

OpenText™ Core Application Security Plugin for IntelliJ IDEA

User Guide

Version : 25.4

PDF Generated on : December 30, 2025

Table of Contents

1. User Guide	1
1.1. Getting started	2
1.1.1. Product name changes	3
1.1.2. Requirements for using the Core Application Security Plugin for IntelliJ IDEA	4
1.1.3. Installing the Core Application Security Plugin for IntelliJ IDEA	5
1.1.4. Configuring the Core Application Security Plugin for IntelliJ IDEA	6
1.2. Uploading static code to OpenText Core Application Security	8
1.3. Uploading open source code to OpenText Core Application Security	21
1.4. Reviewing analysis results	26
1.4.1. Opening analysis results	27
1.4.2. Analysis results in the Core Application Security Plugin for IntelliJ IDEA	29
1.4.3. Reviewing issues	35
1.4.4. Auditing issues	37
1.4.5. Locating the source code associated with static scan issues	38
1.5. Auto-remediation with SAST Aviator	39
1.5.1. Best practices for auto-remediation	44

1. User Guide

Software Version: 25.4.0

Document Release Date: 25.4.0

Software Release Date: 25.4.0

1.1. Getting started

This help describes how to install the Core Application Security Plugin for IntelliJ IDEA and use it to upload code for static analysis to OpenText™ Core Application Security and open analysis results for remediation. This plugin works with Android Studio and other JetBrains IDEs as listed in the JetBrains Marketplace. The procedures in this guide reference the IntelliJ IDEA interface and the instructions might be slightly different for the other IDEs.

- [Product name changes](#)
- [Requirements for using the Core Application Security Plugin for IntelliJ IDEA](#)
- [Installing the Core Application Security Plugin for IntelliJ IDEA](#)
- [Configuring the Core Application Security Plugin for IntelliJ IDEA](#)

1.1.1. Product name changes

OpenText is in the process of changing the following product names:

Previous name	New name
Fortify Static Code Analyzer	OpenText™ Static Application Security Testing (OpenText SAST)
Fortify Software Security Center	OpenText™ Application Security
Fortify WebInspect	OpenText™ Dynamic Application Security Testing (OpenText DAST)
Fortify on Demand	OpenText™ Core Application Security
Debricked	OpenText™ Core Software Composition Analysis (OpenText Core SCA)
Fortify Applications and Tools	OpenText™ Application Security Tools

The product names have changed on product splash pages, mastheads, login pages, and other places where the product is identified. The name changes are intended to clarify product functionality and to better align the Fortify Software products with OpenText. In some cases, such as on the documentation title page, the old name might temporarily be included in parenthesis. You can expect to see more changes in future product releases.

1.1.2. Requirements for using the Core Application Security Plugin for IntelliJ IDEA

To use the Core Application Security Plugin for IntelliJ IDEA, you must have the following:

- Your OpenText™ OpenText Core Application Security login credentials or your SSO login URL if your organization has configured SSO for the tenant
- An API root URL.

For a list of data center API root URLs, see the *OpenText™ Core Application Security User Guide*.

- To upload your code to OpenText Core Application Security:
 - Your login account must have the Start Static Scan permission.
 - To have the Core Application Security Plugin for IntelliJ IDEA automatically package all the necessary dependencies and source code (including files required for a Debricked open source scan), you must have a locally installed OpenText™ Fortify ScanCentral SAST client version 22.1.2 or later and Core Application Security Plugin for IntelliJ IDEA version 23.1 or later.

You can download the Fortify ScanCentral SAST client from the OpenText Core Application Security Tools page. For installation instructions, see the README.txt file included in the downloaded ZIP.

- To upload Debricked open source scans to Core Application Security, ensure that the **Run Debricked Open Source Scan** option is selected in the application.

1.1.3. Installing the Core Application Security Plugin for IntelliJ IDEA

To install the IntelliJ plugin:

1. From the IDE, open the Settings dialog box as follows:
 - On Windows or Linux, select **File > Settings**.
 - On macOS, select **<IDE_name> > Preferences**.
2. In the left pane, select **Plugins**.
3. Select the **Marketplace** tab, and then in the search box type **OpenText Core Application Security**.
4. Click **Install**.
5. Click **OK**.

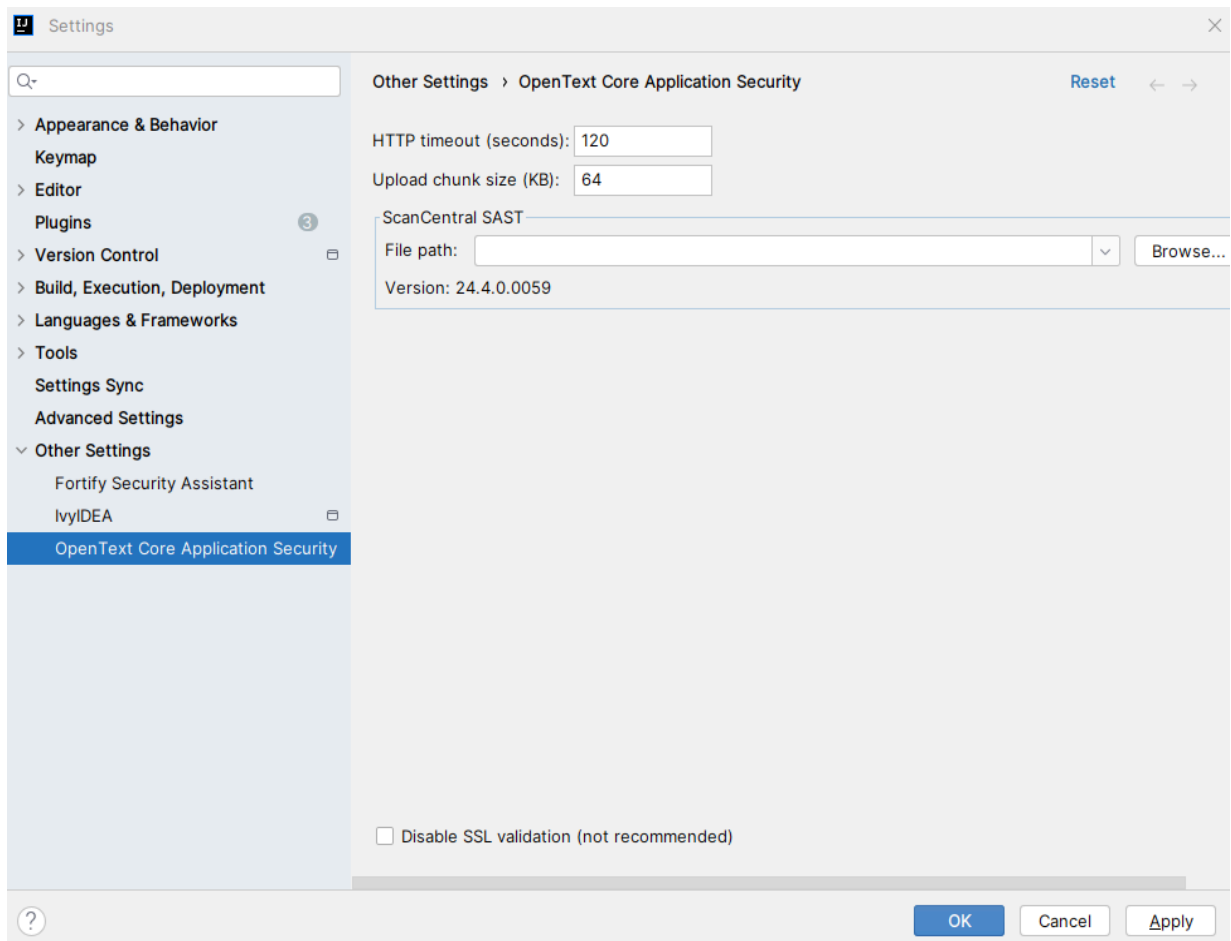
The **Tools** menu now includes the OpenText Core Application Security commands.

1.1.4. Configuring the Core Application Security Plugin for IntelliJ IDEA

You can configure options for uploading source code and for communicating with OpenText Core Application Security at any time.

To configure the IntelliJ plugin options:

1. From the IDE, select **Tools > OpenText Core Application Security > Options**.



2. To change the amount of time to wait when communicating with OpenText Core Application Security before giving up, type the time in seconds in the **HTTP timeout** box.

Valid values for HTTP timeout are 0 through 21600 (6 hours). A value of 0 indicates that there is no timeout. The default HTTP timeout is 120 seconds (2 minutes).

3. To change the size of individual pieces that are uploaded to OpenText Core Application Security, type the size in kilobytes in the **Upload chunk size** box.

The packaged source code is uploaded in chunks for optimal performance and reliability. The valid chunk size values are 1 through 10000 KB. To upload a large file with a stable connection, you can specify larger chunks sizes. To upload a small file with a connection that is not as reliable (for example, a wireless connection), use the default chunk size of 64 KB or smaller.

4. If you are using the Fortify ScanCentral SAST client to package your code, under **ScanCentral SAST** click **Browse** to the right of **File path** to specify the location of the executable.

You can download the Fortify ScanCentral SAST client from the Tools page. For installation instructions, see the README.txt file included in the downloaded ZIP.

After you specify the path to the Fortify ScanCentral SAST client, its version is displayed.



Note

If you do not specify the location of the Fortify ScanCentral client in the configuration options, then the first time you upload your project to OpenText Core Application Security using ScanCentral packaging, you are prompted to specify the location. <<This works in the VS FoD extension, but does not work in the IntelliJ plugin>>

5. To enable the plugin to accept any server certificate, select the **Disable SSL validation** check box.



Note

For security reasons, selecting this option is not recommended.

6. Click **OK**.

1.2. Uploading static code to OpenText Core Application Security

Before you start, make sure you have the following:

- Your OpenText Core Application Security login credentials



Note

To upload your code, you must have the Start Static Scan permission.

- A project open in IntelliJ IDEA or Android Studio
- To include a Debricked open source scan for this solution and automatically package the project with all the required files, verify that you have the following:
 - Fortify ScanCentral SAST client version 22.1.2 or later
 - Core Application Security Plugin for IntelliJ IDEA version 23.1 or later



Note

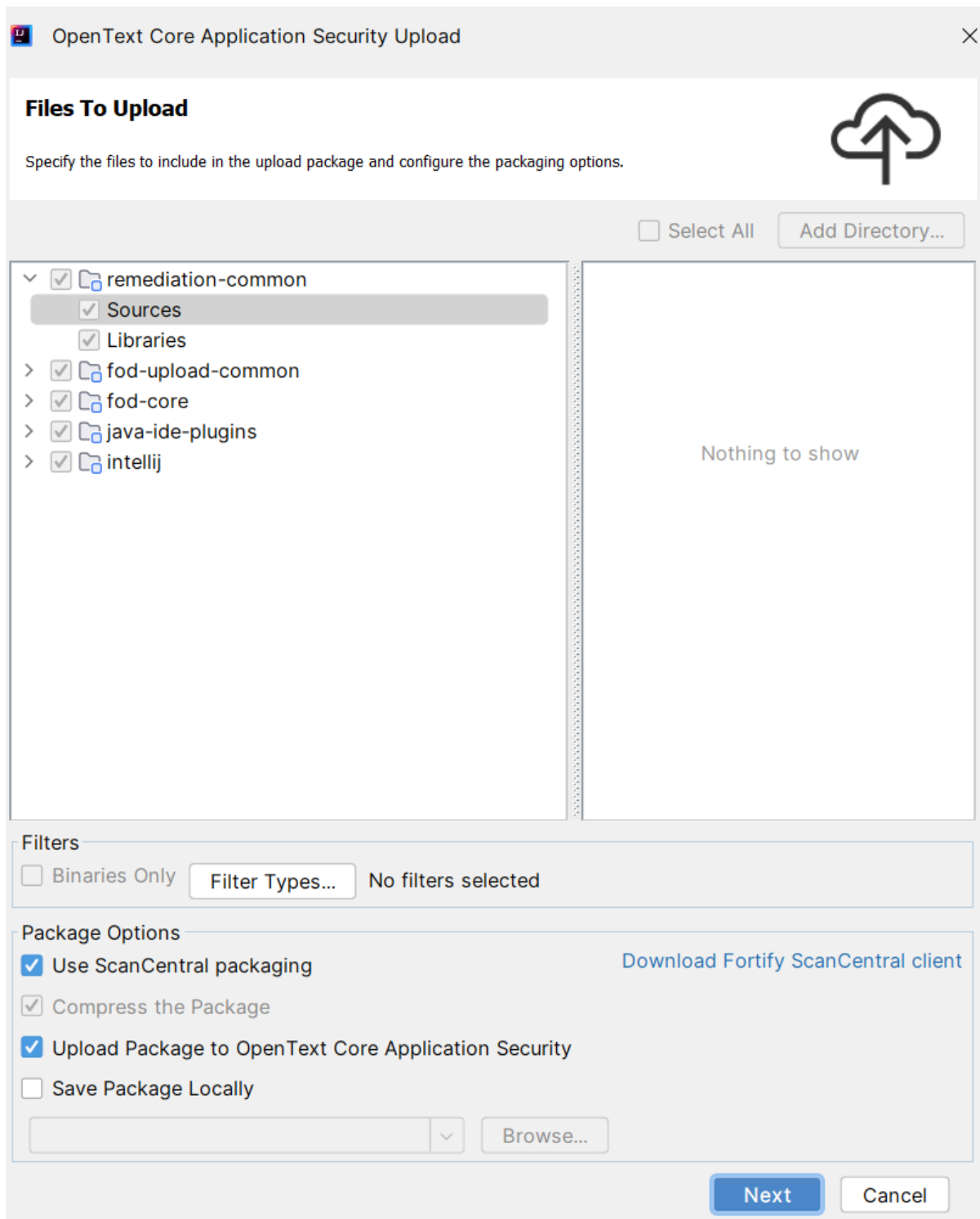
To include a Debricked open source scan for the project without using the Fortify ScanCentral SAST client, make sure your project includes the file required to detect dependencies as described in the *OpenText™ Core Application Security User Guide* before you upload the code to OpenText Core Application Security. You must manually verify that all the files to package are selected including the required file prepared for open source scanning.

To upload source code to Core Application Security Plugin for IntelliJ IDEA:

1. From the IDE, select **Tools > OpenText Core Application Security > Start Scan > Static**.

The OpenText Core Application Security Upload wizard opens.

Files to Upload



The left pane displays all the open modules.

2. To automatically package the project with Fortify ScanCentral SAST client, select **Use ScanCentral packaging**.

Fortify ScanCentral client can automatically package all the necessary dependencies and source code required for the scan. To use this feature, you must have a locally installed Fortify ScanCentral SAST client and specify the installation location in the plugin configuration (see [Configuring the Core Application Security Plugin for IntelliJ IDEA](#)).

To download the Fortify ScanCentral SAST client:

1. Click **Download Fortify ScanCentral client**.
2. Log in to OpenText Core Application Security and download the ScanCentral Client utility from the Tools page.

For instructions on how to install the Fortify ScanCentral SAST client, see the `README.txt` file included in the downloaded ZIP.

3. To manually select all the files you want to upload (without using the Fortify ScanCentral SAST client):

1. Clear the **Use ScanCentral packaging** check box.
2. Select the files you want to upload.

To refine the files to upload, perform one or more of the procedures described in the following table.

File Upload Refinement	Procedure
Omit specific modules.	<ul style="list-style-type: none"> ◦ In the left pane, clear the check box for the module you want to exclude from your upload package.
Omit specific files located in a module listed in the left pane.	<ol style="list-style-type: none"> 1. Select a folder in the left pane. 2. In the right pane, clear the check boxes for the files you want exclude from your upload package.
Upload only binary files (EAR, EXE, CLASS, DLL, JAR, and WAR files).	<ul style="list-style-type: none"> ◦ Select the Binaries only check box.
Upload only specific file types.	<ol style="list-style-type: none"> 1. Click Filter Types. 2. In the Select Types dialog box, select the check boxes for the file types that you want to upload. 3. Click OK.
Upload one or more additional projects.	<ol style="list-style-type: none"> 1. Click Add Projects. 2. In the Folder Selection dialog box, navigate to and select the additional project to upload and click OK.

File Upload Refinement	Procedure
Upload additional external resources.	<ol style="list-style-type: none"> 1. Click Add Directory. 2. In the Browse for Folder dialog box, navigate to and select the resources to upload, and then click Open. <p>After you add a folder, you can select specific files within it.</p> <p>Notes:</p> <ul style="list-style-type: none"> ■ Subfolders are included. ■ Folders with native Unicode symbols in the name are not supported.

4. To upload the package to OpenText Core Application Security, leave the **Upload package to OpenText Core Application Security** check box selected.

5. To save the package to a local directory:

1. Select the **Save package locally** check box.
2. Click **Browse**, and then navigate to a folder where you want to save the package.
3. Type a name for the package, and then click **Save**.

6. To reduce the package size, select the **Compress the package** check box.

Although this reduces the size of the package to be uploaded, the packaging process takes longer.

7. Do one of the following:

- If are saving the package locally without uploading it to Fortify on Demand, click **Finish**.

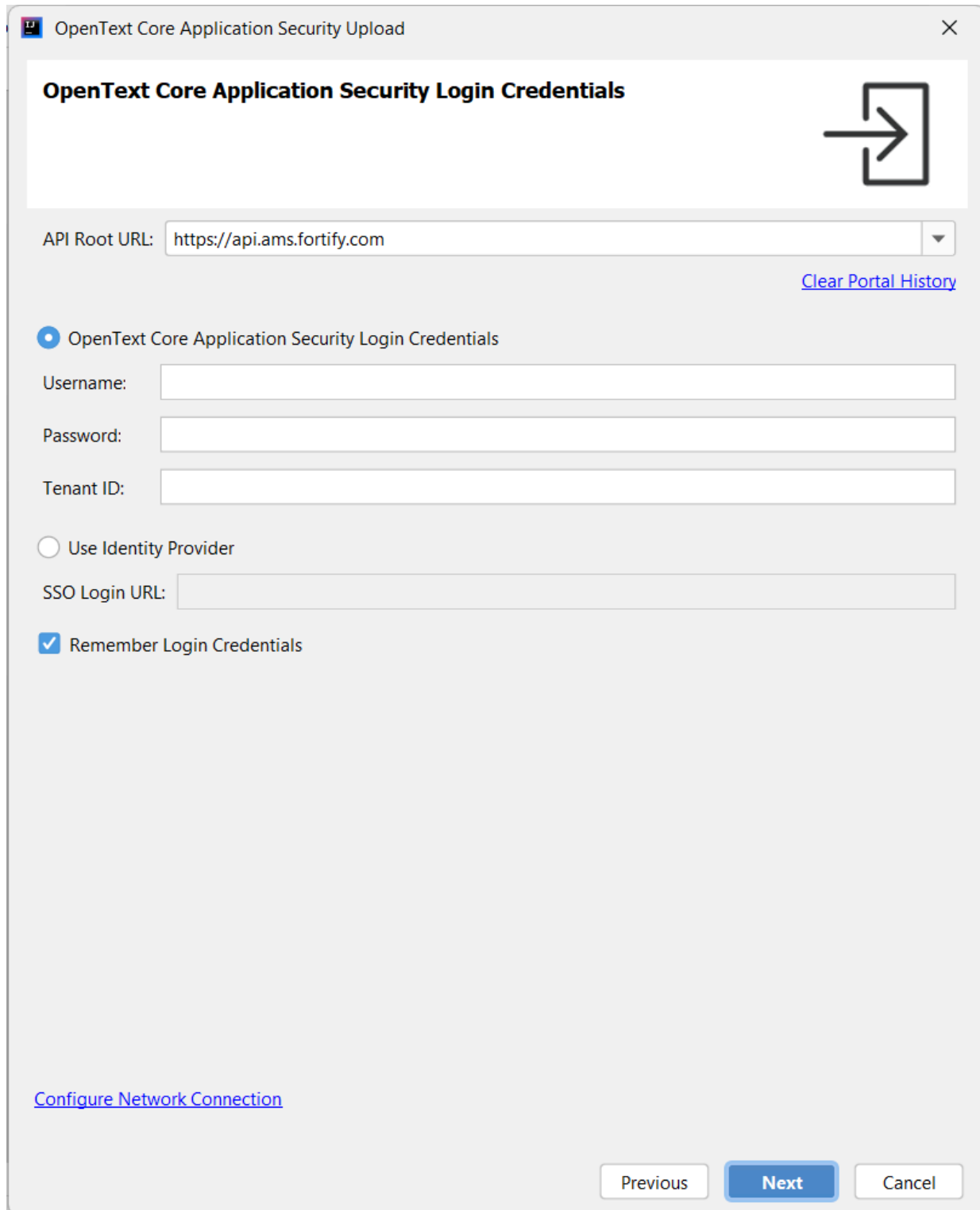
A package (ZIP) is created in the location you specified.

- To upload the package to OpenText Core Application Security, click **Next** to proceed to the **OpenText Core Application Security Login Credentials**

page.

OpenText Core Application Security Login Credentials

If you have already logged in to OpenText Core Application Security, then the next step is to select an application and release (see [Release Selection and Static Scan Setup](#)).



OpenText Core Application Security Upload

OpenText Core Application Security Login Credentials

API Root URL: [Clear Portal History](#)

☒ OpenText Core Application Security Login Credentials

Username:

Password:

Tenant ID:

☐ Use Identity Provider

SSO Login URL:


☒ Remember Login Credentials

[Configure Network Connection](#)

Previous Next Cancel

1. In the **API Root URL** box, type the API root URL.

2. Provide your login credentials. Use one of the two methods described in the following table.

Login Method	Procedure
Provide your OpenText Core Application Security credentials.	<ol style="list-style-type: none"> 1. Provide your Username, Password, and Tenant ID. 2. To save your credentials, select the Remember Login Credentials check box. <div>  <p>Note</p> <p>For security reasons, the plugin does not save your password.</p> </div> <ol style="list-style-type: none"> 3. To configure your proxy network preferences, click Configure Network Connection, and then follow the instructions provided by the IDE help.
Use Single Sign-On (SSO) for OpenText Core Application Security if your organization has configured SSO for its tenant.	<ol style="list-style-type: none"> 1. Select Use Identity Provider. 2. In the SSO Login URL box, type the URL provided by your Security Lead.

3. Click **Next**.

4. If your tenant requires two-factor authentication, then do the following:

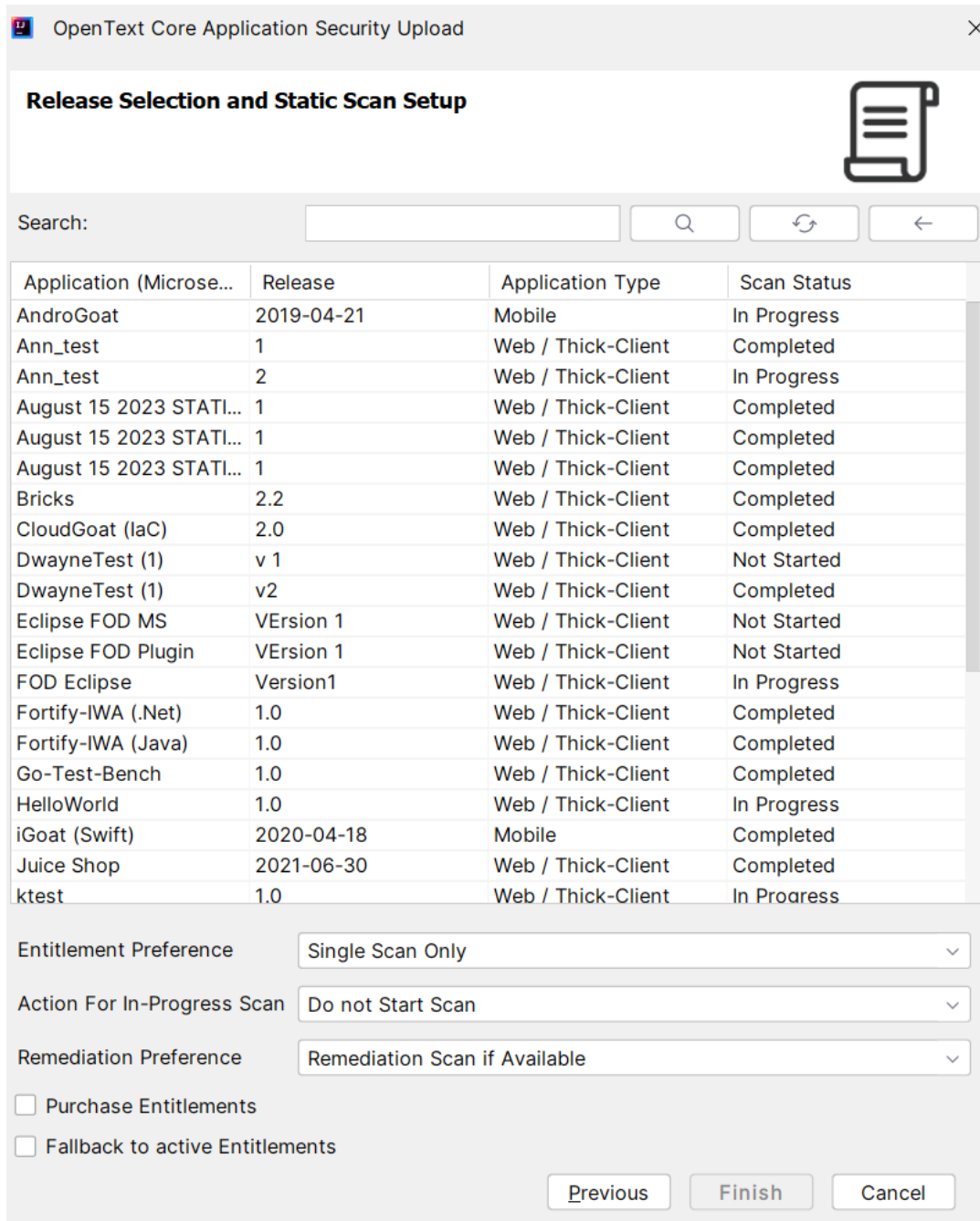
1. In the Two-Step Verification dialog box, select a delivery method for the security code (**SMS** or **Email**), and click **OK**.
2. Obtain the security code delivered using the method you selected.

3. Enter the code in the **Security Code** box, and then click **OK**.

The Core Application Security Plugin for IntelliJ IDEA allows you three attempts to enter the security code. If necessary, click **Resend Code** to have a new security code sent to you.

Release Selection and Static Scan Setup

1. Select the application and release for your upload package.



The screenshot shows the 'Release Selection and Static Scan Setup' dialog box. It features a search bar at the top with a magnifying glass icon and a refresh button. Below the search bar is a table with four columns: Application (Microse...), Release, Application Type, and Scan Status. The table lists various applications and their scan statuses. Below the table are three dropdown menus for 'Entitlement Preference' (Single Scan Only), 'Action For In-Progress Scan' (Do not Start Scan), and 'Remediation Preference' (Remediation Scan if Available). At the bottom, there are two checkboxes: 'Purchase Entitlements' and 'Fallback to active Entitlements'. The dialog box has 'Previous', 'Finish', and 'Cancel' buttons at the bottom right.

Application (Microse...)	Release	Application Type	Scan Status
AndroGoat	2019-04-21	Mobile	In Progress
Ann_test	1	Web / Thick-Client	Completed
Ann_test	2	Web / Thick-Client	In Progress
August 15 2023 STATL...	1	Web / Thick-Client	Completed
August 15 2023 STATL...	1	Web / Thick-Client	Completed
August 15 2023 STATL...	1	Web / Thick-Client	Completed
Bricks	2.2	Web / Thick-Client	Completed
CloudGoat (IaC)	2.0	Web / Thick-Client	Completed
DwayneTest (1)	v 1	Web / Thick-Client	Not Started
DwayneTest (1)	v2	Web / Thick-Client	Completed
Eclipse FOD MS	Version 1	Web / Thick-Client	Not Started
Eclipse FOD Plugin	Version 1	Web / Thick-Client	Not Started
FOD Eclipse	Version1	Web / Thick-Client	In Progress
Fortify-IWA (.Net)	1.0	Web / Thick-Client	Completed
Fortify-IWA (Java)	1.0	Web / Thick-Client	Completed
Go-Test-Bench	1.0	Web / Thick-Client	Completed
HelloWorld	1.0	Web / Thick-Client	In Progress
iGoat (Swift)	2020-04-18	Mobile	Completed
Juice Shop	2021-06-30	Web / Thick-Client	Completed
ktest	1.0	Web / Thick-Client	In Proaress

Entitlement Preference: Single Scan Only

Action For In-Progress Scan: Do not Start Scan

Remediation Preference: Remediation Scan if Available

☐ Purchase Entitlements

☐ Fallback to active Entitlements

Previous Finish Cancel

If you have permission, you can create a new release for an existing application.

To quickly find an application and release, type the name or partial name of an application or release in the **Search** box, and then press **Enter**. The search is

case-insensitive. To clear the search results, clear the **Search** box, and then press **Enter**.



Note

To refresh the list of applications, click **Refresh** .

To create a new release for an existing application:

1. Click **Create New Release** .

The Create a Release dialog box opens.

2. Select an application from the **Application** list.

3. In the **Release Name** box, type a unique name for the release.

Valid characters include letters, digits, underscores, and spaces.

4. (Optional) Type a description for the new release.

5. From the **SDLS Status** list, select the Software Development Lifecycle.

6. Select **Copy State from Existing Release** to copy vulnerabilities and other details from a previous release to the new one, and then select the release that you want to copy from the list.

7. Click **OK**.

2. Select an **Entitlement Preference** from the list.

If multiple entitlements are available, the scan will use the oldest entitlement. If the release has an active subscription, the scan will use the active subscription.

3. From the **Action for In-Progress Scan** list, select what should happen if the selected release scan status is **In Progress**.

You can choose to not start the scan, cancel the in-progress scan and start the new scan, or queue the new scan.

4. From the **Remediation Preference** list, select whether to run a remediation scan.

5. To purchase entitlements for this scan (if available), select the **Purchase Entitlements** check box.

6. To fall back to the next active entitlement once the current entitlement subscription ends, select **Fallback to active Entitlements** check box.

7. Select an assessment type from the **Assessment Type** list.



Note

Steps 7 through 10 (shown under **Static Scan Details**) are only applicable if the selected release does not have scan settings configured yet. The fields described in these steps are hidden if the scan settings are already configured.

Static Scan Details

10917 Unit(s) Available

Assessment Type: microservice - Subscription (2 Units)

Technology Stack: JAVA/J2EE/Kotlin

Language Level: 17

☐ Open Source Component Analysis

Audit Preference: Automated

☐ Fortify Aviator

<https://aws.amazon.com/marketplace/pp/prodview-3b3i27cz6kzw2>

☐ Include third-party libraries for static security assessment (will lead to longer turnaround times)

To upload your project, you must select an assessment type that is less than or equal to the entitlements that you have available.

8. Specify the **Technology Stack** and the **Language Level**.


These fields display the values automatically detected for the selected IntelliJ IDEA project. The **Auto detect** option can identify Java/J2EE/Kotlin, C/C++, PYTHON, Ruby, and CFML projects. If you have previously used the IntelliJ IDEA plugin, the technology stack and language level that you used previously are restored.

9. If you want open source libraries identified in the analysis (and you have entitlements for this feature), select the **Open Source Component Analysis** check box.

The open source scan results include identified open source components and associated security issues.

This

10. To specify additional advanced scanning and auditing preferences, make selections for the options described in the following table.

Option	Description
Audit Preference	<p>This option is only available if enabled for your tenant.</p> <p>The audit preference settings are:</p> <ul style="list-style-type: none"> ◦ Manual—A security expert manually reviews the scan results and removes false positives. ◦ Automated—False positives identified by Fortify Audit Assistant with high confidence are automatically suppressed and results are published without manual review. This can reduce the turnaround time. <div>  <p>Note</p> <p>Fortify Aviator is only applicable for Automated audit.</p> </div>
Fortify Aviator	<p>For scans using Automated audit, select the check box to have SAST Aviator audit results and provide enhanced remediation assistance.</p>

Option	Description
Include third-party libraries for static security assessment	<p>Authorizes OpenText Core Application Security to assess the code for vulnerabilities to include in reports, vulnerability count, and risk rating.</p> <p>Selecting this option indicates that your organization has received consent from all third-party vendors to scan their libraries.</p>

11. Click **Finish** to upload your code to OpenText Core Application Security.

Information about the IDE (name and version) used for this upload is saved and shown in the scan summary.

1.3. Uploading open source code to OpenText Core Application Security

Before you start, make sure you have the following:

- Your OpenText Core Application Security login credentials



Note


To upload your code, you must have the Start Static Scan permission.

- A project open in IntelliJ IDEA or Android Studio
- To upload a Debricked open source scan for this solution and automatically package the project with all the required files, verify that you have the following:
 - Fortify ScanCentral SAST client version 22.1.2 or later
 - Core Application Security Plugin for IntelliJ IDEA version 23.1 or later
 - Ensure that the **Run Debricked Open Source Scan** option is selected for the application in Core Application Security.


1. From the IDE, select **Tools > OpenText Core Application Security > Start Scan > Open Source**.

OpenText Core Application Security Login Credentials

If you have already logged in to OpenText Core Application Security, then the next step is to select an application and release (see [Release Selection and Static Scan Setup](#)).


Start Open Source Scan

OpenText Core Application Security Login Credentials



API Root URL:

▼

[Clear Portal History](#)

☒ OpenText Core Application Security Login Credentials

Username:

Password:

Tenant ID:

☐ Use Identity Provider

SSO Login URL:

☒ Remember Login Credentials


[Configure Network Connection](#)

Previous

Next

Cancel

1. In the **API Root URL** box, type the API root URL.
2. Provide your login credentials. Use one of the two methods described in the following table.

Login Method	Procedure
Provide your OpenText Core Application Security credentials.	<ol style="list-style-type: none"> 1. Provide your Username, Password, and Tenant ID. 2. To save your credentials, select the Remember Login Credentials check box. <div>  <p>Note</p> <p>For security reasons, the plugin does not save your password.</p> </div> <ol style="list-style-type: none"> 3. To configure your proxy network preferences, click Configure Network Connection, and then follow the instructions provided by the IDE help.
Use Single Sign-On (SSO) for OpenText Core Application Security if your organization has configured SSO for its tenant.	<ol style="list-style-type: none"> 1. Select Use Identity Provider. 2. In the SSO Login URL box, type the URL provided by your Security Lead.

3. Click **Next**.

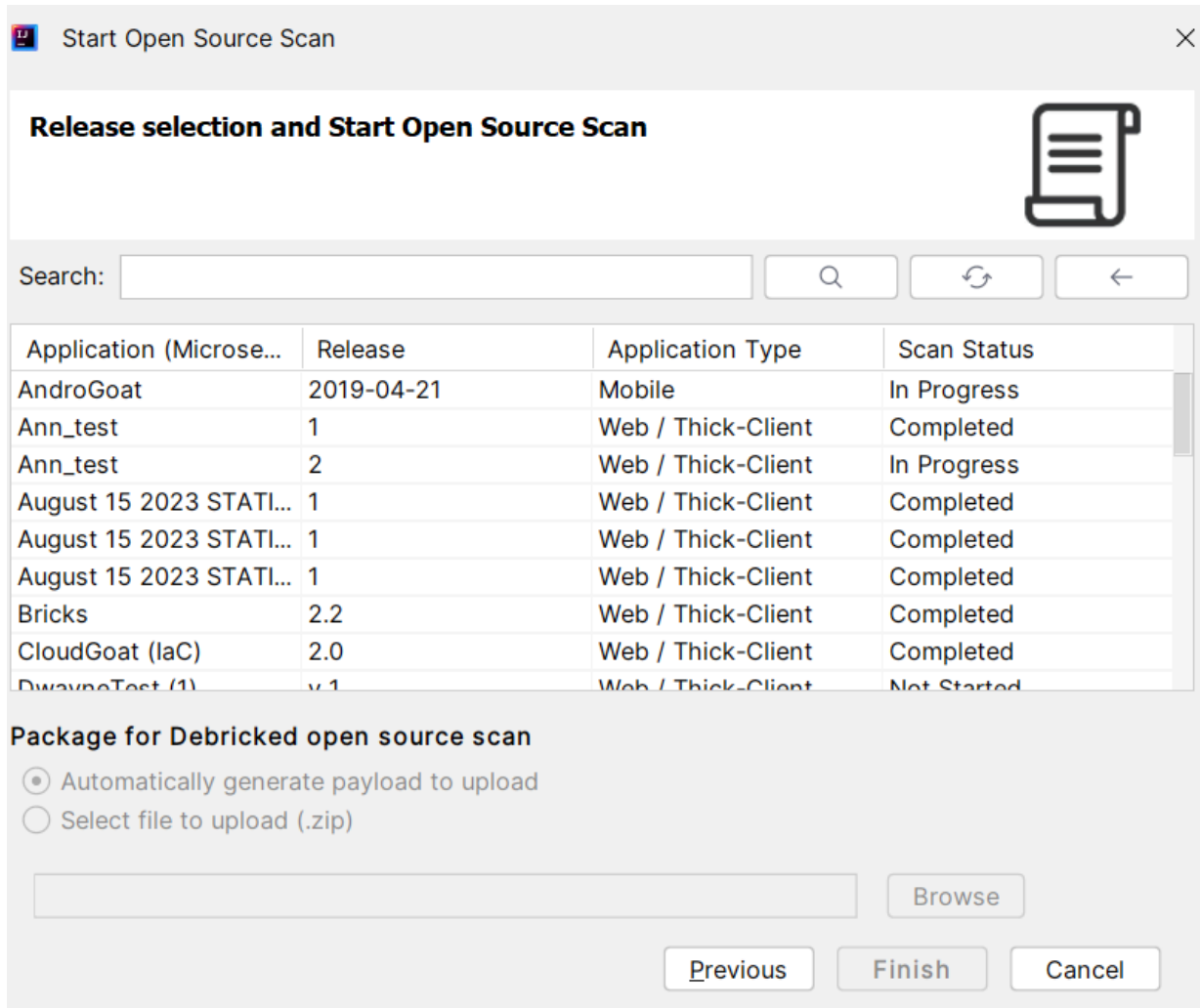
4. If your tenant requires two-factor authentication, then do the following:

1. In the Two-Step Verification dialog box, select a delivery method for the security code (**SMS** or **Email**), and click **OK**.
2. Obtain the security code delivered using the method you selected.
3. Enter the code in the **Security Code** box, and then click **OK**.

The Core Application Security Plugin for IntelliJ IDEA allows you three attempts to enter the security code. If necessary, click **Resend Code** to have a new security code sent to you.

Release selection and Start Open Source Scan

1. Select the application and release for your upload package.



Application (Microse...	Release	Application Type	Scan Status
AndroGoat	2019-04-21	Mobile	In Progress
Ann_test	1	Web / Thick-Client	Completed
Ann_test	2	Web / Thick-Client	In Progress
August 15 2023 STATI...	1	Web / Thick-Client	Completed
August 15 2023 STATI...	1	Web / Thick-Client	Completed
August 15 2023 STATI...	1	Web / Thick-Client	Completed
Bricks	2.2	Web / Thick-Client	Completed
CloudGoat (IaC)	2.0	Web / Thick-Client	Completed
DwayneTest (1)	v.1	Web / Thick-Client	Not Started

Package for Debricked open source scan

☒ Automatically generate payload to upload
☐ Select file to upload (.zip)

2. In the **Package for Debricked open source scan** section, select one of the following options:
 1. If you want automatically package the project into a .zip file, select **Automatically generate payload to upload**.
 2. If you want to upload an existing package of the project, choose the **Select file to upload (.zip)** option and click **Browse** to upload a .zip file.
3. Click **Finish** to upload your code to OpenText Core Application Security

The **Start Open Source Scan** status bar shows the status for generating and uploading the Fortify ScanCentral SAST package to OpenText Core Application

Security.

After successful upload of your code to OpenText Core Application Security, Core Application Security Plugin for IntelliJ IDEA displays a **Scan ID**.

1.4. Reviewing analysis results

From the IDE, you can open OpenText Core Application Security analysis results for an application and release to remediate and audit.

- [Opening analysis results](#)
- [Analysis results in the Core Application Security Plugin for IntelliJ IDEA](#)
- [Reviewing issues](#)
- [Auditing issues](#)
- [Locating the source code associated with static scan issues](#)

1.4.1. Opening analysis results

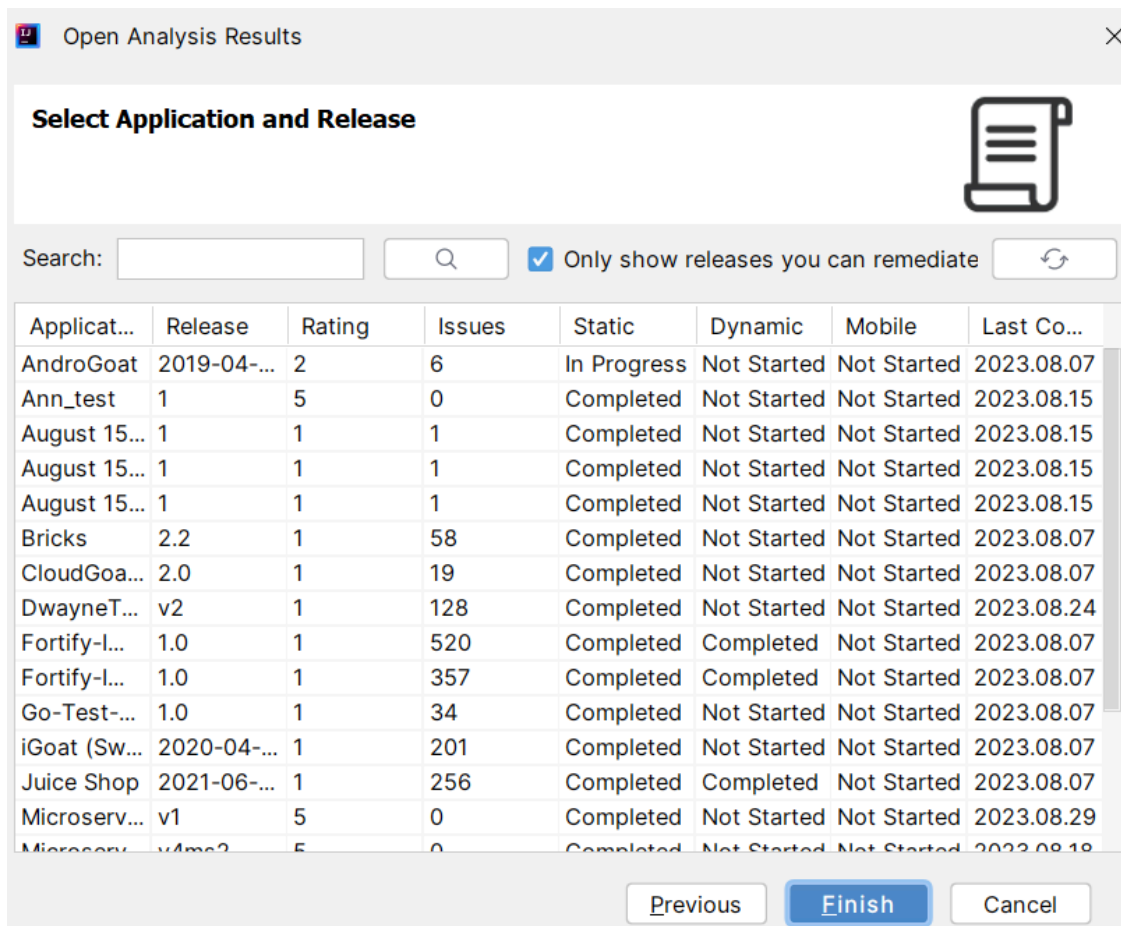
If you already have analysis results open, you can use this procedure to close the current analysis results and open the analysis results for a different application and release.

To open the analysis results from OpenText Core Application Security:

1. From the IDE, select **Tools > OpenText Core Application Security > Open Analysis Results**.

The Fortify on Demand Open Analysis Results wizard opens.

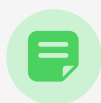
2. Select an application and release for the analysis results you want to open.



You can only open results for an application and release that has had at least one successfully completed scan. Clear the **Only show releases you can remediate** check box to see all applications and releases.

To quickly find an application and release, type the name or partial name of an application or release in the **Search** box, and then press **Enter**. The search is

case-insensitive. To clear the search results, clear the **Search** box, and then press **Enter**.



Note

To refresh the list of applications, click **Refresh** .

3. Click **Finish**.

The analysis results are displayed in the Fortify windows. See [Analysis Results in the Core Application Security Plugin for IntelliJ IDEA](#) for a description of the Fortify windows.

1.4.2. Analysis results in the Core Application Security Plugin for IntelliJ IDEA

After the analysis results are opened, the Core Application Security Plugin for IntelliJ IDEA displays four audit-focused windows. The **Analysis Results** window displays the results. The **Analysis Trace**, **Audit Summary**, and **Issue Summary** windows are visible, but do not contain any information until you select an issue from the **Analysis Results** window.



Note

To open a Core Application Security window that is not currently visible, select **Tools > OpenText Core Application Security > Show View** and select the window you want to open.

The following table describes the Fortify windows.







View	Description
Analysis Results	<p>The Analysis Results window provides a way to group and select the issues to audit. This view also displays the relevant trace output for issues (see the following description of the Analysis Trace window).</p> <p>The color-coded tabs in the Analysis Results group the issues by severity level. The last tab contains all issues. The Group By list options sort the issues into subfolders. The option you select is applied to all visible folders.</p>
Analysis Trace	<p>For Static Analysis Results—After you select an issue in the Analysis Results window, the Analysis Trace window displays the relevant trace output. This is a set of program points that show how the analyzer found the issue. For dataflow and control flow issues, the set is presented in the order executed. For dataflow issues, this evidence is the path that the tainted data follows from the source function to the sink function. See the following descriptions of the analysis trace icons.</p> <p>For Dynamic Analysis Results—After you select an issue in the Analysis Results window, the Analysis Trace window displays details about the request parameters.</p> <p>This window also provides an abstract that briefly describes the issue.</p>



View	Description
Audit Summary	After you select an issue in the Analysis Results view, the Audit Summary view displays audit information for the selected issue. You can edit issue information, add comments, and review the audit history. For more information, see Auditing Issues .
Issue Summary	After you select an issue in the Analysis Results window, the Issue Summary window provides detailed information about the issue. The Details tab provides an abstract of the issue, a detailed explanation, and might also include examples with descriptive text and code samples, and the scan type (static, dynamic, or mobile). The Recommendations tab displays recommendations to remediate the issue, along with tips and references for additional research.









The Editor is where the IntelliJ plugin displays the source code (if available) for static scans or the request and response details for dynamic scans. The Editor opens after you select an issue in the **Analysis Results** window.





Analysis Trace Icons

The analysis trace icons described in the following table show how dataflow moves in the section of the source code or execution order.

Icon	Description	
	Data is assigned to a field or variable	
	Information is read from a source external to the code such as an HTML form or a URL	
	Data is assigned to a globally scoped field or variable	
	A comparison is made	
	The function call receives tainted data	
	The function call returns tainted data	

Icon	Description
	<p>Passthrough, tainted data passes from one place to another</p> <p>This is typically shown as <code>functionA(x : y)</code> to indicate that data is transferred from x to y. The x and y values are one of the following:</p> <ul style="list-style-type: none"> • An argument index • <code>return</code> —The return value of a function • <code>this</code> —The instance of the current object • A specific object field or key
	<p>An alias is created for a memory location</p>

Icon	Description	
	Data is read from a variable	
	Data is read from a global variable	
	Tainted data is returned from a function	
	A pointer is created	
	A pointer is dereferenced	
	The scope of a variable ends	
	The execution jumps	
	A branch is taken in the code execution	

Icon	Description	
	A branch is not taken in the code execution	
	Generic	
	A runtime source, sink, or validation step	
	Taint change	

1.4.3. Reviewing issues

To view and select issues:

1. From the **Group By** list, select a value to use to sort issues in all visible folders into groups.

The default grouping is **category**.

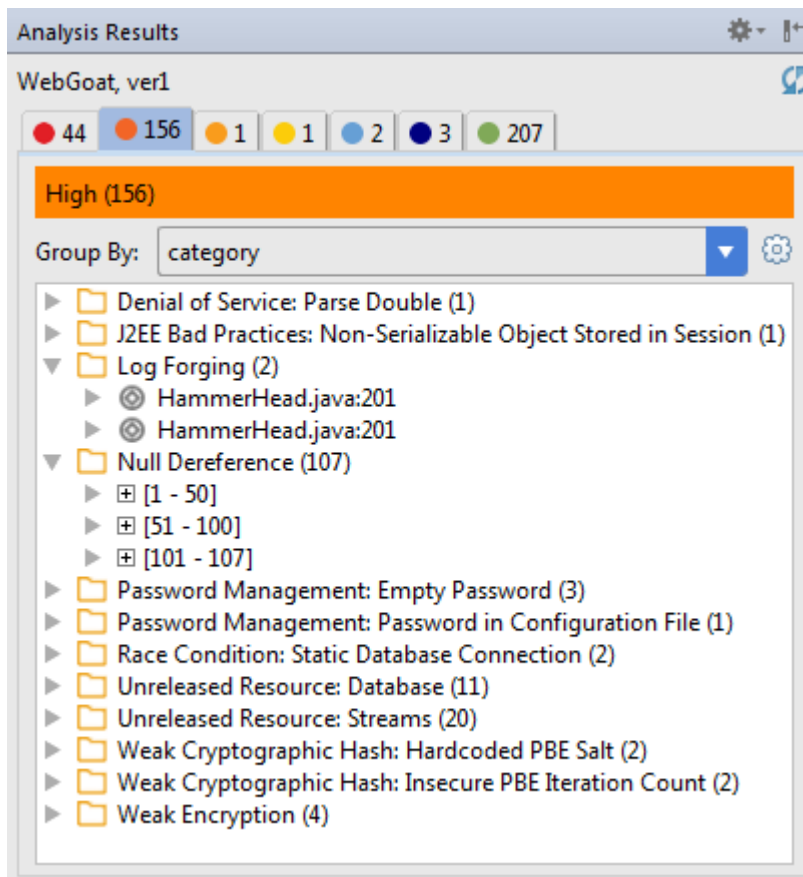
2. Click a colored tab to view the associated issues.

The issue type subfolders listed are based on the selected **Group By** value.

3. To show suppressed or fixed issues, click **Set visibility options** .


4. To view the list of issues in a subfolder, expand the subfolder.

The Core Application Security Plugin for IntelliJ IDEA retrieves the corresponding issues from OpenText Core Application Security.



Note

If a folder contains more than 50 issues, the issues are grouped into subfolders in blocks of 50 with folder names that indicate which issues are included. For example, if a folder contains 71 issues, the first 50 issues are in a subfolder labeled **[1-50]** and the next set of issues are in a subfolder labeled **[51-71]**.

5. To see any updates to the analysis results made on OpenText Core Application Security, click **Refresh** .
6. Select an issue.

The **Analysis Trace**, **Audit Summary**, and **Issue Summary** windows display information about the selected issue.

1.4.4. Auditing issues

If you have the Edit Issues permission, you can assign a user, set the developer status, and add comments for issues in the **Audit Summary** view. If you have the Audit Issues permission, you can also edit the issue's auditor status and severity.

To audit an issue:

1. Make sure that the **Audit Summary** view is open.
2. From the issues list in the **Analysis Results**, select an issue.

You can select multiple issues in the **Analysis Results** view to make the same edits to multiple issues.

3. In the **Audit Summary** view, select the user to assign the issue from **User** list.
4. To change the issue's development status, select the status from the **DeveloperStatus** list.
5. To change the auditor status, select the status from the **AuditorStatus** list.
6. To change the issue severity, select an issue severity from the **Severity** list.
7. To add a comment for the issue, type your comment in the box at the bottom of the **Comments** area, and then click **Add Comment**.

Your comment is displayed in the **Comments** section.

The Core Application Security Plugin for IntelliJ IDEA saves your changes for the OpenText Core Application Security application and release.

1.4.5. Locating the source code associated with static scan issues

You can use the Core Application Security Plugin for IntelliJ IDEA to locate security-related issues in your code.

To jump to the line of source code that contains the issue selected in the plugin:

- Select an issue in the **Analysis Results** window or select a line in the **Analysis Trace** window.

If the issue is located in source code available in the current project, Core Application Security Plugin for IntelliJ IDEA opens the relevant file in the Editor. Otherwise, the plugin attempts to download the source code from OpenText Core Application Security and if the relevant file is available, it is opened in the Editor.



Note

If the file was downloaded, the file name is prepended with Remote<yyyy-mm-dd>- where the <yyyy-mm-dd> is the date the file was last scanned.

The Core Application Security Plugin for IntelliJ IDEA highlights the line of code associated with the issue.

1.5. Auto-remediation with SAST Aviator

You can use OpenText™ SAST Aviator to automatically remediate your code from IntelliJ IDEA using AI-powered security fixes for vulnerabilities detected during a static scan.

SAST Aviator auto-remediation identifies vulnerabilities for static scans configured with SAST Aviator enabled auditing and automatically applies the remediations proposed by SAST Aviator to the source code with a single click. This enables users to fix their source code with minimal manual intervention. The SAST Aviator service leverages Generative Artificial Intelligence (GAI) with a Large Language Model (LLM) to audit results and provide auto-remediation guidance.



Caution

AI-generated remediation suggestions may contain inaccuracies or errors. OpenText recommends you to review all suggestions carefully before applying any fixes.

Requirements

To enable auto-remediation for static assessments in Core Application Security:

- SAST Aviator must be enabled for the Core Application Security tenant.
- Audit preference must be set to Automatic in the static scan settings.
- Fortify Aviator option must be enabled in the static scan settings.

Identifying and selecting vulnerabilities to auto-remediate

Vulnerabilities with SAST Aviator auto-remediation capability are indicated by the Aviator () indicators in the **Analysis Results** view.

To select a vulnerability to auto-remediate:

1. From the IDE, select **Tools > OpenText Core Application Security > Open Analysis Results**.

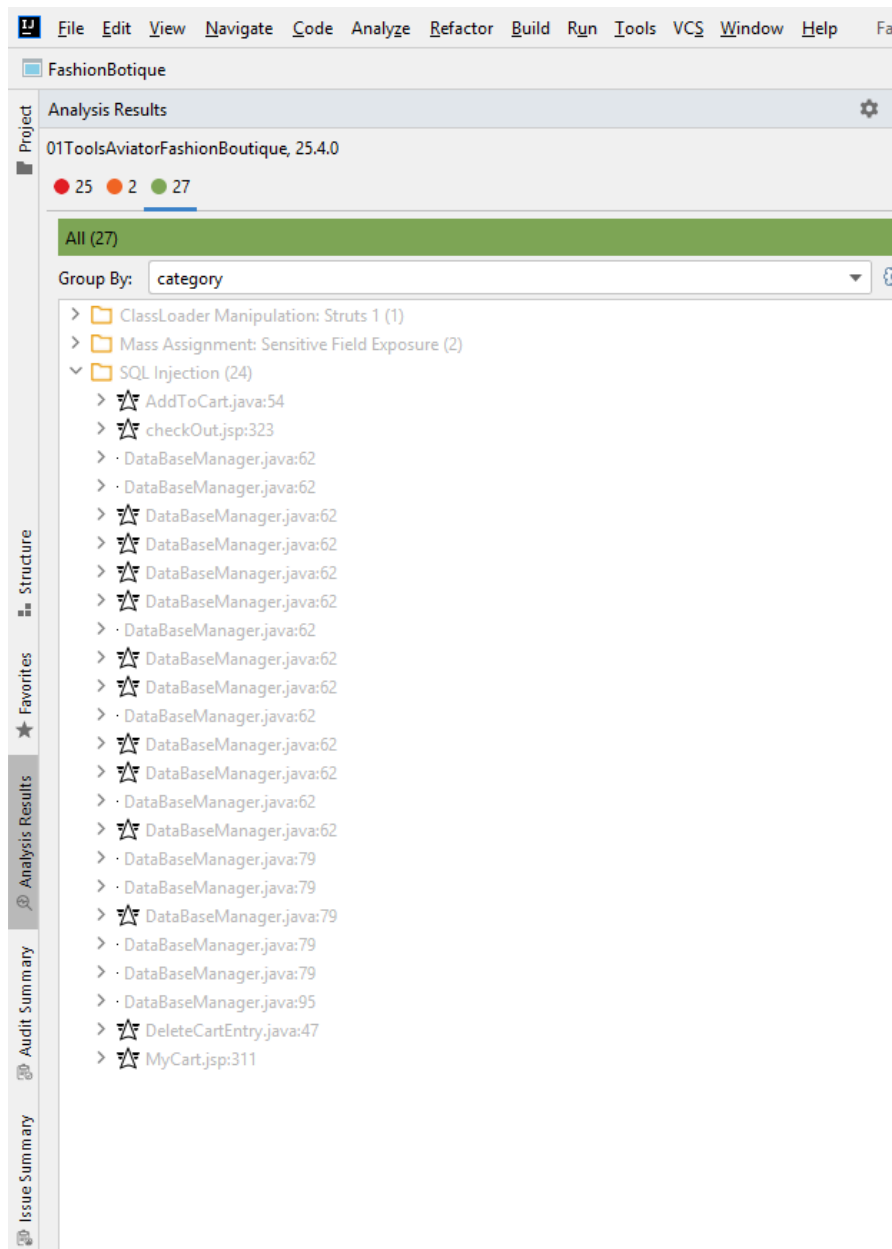
The Open Analysis Results wizard opens.

2. Select an application and release for the analysis results you want to open and that has had at least one successfully completed scan.

3. Click **Finish**.

The Analysis Results view is displayed.

The following example image shows the Analysis Results view with vulnerabilities indicated with the Aviator (✈) indicators:



4. Click on any vulnerability indicated with the Aviator () indicator.

The plugin automatically opens the file in the IntelliJ editor and the Aviator () indicator is displayed near the specific line of code causing the issue.



Auto-Remediate the vulnerabilities

There are two ways to auto-remediate vulnerabilities in the source code:

- Review the proposed code changes in the Diff Review window and accept the changes.
- Quick remediate to apply the fixes instantly without any reviews.

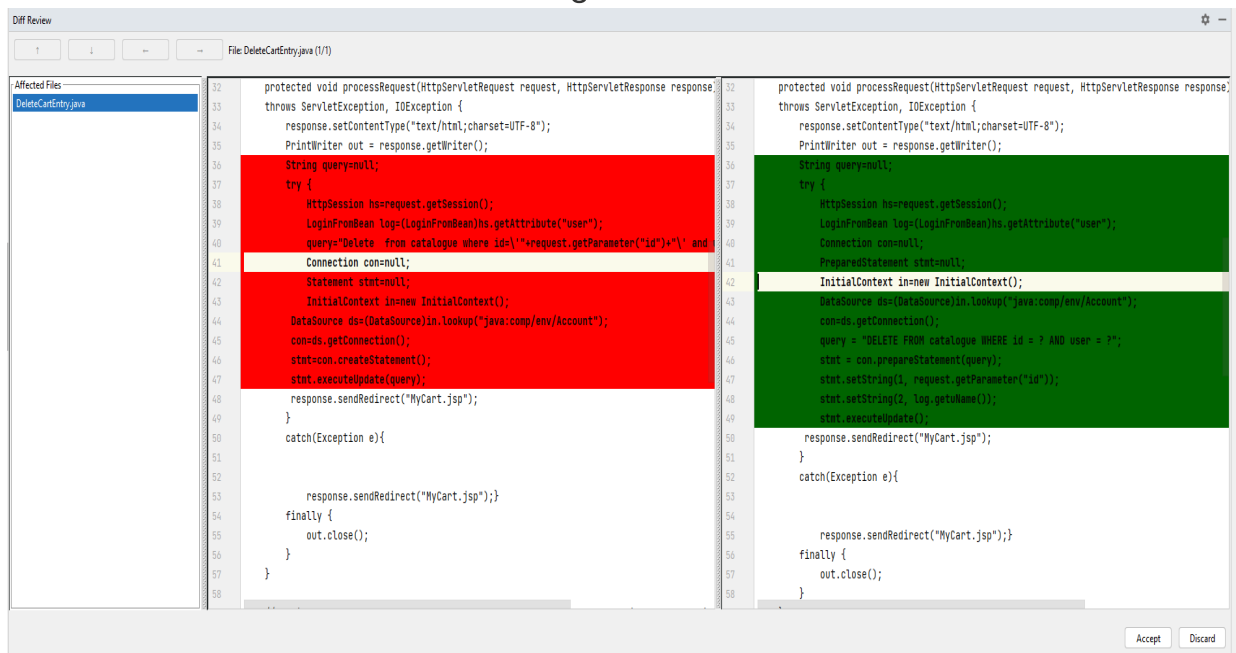
Review and Accept

To review and accept the changes proposed by SAST Aviator:

1. In the IntelliJ editor, navigate to the lines of code indicated by the Aviator () indicator and left-click the **Aviator** () indicator.

The Diff Review window displays the affected files, affected lines of codes, and code fixes.

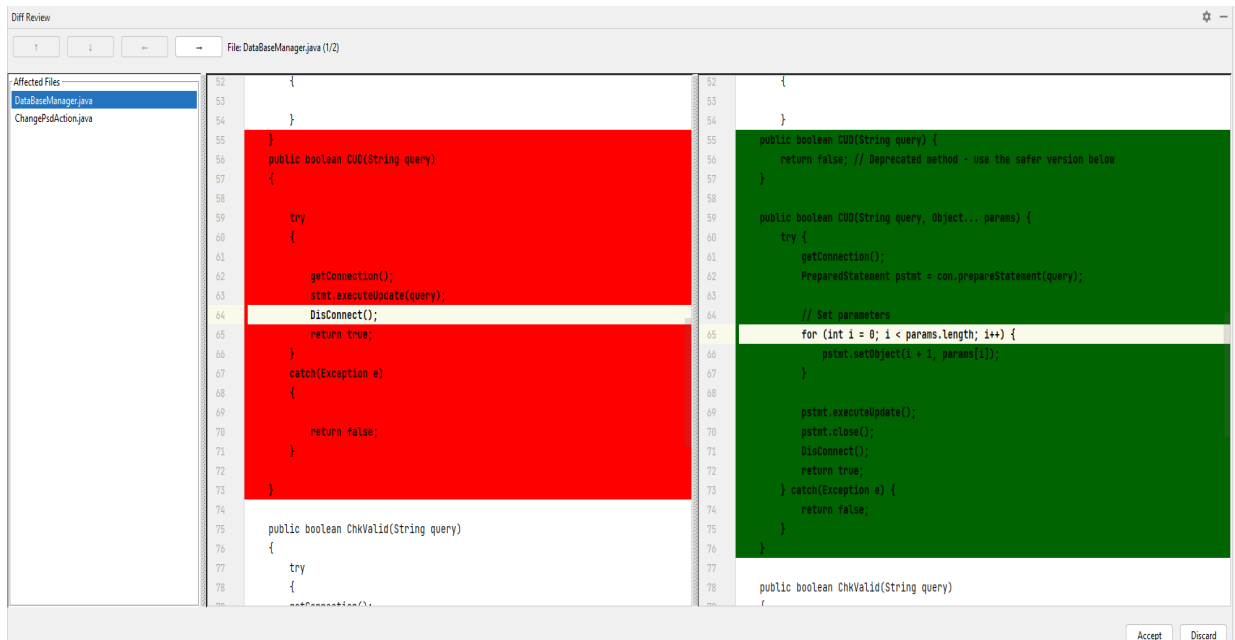
2. The Diff Review displays a split-screen comparison where the left screen indicates the vulnerable code in red and the right screen indicates the proposed fixes for the affected lines of code in green.



Important

Some security vulnerabilities require changes across multiple files to be properly resolved. SAST Aviator applies these fixes and ensures all related changes are applied as a single remediation operation.

If multiple source code files are affected for the vulnerability, the files are listed in the **Affected Files** area. You can use the navigation buttons (← or →) in the **Affected Files** area to review all the vulnerable lines of code and the proposed fixes.



If there multiple block of codes in a single file, use the navigation buttons (↑ and ↓) to navigate between the code blocks.

- Click either **Accept** to apply the fixes or click **Discard** to reject the suggested changes.

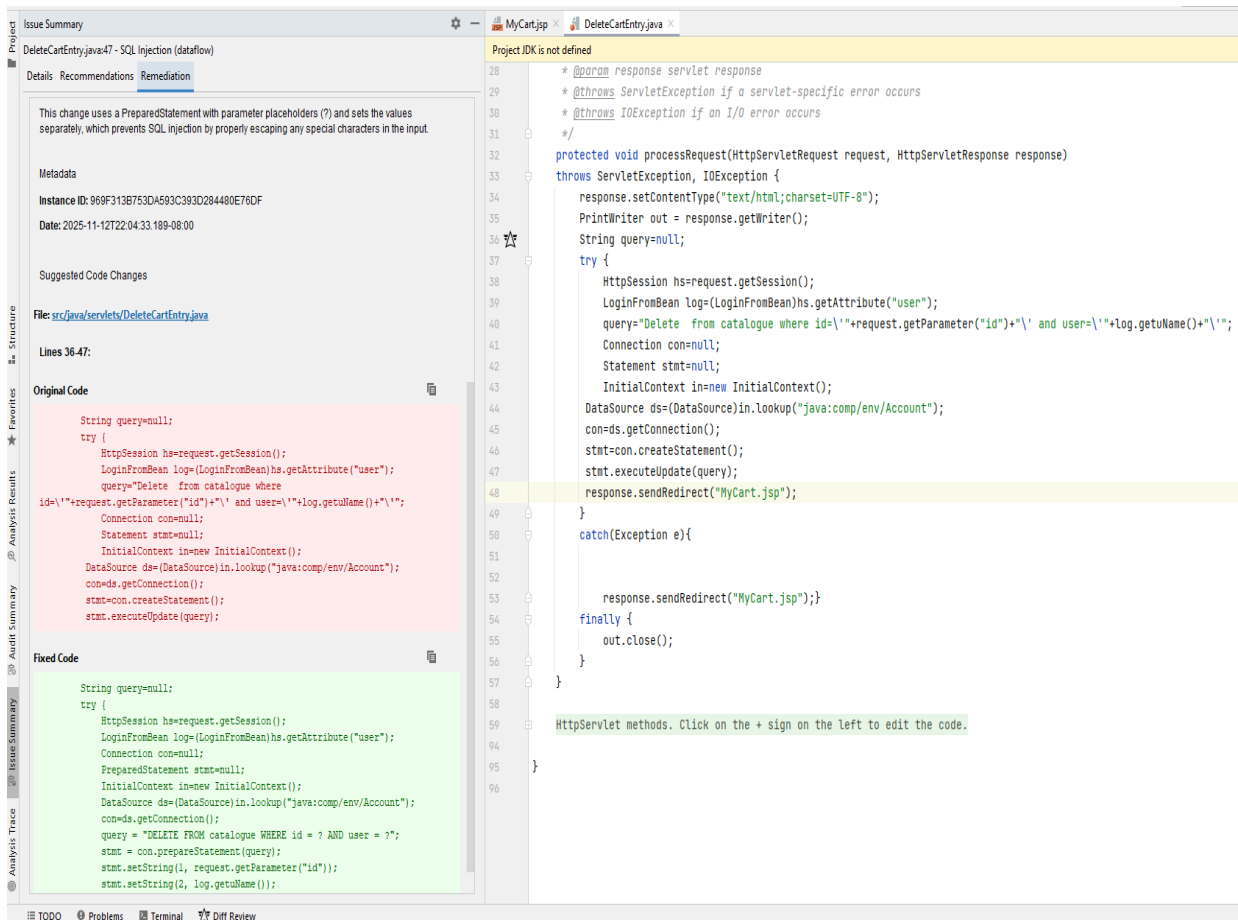
When you click **Accept**, SAST Aviator replaces the vulnerable lines of code with the fixes and the Aviator (✶) indicator disappears after a fix is successfully applied.



Note



If any changes are made to the corresponding affected source code file while the **Diff Review** window is open, the **Accept** button will be disabled. Re-select the vulnerability to view the updated remediation.

- The Core Application Security Plugin for IntelliJ IDEA navigates to the **Remediation** tab on the **Issue Summary** view. You can view the details of the SAST Aviator remediation such as fix description, file name, affected lines of code, original code, and fixed code. You can copy the lines of code in the **Original Code** and **Fixed Code** area to verify the suggested fixes locally.



Quick Remediation

To quick remediate and apply the fixes instantly without any reviews:

1. In the IntelliJ editor, navigate to the lines of code indicated by the Aviator () indicator and right-click the **Aviator** () indicator.



Tip

Click **Show Details** to view the details in the **Remediation** tab on the **Issue Summary** view.

2. Click **Quick Remediate**.

The fixes are applied immediately without any reviews and the Aviator indicator disappears from the lines of code.

3. The Core Application Security Plugin for IntelliJ IDEA navigates to the **Remediation** tab on the **Issue Summary** view. You can view the details of the SAST Aviator remediation such as fix description, file name, affected lines of code, original code, and fixed code.

1.5.1. Best practices for auto-remediation

OpenText recommends the following best practices when you use the SAST Aviator auto-remediation:

- Browse the **Analysis Results** view to understand all the security issues in your project. Focus on SAST Aviator-enabled vulnerabilities, as these vulnerabilities can be fixed quickly.
- Review the **Remediation** tab on the Issue Summary view before applying fixes, especially for complex vulnerabilities, to ensure the suggested changes to your code are viable and necessary for code security.
- For **Quick Remediation**, use **Show Details** to understand the security issue and proposed code changes before applying the fix.



© Copyright 2025 Open Text

For more info, visit <https://docs.microfocus.com>
