Violations Explorer

In this section:

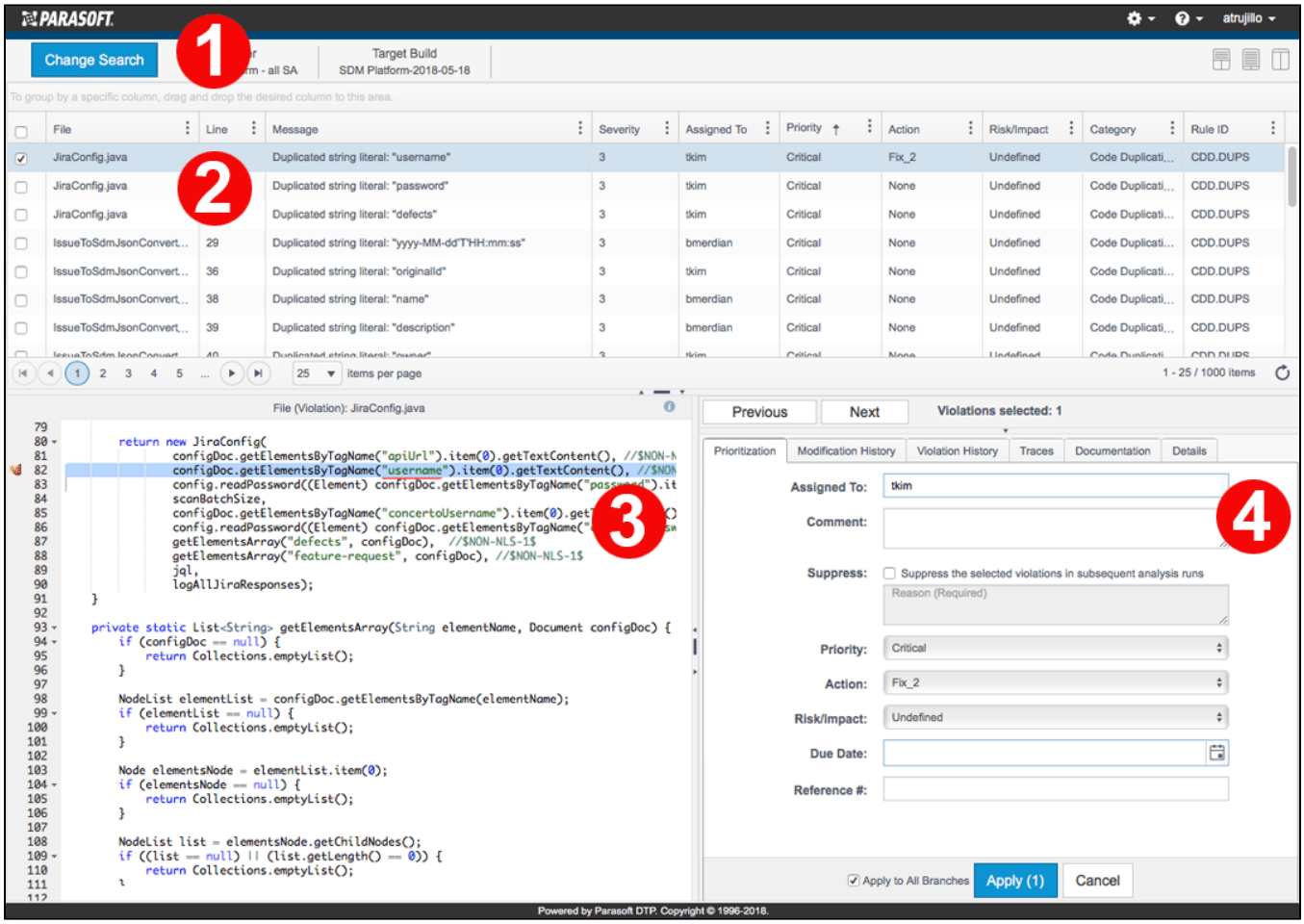
Overview

The Violations Explorer is an interface for systematically reviewing static violations and facilitating remediation. All static analysis violations widgets drill-down to the Violations Explorer.

> **Running static analysis on multiple branches using the same run configuration**
>
> When running static analysis on multiple branches using the same run configuration, the same instance of a violation will be reported across builds as a new violation. As a result, widgets that present changes in the number of violations will not be accurate. You can change this behavior by Configuring Static Analysis Settings.

The Violations Explorer is made up of four main parts:

1. Search panel; See Searching for Violations.
2. Search results panel; See Viewing Search Results.
3. Sources panel; see Viewing Sources.
4. Actions panel; see Addressing Violations, Viewing Flow Analysis Traces, and Viewing Duplicate Code Violations.



Searching for Violations

Violations stored in DTP are searchable by several parameters. Use the search area to hone in on specific types of violations. You can change the criteria in the search area to find violations throughout your development projects.

Click the **Change Search** button to open the search dialog and configure your search criteria. The following search criteria are available:

Filter **Build**	A filter and build ID are the minimum criteria for searching violations. By default, the latest build selected when you change the filter, but you can choose a different build from the drop-down menu. The build selected functions as a target build when a baseline build is selected. See the following sections for additional information: • DTP Concepts • Creating and Managing Filters • Build Administration
Baseline Build **State**	A baseline build is any historical build used for comparison with another build. Choose a baseline build from the drop-down menu to search for violations reported from the baseline build to the build selected with the filter. You can search for new, fixed, or existing violations by choosing a state from the State drop-down menu.
Severity	You can search by one or more severity levels. Severity is determined by the test configuration. You can customize the severity level associated with a rule by creating a rule map. See Configuring Code Analysis Rules for additional information.
Priority	You can search by one or more assigned priorities. Priorities can be customized through the REST API.
Author	You can search by one or more code authors. Authorship is determined from the settings in the code analysis tool.
Assignee	You can search by one or more assignees.
Action	You can search by one or more assigned actions. Actions can be customized through the REST API.
Type	You can search for regular violations, suppressed violations, or all violations.
Resource Groups	A resource group is a collection of resources (i.e., files and/or folders) defined by a set of one or more Ant file patterns. You can search by one or more resource groups. Resource groups can be defined in through the REST API.
Include File Pattern **Exclude File Pattern**	You can specify Ant patterns to narrow or broaden the scope of your search. See Searching for Violations by File for details on configuring file patterns.
Risk /Impact	You can search by one or more risk/impact values. Risk/impact is the extent to which a violation impacts the business. Risk/impact can be customized through the REST API.
Reference Number	You can constrict your search to a specific reference number. Reference numbers can be added manually or automated through the REST API.
Category	You can search by one or more static analysis categories. Static analysis rules are organized into categories, but you can define or remap rules and categories by creating a rule map. See Configuring Code Analysis Rules for additional information.
Module	You can search by one or more specific modules.
Limit	You can set a limit for the number of violations shown in the Violations Explorer.

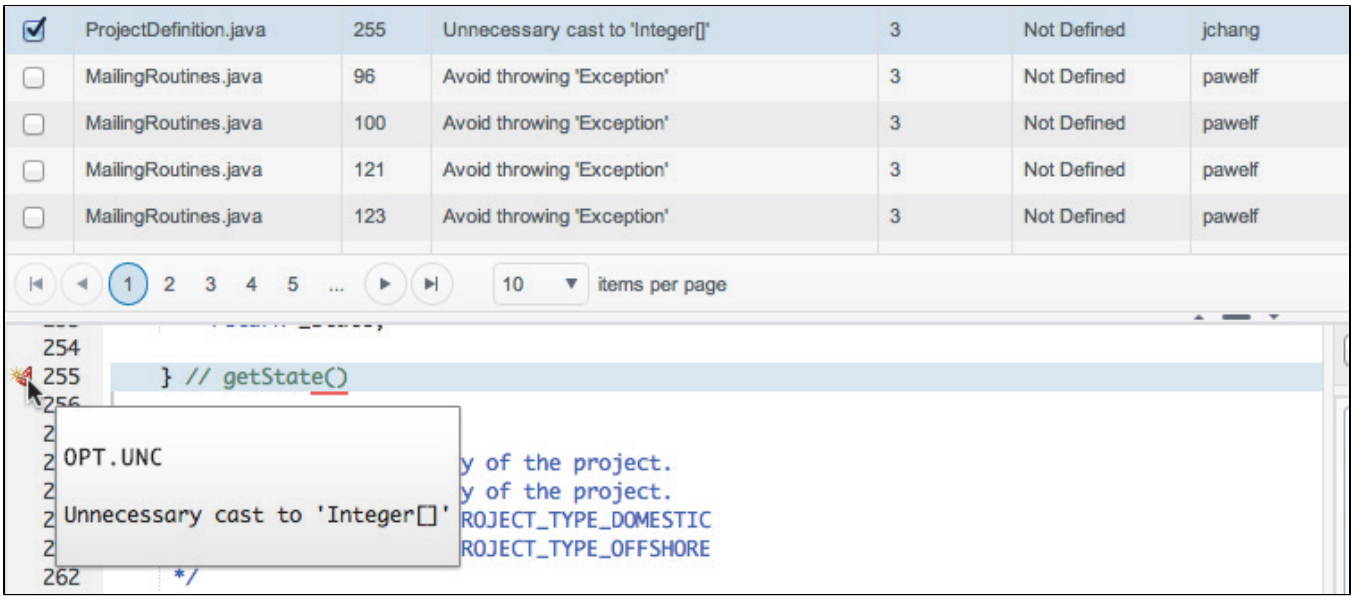
Searching for Violations by File

You can search for a file and return the violations found in the file. The following table provides examples on how to set file paths.

Value	Result

test	Returns all violations with file paths containing the string "Test", for example: • com/parasoft/dtp/SampleTest.java • com/parasoft/Test/Violation • SampleTestProject/trunk/index.html But not: • com/parasoft/dtp/Example.xml
com/ex	Returns all violations with file paths containing the string "com/ex", for example: • com/example/schema.json • branch/dcom/extra/README.md But not: • ex/complete/new.txt
com/parasoft/**	Returns all violations in the "com/parasoft" directory tree, for example: • com/parasoft/dtp/SampleTest.java • com/parasoft/config.xml But not: • com/example/schema.json • main/com/parasoft/example.txt
**/test/*.java	Returns all violations in files with the ".java" suffix under test directories from anywhere in the directory tree, for example: • test/Test.java • com/parasoft/dtp/test/SampleTest2.java But not: • com/parasoft/dtp/test/examples/Example.java • test/license.txt

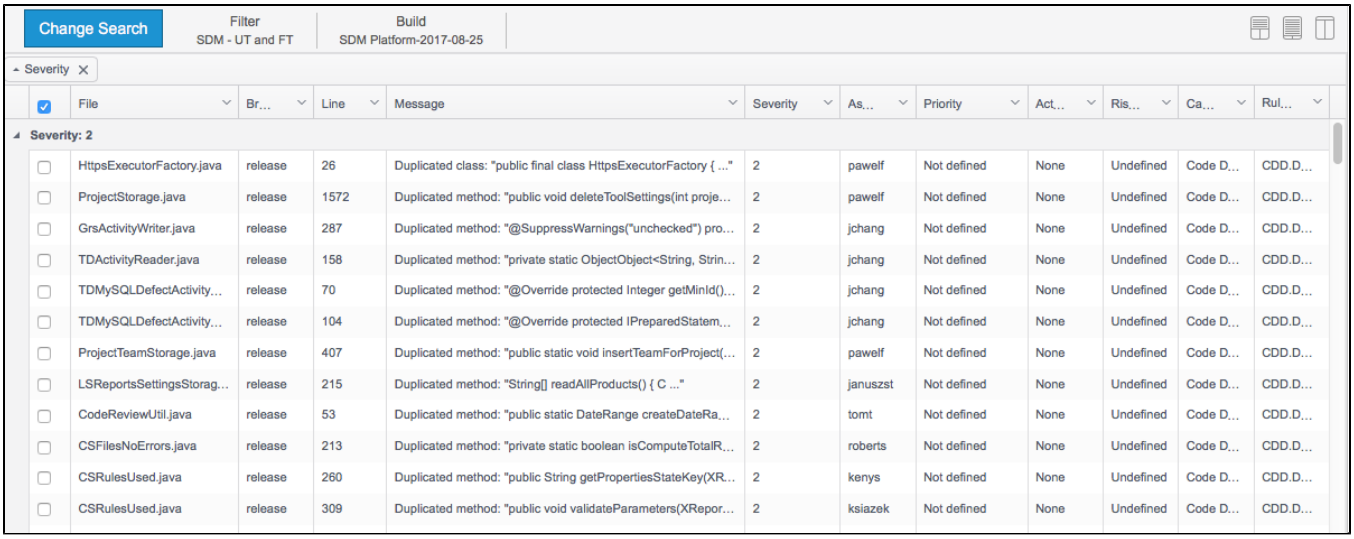
Click on a violation in the search results table to view the violation as it exists in the code.



You can configure DTP to display sources from source control or from sources sent by connected Parasoft tools during code analysis and test execution. See Configuring Source Code Views for additional information on how sources are displayed in DTP.0

Viewing Search Results

The search results panel returns any violations found according the search parameters.



Click on a violation to view the content of the source file, details about the violation, and enable actions for remediation. When you make a selection in the violations table, the file name and the component that opened the file appears in the code panel.

You can also use the sorting mechanisms and customize the table to refine your view of the violation data. See Navigating Explorer Views for details.

By default, the maximum number of violations shown is 1000. You can change the limit by adding the `&limit=[number]` parameter to the URL. For example, the following URL would allow you to see up to 2000 violations:

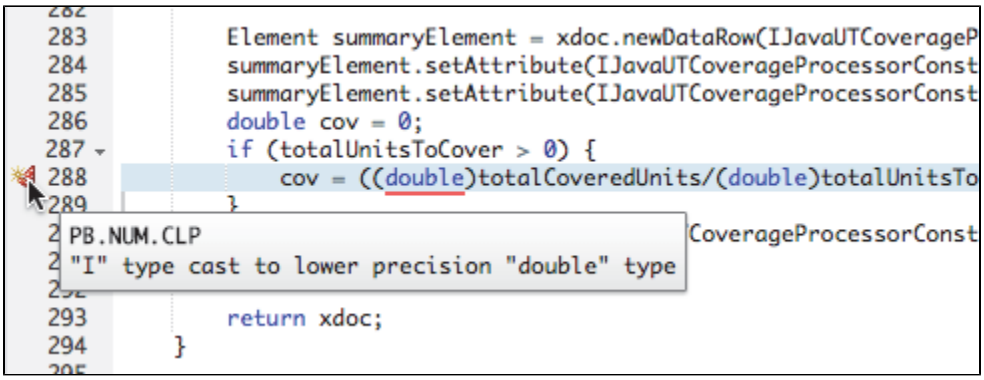
```
[DTP_HOME]/grs/dtp/explorers/violations?filterId=11&limit=2000
```

You can set the limit parameter to any value, but changing the maximum number of violations shown to a large value may affect the performance of the Violation Explorer.

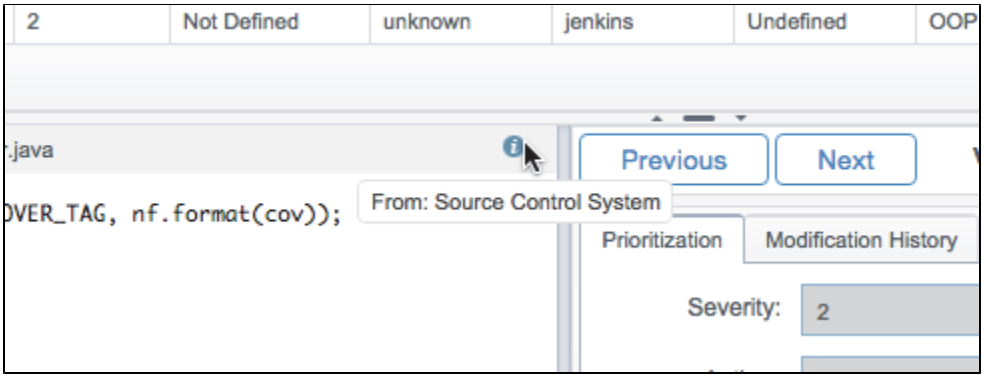
Viewing Sources

The sources panel allows you to view violation instances as they appear in the code. You must have permissions to view source code in Report Center explorer views (see Configuring User Permissions and Groups for additional information).

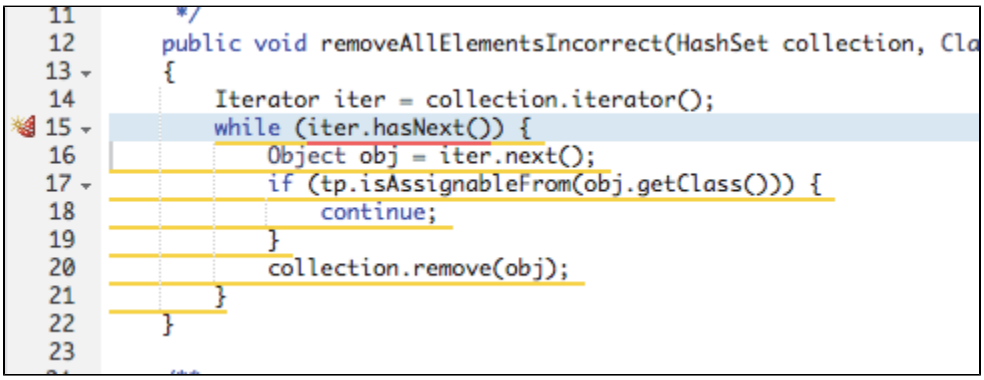
Mouse over the marker in the line number margin to view a tooltip of with the violation error message.



Mouse over the information icon to see where sources are being displayed from.



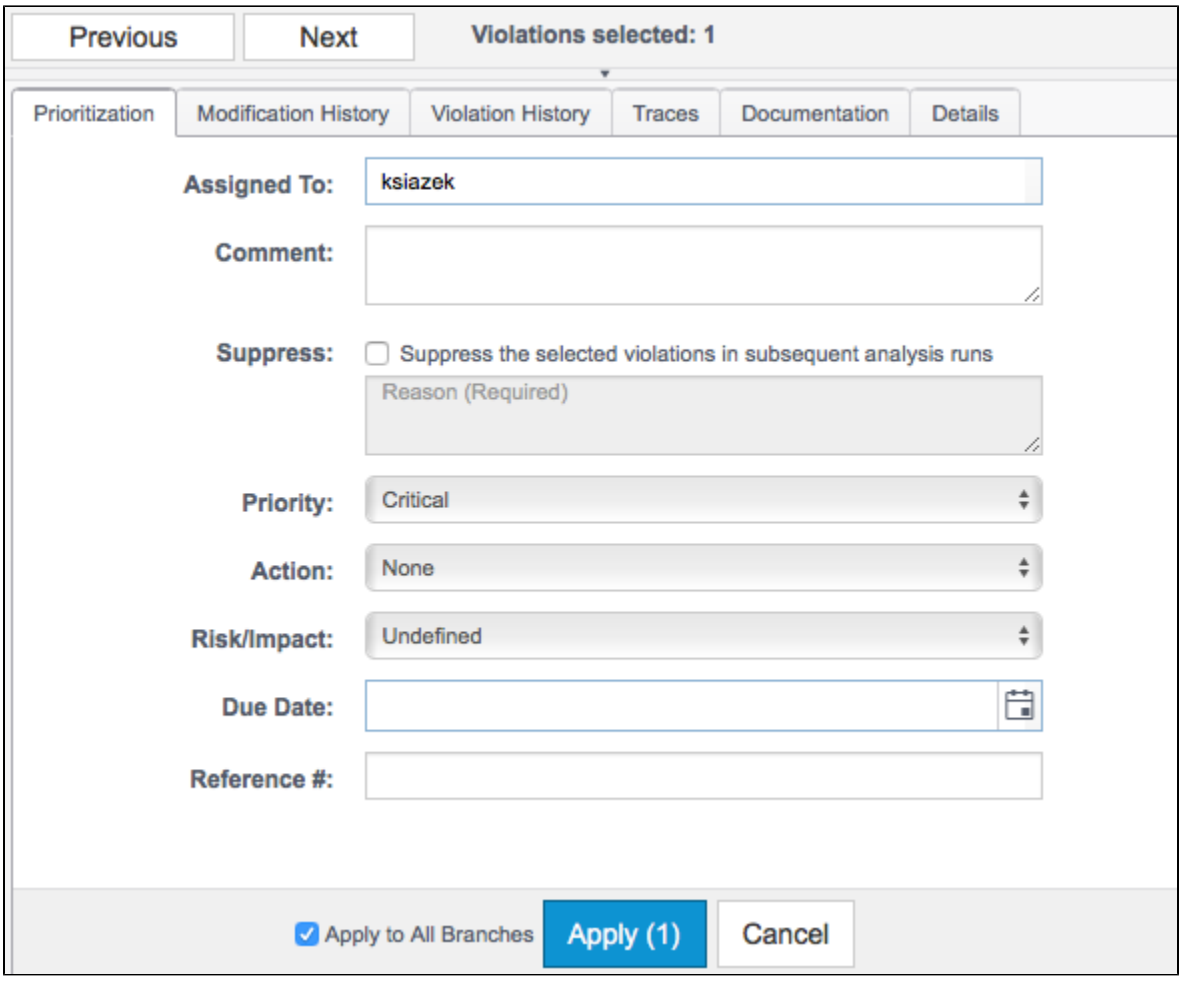
You can also see paths through the code leading to the violation in the code panel when you use the flow analysis trace feature.



See Viewing Flow Analysis Traces for additional information about viewing code in the Violations Explorer.

Addressing Violations

There are several tools in the Violations Explorer to help you address violations in a way that's consistent with your organization's policies, needs, and goals. You can put violations into a software quality workflow through the Prioritization panel.



Users must have permissions to prioritize violations, as well as view sources. Permission to prioritize violations can be granted for all violations or limited to violations owned by the user. The following table describes a project membership scenario and how permissions may be assigned (see Permissions for additional information):

User Type	Additional Permission	Access Granted
Admin		• View sources • Prioritize all
Leader		• View sources • Prioritize all
Member		• View sources • Prioritize owner
Non-member 1		No access
Non-member 2	project	• View project data • Cannot view sources • Cannot prioritize
Non-member 3	project, prioritizeOwner	• Cannot view sources • Prioritize own violations

Non-member 4	project, viewSources	• View sources • Cannot prioritize

Assigning Violations to Developers for Remediation

You can assign violations to other authors of violations or to a member of the Project associated with the Filter.

1. Select violation(s) in the search results area; the file name appears in the code view panel
2. Click the **Prioritization** tab and click the **Assigned To** field.
3. Enter an assignee user name. The form will auto-fill based on the users in the system.
4. Make any other changes and click **Apply**. The **Apply to All Branches** option is enabled by default. Disable this option if you want to apply changes to only the selected instance of the violation; see Applying Changes to Violations.

Adding Comments to Violations

1. Select violation(s) in the search results area
2. Click the **Prioritization** tab and enter a value in the **Comments** field.
3. Make any other changes and click **Apply**. The **Apply to All Branches** option is enabled by default. Disable this option if you want to apply changes to only the selected instance of the violation; see Applying Changes to Violations.

Suppressing Violations

> **Suppressions versus the Do Not Show priority**
>
> In versions of DTP prior to 5.4, the only way to suppress static analysis violations was to set the priority to Do Not Show (see Prioritizing Violations). This approach hides violations in DTP, but is not a programmatic suppression. You can convert violations flagged with the Do Not Show priority into true suppressions. See Upgrade Notes in the 5.4.0 release notes for details.

1. Select violation(s) in the search results area and click the **Prioritization** tab.
2. Enable the **Suppress the selected violations in subsequent analysis runs** option and provide information about the suppression in the Reason text field. The suppression will be implemented in the next static analysis execution. You can release suppressed violations by disabling this option. Changes will be implemented during the next analysis run.
3. Make any other changes and click **Apply**. The **Apply to All Branches** option is enabled by default. Disable this option if you want to apply changes to only the selected instance of the violation; see Applying Changes to Violations.

Prioritizing Violations

1. Select violation(s) in the search results area; the file name appears in the code view panel.
2. Click the **Prioritization** tab and choose a priority from the drop-down menu.
3. Make any other changes and click Apply; the Apply to All Branches option is enabled by default. Disable this option if you want to apply changes to only the selected instance of the violation; see Applying Changes to Violations.

Assigning Actions to Violations

Actions are strings of metadata that you can use to define how you choose to remediate reported violation. DTP ships with set of pre-defined actions: None, Fix, Reassign, Review, Suppress, and Other. You can edit or remove the pre-defined action types (except for the None type) using the API. For details on configuring actions, choose **API Documentation** from the **Help** drop-down menu in the Report Center navigation bar.

1. Select violation(s) in the search results area
2. Click the **Prioritization** tab and choose a value from the **Action** drop-down menu.
3. Make any other changes and click **Apply**. The **Apply to All Branches** option is enabled by default. Disable this option if you want to apply changes to only the selected instance of the violation; see Applying Changes to Violations.

Assigning Violation Risk and Impact Levels

The Violations Explorer allows you to flag violations that pose a risk or have an impact on the policy goals associated with your application.

1. Select violation(s) in the search results area.
2. Click the **Prioritization** tab and choose a value from the **Risk/Impact** drop-down menu.
3. Make any other changes and click **Apply**. The **Apply to All Branches** option is enabled by default. Disable this option if you want to apply changes to only the selected instance of the violation; see Applying Changes to Violations.

Assigning Due Dates to Violations

1. Select violation(s) in the search results area
2. Click the **Prioritization** tab and click the calendar icon in the **Due Date** field to choose a date.

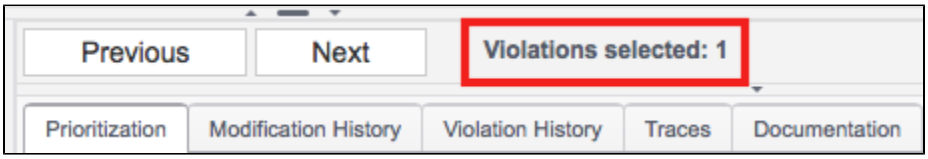
3. Make any other changes and click **Apply**. The **Apply to All Branches** option is enabled by default. Disable this option if you want to apply changes to only the selected instance of the violation; see Applying Changes to Violations.

Assigning Reference Numbers to Violations

1. Select violation(s) in the search results area
2. Click the **Prioritization** tab and enter a value in the **Reference #** field.
3. Make any other changes and click **Apply**. The **Apply to All Branches** option is enabled by default. Disable this option if you want to apply changes to only the selected instance of the violation; see Applying Changes to Violations.

Applying Changes to Violations

When you update a violation, you can apply the change to single instance of the violation or apply the changes to the violation in all source control branches in which it occurs. A confirmation message appears when your changes are applied:

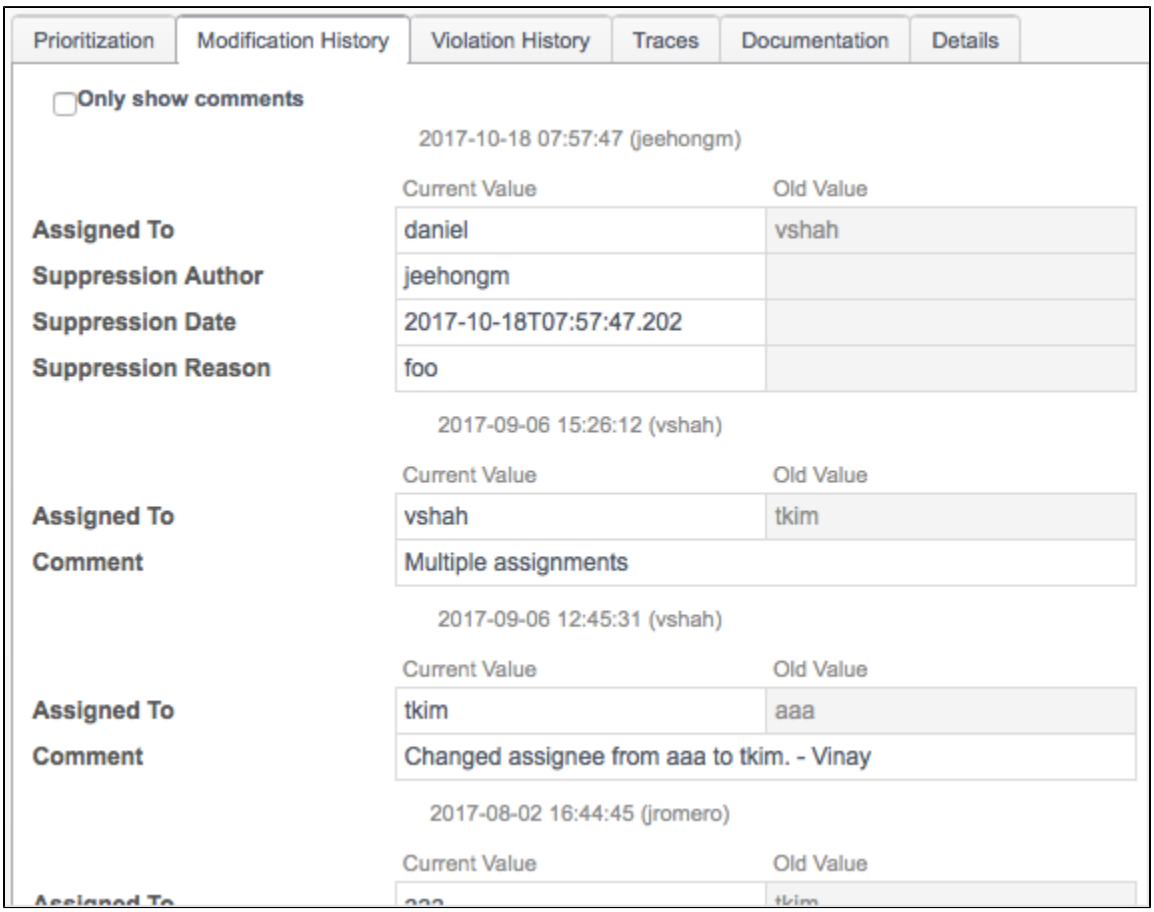


Reviewing Violation Information

All changes applied to violations can be viewed in the actions panel, which provides a detailed view of historical information associated with selected violations. Rule documentation for a selected violation is also available.

Modification History

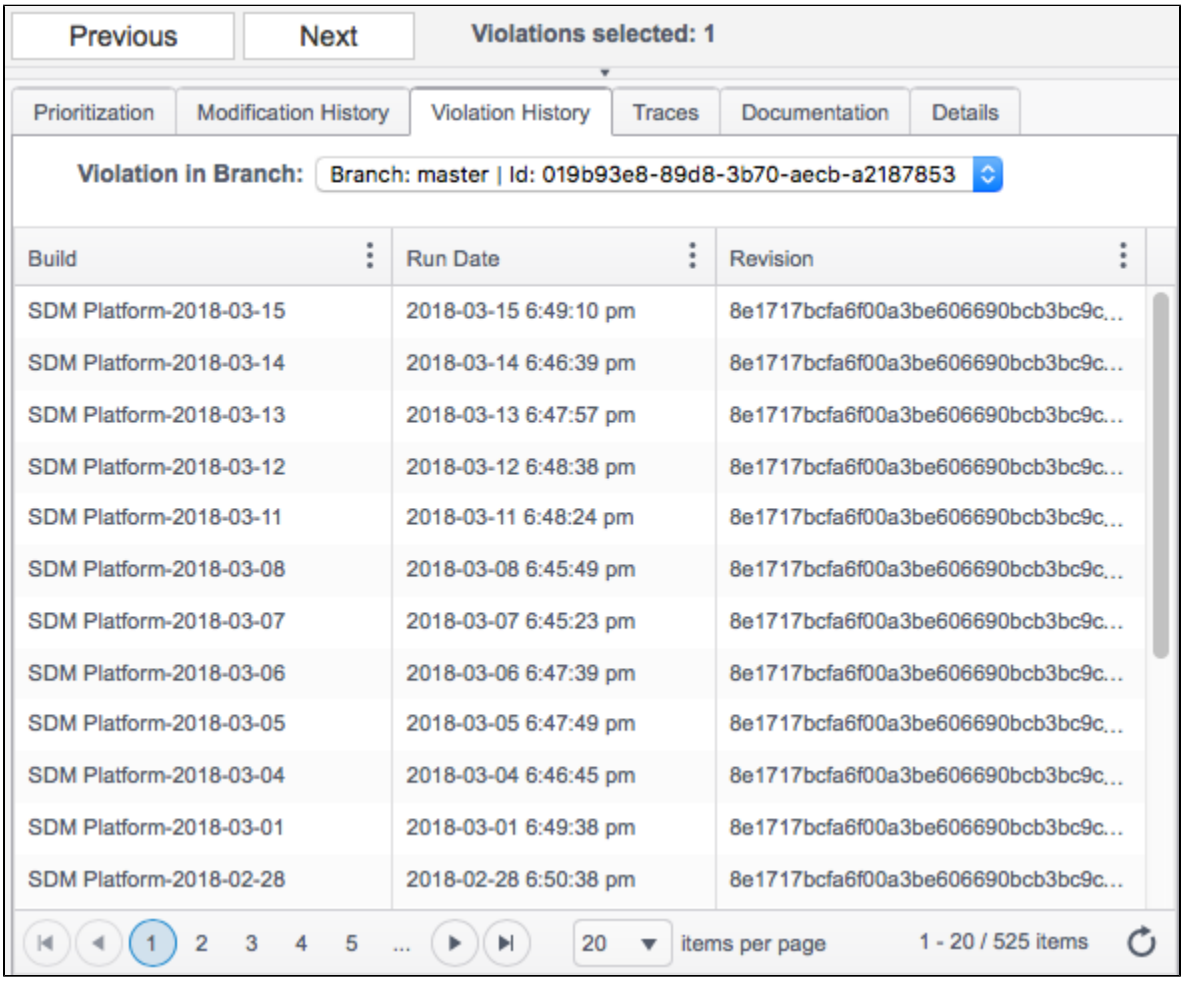
Click the **Modification History** tab in the actions panel to view a summary of prioritization changes, such as re-assignments and impact level changes, for the violation. You can not view the modification history of two or more violations.



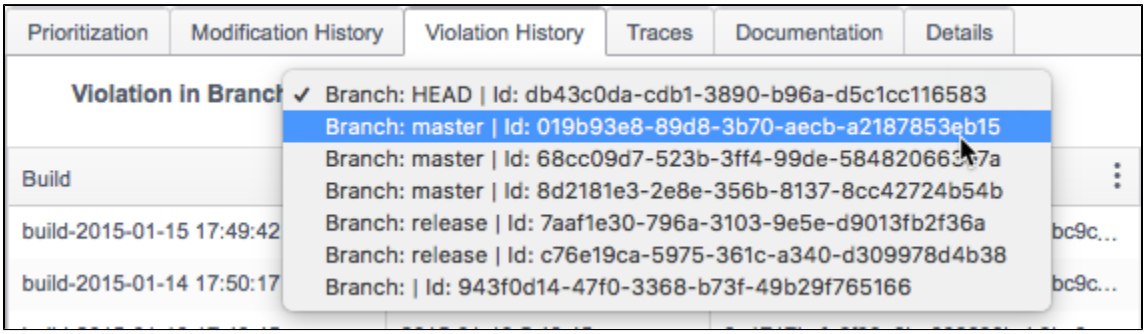
Enable the **Only show comments** option to hide all updates except for the comments log.

Violation History

Click the **Violation History** tab in the actions panel to view the static analysis runs and the dates in which the selected violation was detected. You can not view the violation history of two or more violations.

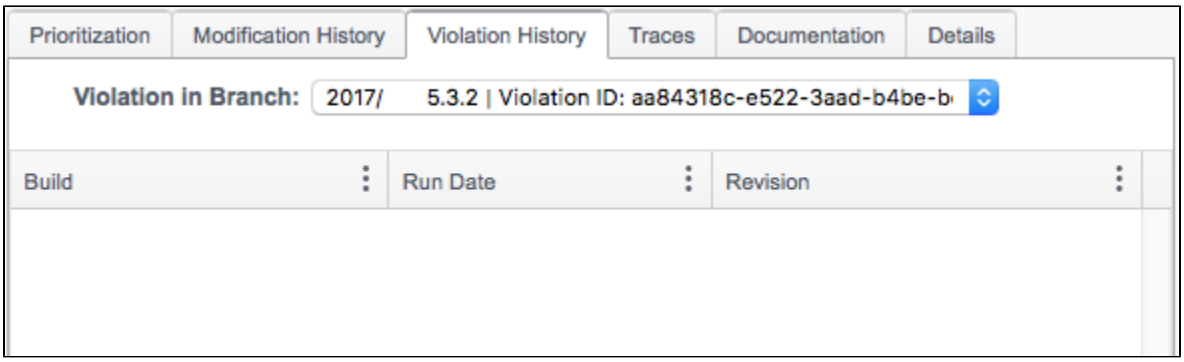


The tab also shows the source control history associated with the violation. The same violation may exist in multiple branches under different violation IDs. Choose a branch from the **Violations in Branch** drop-down menu to view the history of the selected violation in other branches.



The table in the Violation History will be refreshed if you switch to a different branch, *but other areas of the Violations Explorer will continue to show information for the selected instance of the violation*.

The violation history table will be empty if the build information was removed from DTP, i.e., during regular database pruning or manual deletion.



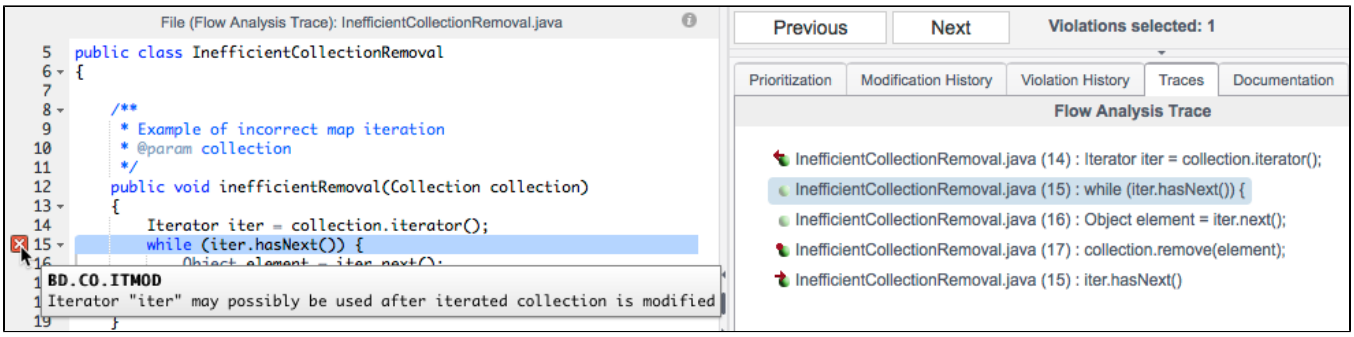
Traces

Click the **Traces** tab to view flow analysis results or CDD (code duplicate detection) analysis rules if either type of analysis was performed.

Viewing Flow Analysis Traces

If data flow analysis (dynamic analysis) has been performed, the path leading up to a violation appears under the Traces tab. Flow analysis can help you make decisions about how the code is structured, understand why the violation may have occured, and determine the significance of the violation.

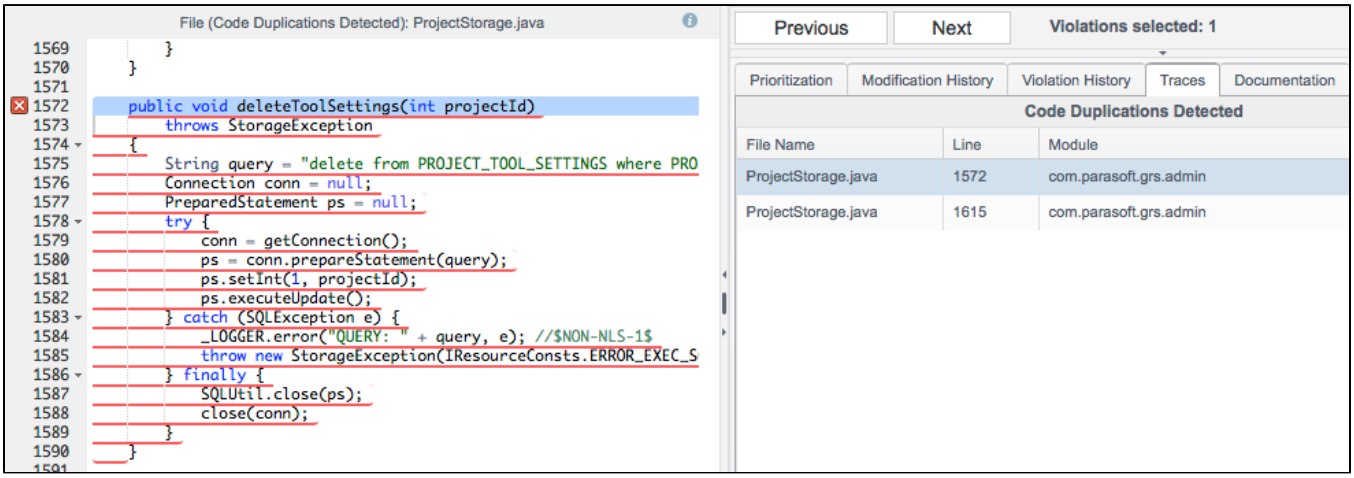
Click on a trace to view the violation path in the code panel.



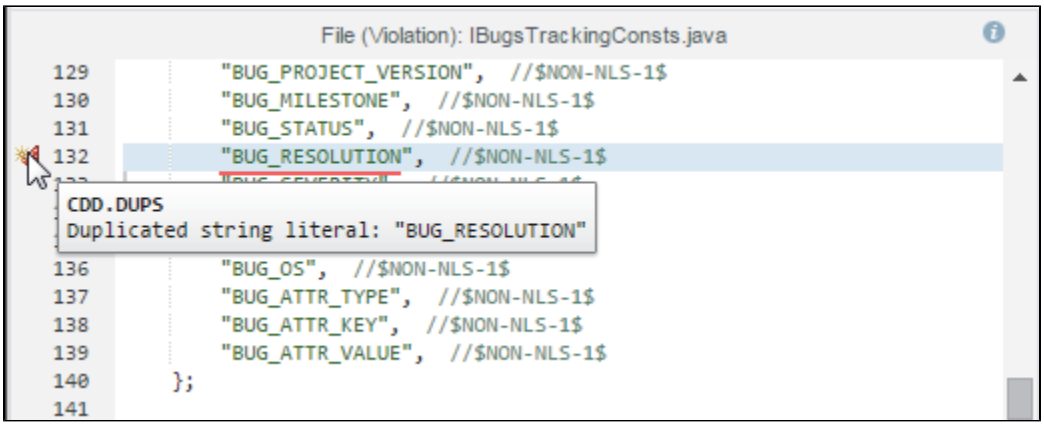
Users must have permissions to view source code. See Permissions for additional information.

Viewing Duplicate Code Violations

If violations were detected by CDD (code duplicate detection) analysis rules, then you can view them in the Traces tab. Duplicate code may indicate poor application design, as well as increase maintenance costs. Click on a CDD violation in the Violations Explorer search results panel to open the violation path.



This panel shows the file name, line number, and path to each instance of the duplicated code. DTP also shows the sources containing the duplicate code in the sources panel.



You must have permissions to view source code. See Permissions for additional information.

Click on entries in the Code Duplications Detected panel to view instances of the duplicated code.

You can perform normal violation remediation actions, such as setting a priority and assigning the violation to a developer. See Addressing Violations.

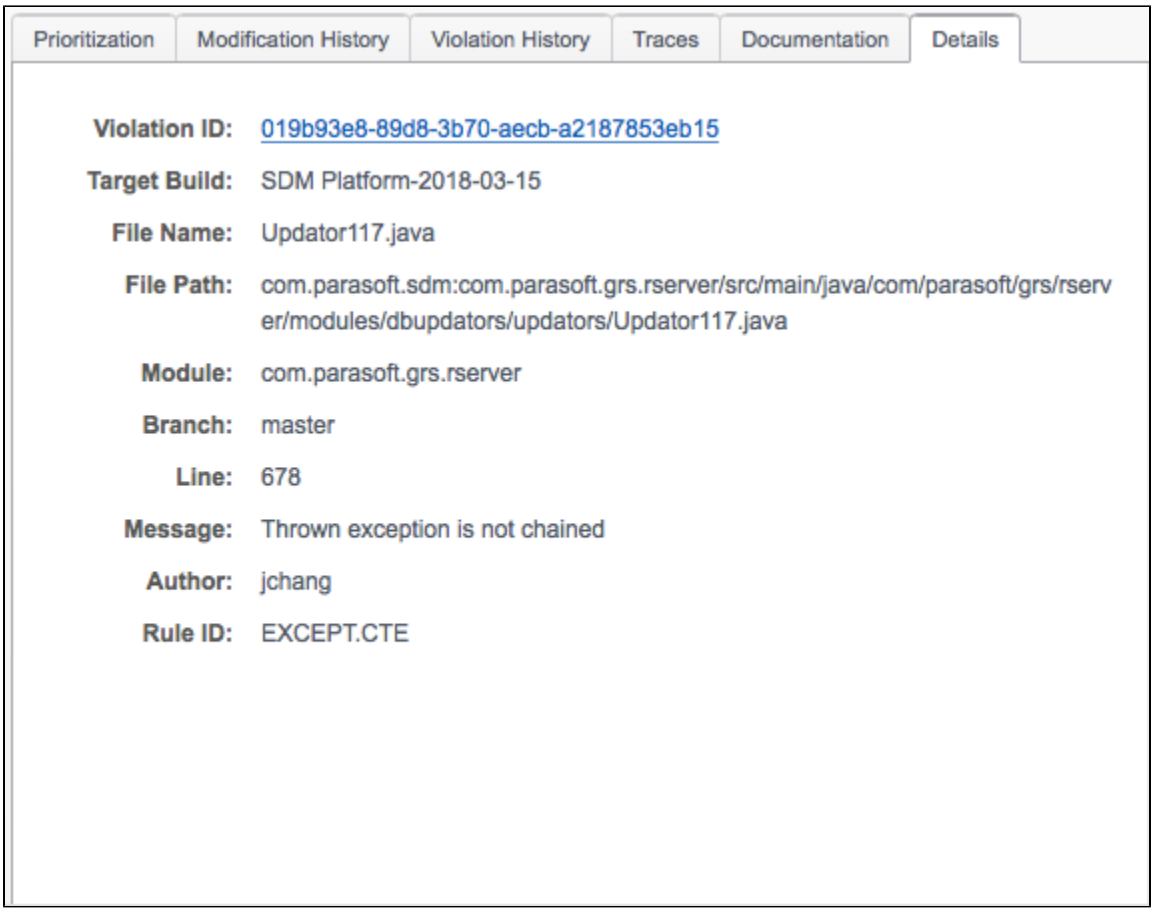
Documentation

Click the **Documentation** tab to view the static analysis rule that the code violates. You can not view the rule documentation for two or more violations.

Details

Click the **Details** tab in the actions panel to view current information about the location, owner, rule ID, and message associated with the selected violation.

The **Violation ID** field appears if a violation is selected in the search results table. The ID links directly to the violation and the selected filter. You can share this link so that others can directly access view the violation in DTP.


Prioritization
Modification History
Violation History
Traces
Documentation
Details
Violation ID: 019b93e8-89d8-3b70-aecb-a2187853eb15
Target Build: SDM Platform-2018-03-15
File Name: Updator117.java
File Path: com.parasoft.sdm:com.parasoft.grs.rserver/src/main/java/com/parasoft/grs/rserver/modules/dbupdators/updators/Updator117.java
Module: com.parasoft.grs.rserver
Branch: master
Line: 678
Message: Thrown exception is not chained
Author: jchang
Rule ID: EXCEPT.CTE