you can select a relevant folder before asking your questions.
- **Project Size**: To adjust the slider to compress the large files as per the requirements, to get more accurate results.
- **New Chat**: The New Chat option allows you to start fresh conversations.
- **Chat History**: The Chat History feature saves your previous conversations, allowing you to revisit, review, and continue them at any time. It helps you keep track of your interactions and access useful information without needing to repeat questions.
- **Fix Issue/Add feature**- By selecting the **Fix Issue/ Add feature** checkbox, you can resolve issues for the selected project file by providing the required input and you can also add a feature to the project. Once the Fix Issue/Add Feature checkbox is clicked, the Lite Search and Search option will be enabled.
- **Lite Search**- Select the **Lite Search** radio button to perform a quick surface-level search. Once you select Lite Search, the **Quick Results** checkbox and **Filter** slider options will be enabled.
- **Search**- Select the **Search** radio button to perform a more detailed analysis at a moderate level. Once you select Search, the **Quick Results** checkbox and **Filter** slider options will be enabled.
- **Deep Search**- Select the **Deep Search** radio button to perform an in-depth, comprehensive scan of the entire project, thoroughly analyzing all components. In Deep Search, the Quick Result and Filter options are disabled. The **Project Structure** and **Project Size** options are available.

UI Icons and their Functions

Icons	Use
Copy	Select to copy the complete chat history.
Save	Select to save as HTML file.
Message Center	Click the Bell icon to access the Message Center that shows the details of the task performed by the SASVA Ask.
DB button	Click the Data Base (DB) icon to save the response or output to the database. It allows you to resume from where you left off by accessing it through the chat history.
Reindex ReIndex	Click the ReIndex icon if you are not receiving responses as expected. This will refresh the scanned data stored locally and regenerate the response.

© 2025 Persistent Systems Ltd. All rights reserved.

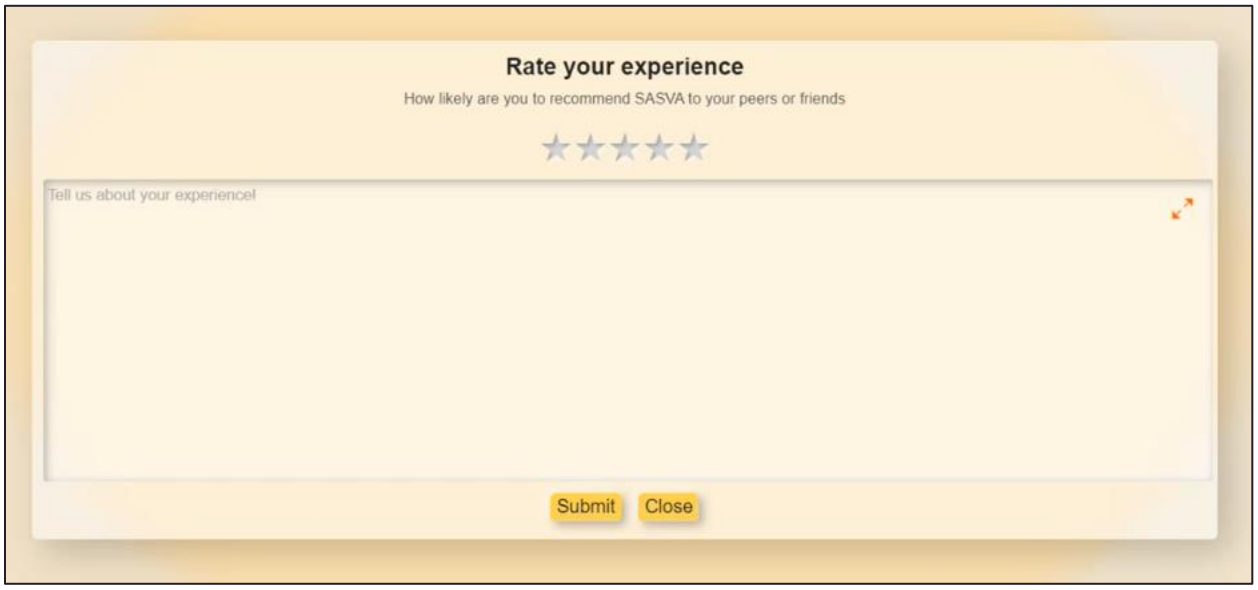
SASVA Logs

The SASVA Logs feature allows you to generate logs of various activities and features within the Plugin. These logs can then be saved in a compressed .zip file for easy storage, sharing, and troubleshooting purposes.
Steps to Save Logs:

1. Click the **SASVA Logs** icon. This will redirect you to the SASVA Logs page.
2. Check the features for which you want to save the generated logs on the SASVA Logs page.
3. Click the **Yes** button to proceed.
4. The system will prompt you to select a location to save the logs in a compressed .zip file.

SASVA Feedback

We appreciate your feedback on your experience with SASVA Plugin. Please click on the SASVA Feedback option, provide your input, and submit.



Manage Settings

Settings generally involve user preferences, environment setup, or server configurations. When you click on the **Settings** option, a settings page will open, displaying two types of settings:

Server Settings
Client Settings

Client and Server settings parameters must be obtained from the SASVA Support team.

Reference

© 2025 Persistent Systems Ltd. All rights reserved.

Update SASVA Plugin

This section outlines the steps to update the SASVA Plugin with the latest available package.
Follow the steps to update the SASVA Plugin:

1. In the left panel of the UI, locate and click on the **Update Available** link.

2. The **SASVA IDE Plugin Distribution** page will appear.
3. Select the IDE type from the **Select IDE Type** dropdown.
4. Download the latest available package for **Windows**.

Generating PAT Classic token for GitHub

Follow these steps to generate a Git Personal Access Token (PAT):

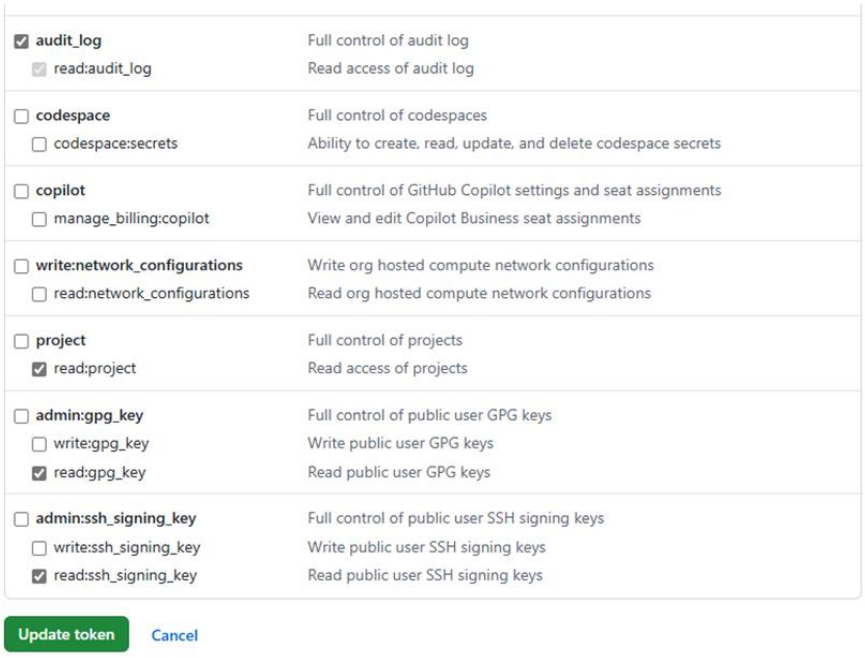
1. Log in to your **GitHub** account.
2. Click your **profile picture** in the top right corner.
3. Select **Settings**.
4. In the left sidebar, go to **Developer settings**.
5. Click **Personal access token**>**Token (classic).**
6. Click **Generate new token**>**Generate new token (Classic)**.
7. Provide a **note** to describe the purpose of the token.
8. Select an **expiration date** for the token.
9. Choose the necessary **scopes/permissions** for the token.

© 2025 Persistent Systems Ltd. All rights reserved.

Scopes define the access for personal tokens. Read more about OAuth scopes.

☑ **repo**	Full control of private repositories
☑ repo:status	Access commit status
☑ repo_deployment	Access deployment status
☑ public_repo	Access public repositories
☑ repo:invite	Access repository invitations
☑ security_events	Read and write security events
☐ **workflow**	Update GitHub Action workflows
☐ **write:packages**	Upload packages to GitHub Package Registry
☑ read:packages	Download packages from GitHub Package Registry
☐ **delete:packages**	Delete packages from GitHub Package Registry
☐ **admin:org**	Full control of orgs and teams, read and write org projects
☐ write:org	Read and write org and team membership, read and write org projects
☑ read:org	Read org and team membership, read org projects
☐ manage_runners:org	Manage org runners and runner groups
☐ **admin:public_key**	Full control of user public keys
☐ write:public_key	Write user public keys
☑ read:public_key	Read user public keys
☐ **admin:repo_hook**	Full control of repository hooks
☐ write:repo_hook	Write repository hooks
☑ read:repo_hook	Read repository hooks
☐ **admin:org_hook**	Full control of organization hooks
☐ **gist**	Create gists
☐ **notifications**	Access notifications
☐ **user**	Update ALL user data
☑ read:user	Read ALL user profile data
☐ user:email	Access user email addresses (read-only)
☐ user:follow	Follow and unfollow users
☐ **delete_repo**	Delete repositories
☐ **write:discussion**	Read and write team discussions
☑ read:discussion	Read team discussions
☐ **admin:enterprise**	Full control of enterprises
☐ manage_runners:enterprise	Manage enterprise runners and runner groups
☐ manage_billing:enterprise	Read and write enterprise billing data
☑ read:enterprise	Read enterprise profile data
☐ scim:enterprise	Provisioning of users and groups via SCIM

© 2025 Persistent Systems Ltd. All rights reserved.



10. Click **Generate token**.
11. Copy the generated token to your clipboard.

Generating a GitLab Personal Access Token

1. Log in to your GitLab account.
2. Go to your profile by selecting your avatar from the sidebar.
3. Select the **Edit Profile** option.
4. On the left sidebar, select **Access Tokens**.
5. Click on **Add a New Token**.
6. Provide a name for your token in the **Token Name** field.
7. (Optional) Add a short description to help identify the token's purpose.
8. Set an **expiration date** for the token.
9. Choose the appropriate **scopes** for the token.
10. Click **Create Token** to generate it.

    Note: Copy and save the token immediately. You won't be able to view it again once you leave the page.

API Token for Jira

Follow these steps to generate a Jira API Token:

1. Log in to Jira.
2. Click your profile picture.
3. Select **Profile.**
4. Click **Manage Your Account**.

© 2025 Persistent Systems Ltd. All rights reserved.

5. Navigate to the **Security** tab.
6. Click **Create and manage API tokens**.
7. Click **Create API token**.
8. Provide a label for the token.
9. Select an expiration date for the token.
10. Click **Create**.
11. Copy the token to your clipboard.

Documentation

The following are the other SASVA 2.1.2 release documents:

	Module name	Document name	Format
1	IDE Plugin documents	a. Release Notes for Visual Studio Code (this document) b. User Guide for Visual Studio Code	PDF
2	North Star documents	c. Release Notes d. User Guide	PDF
3	Release Management documents	e. Release Notes f. User Guide g. Administrator Guide	PDF

SASVA Support

For any assistance or queries, feel free to write to us at support@sasva.ai.

© 2025 Persistent Systems Ltd. All rights reserved.