CERT C++ Compliance

The Parasoft CERT C++ Compliance extension is a set of assets for your DTP infrastructure that enable you to demonstrate compliance with CERT C++ Coding Standard guidelines. The extension is shipped as part of the Security Compliance Pack. Contact your Parasoft representative to download and license the Security Compliance Pack.

In this section:

Background

The CERT C++ Coding Standard was developed by the CERT Coordination Center to improve the safety, reliability, and security of software systems. CERT coding standards consist of "rules" and "recommendations" and are organized into a set of categories. Rules provide code requirements for adhering to the standard, whereas recommendations are intended to provide guidance that improves the safety, reliability, and security of software systems.

Rules and recommendations are collectively referred to as "guidelines." Guidelines in the CERT C++ Secure Coding Standard are cross-referenced with Common Weakness Enumeration (CWE) entries. Programming patterns that fail to meet CWE's guidelines are called "weaknesses."

In terms of risk analysis, CERT uses three metrics to help quantify weaknesses:

- the *severity* of the consequences associated with a failure to comply with the rule
- the *likelihood* that a coding flaw introduced by ignoring the rule will result in an exploitable vulnerability
- the *remediation cost* associated with complying with the rule

The metrics are used to prioritize violations into three levels: L1 (highest priority), L2, and L3. The CERT C++ Compliance extension configures your DTP implementation to show static analysis violations according to their CERT C++ priority, guideline, type, and guideline category.

See https://wiki.sei.cmu.edu/confluence/pages/viewpage.action?pageId=88046682 to learn more about about the standard.

Prerequisites

C/C++test 2020.1 or later (desktop or plug-in edition) with the SEI CERT C++ Rules and Flow Analysis license features enabled. See Security Compliance Pack for additional information.

Process Overview

1. Install the Security Compliance Pack into DTP Extension Designer.
2. Deploy the CERT C++ Compliance artifact into your DTP environment. This also deploys the CERT C++ Compliance extension assets.
3. Analyze code with C/C++test using the SEI CERT C++ Rules test configuration and report violations to DTP. You can configure C/C++test to use the local test configuration or the test configuration shipped with the Security Compliance Pack. The test configuration and rulemap.xml file configures analysis rules to report violations according to CERT C++ guidelines.
4. Add the CERT C++ Compliance dashboard and widgets to your DTP interface. The dashboard widgets and shows the reported violations within the context of CERT C++ guidelines.
5. Interact with the widgets and reports to identify code that needs to be fixed, as well as print out the reports for auditing purposes.

CERT C++ Compliance Extension Assets

The Parasoft CERT C++ Compliance extension helps you create the documentation required for demonstrating compliance with CERT C++. The following artifacts are included in the package.

Rule Map and Test Configuration

Parasoft static and flow analysis rules normally report violations according to a category (e.g., Possible Bug, Interoperability, etc.) and severity (i.e., 1-5). In order to view code analysis violations as CERT C++ guideline violations, DTP requires a rule map file that realigns Parasoft rules to report violations according to CERT C++ guidelines. In addition, the code analysis tool (C/C++test) needs a test configuration file that ensures that only the rules related to the remapped CERT C++ rules are executed. These files are shipped with C/C++test.

Category and Guideline Definition Files

The following configuration files shipped with the extension provide compliance categories in DTP interfaces that are oriented toward CERT C++:

- **CERT_CPP-Categories.xml**
- **CERT_CPP-Guideline-Recommendation.xml**
- **CERT_CPP-Guideline-Rule.xml**
- **CERT_CPP-Guideline.xml**
- **CERT_CPP-Priority.xml**

Test Configurations

You can configure C/C++test to run the test configurations shipped with the tool or with the Security Compliance Pack. Refer to the C/C++test documentation for details. The following test configurations are included with the pack:

- CERT_C_FOR_CPP [10.4.2]. properties (see CERT C Compliance)
- CERT_CPP [10.4.2].properties

Dashboards

After installing the Security Compliance Pack and deploying the CERT compliance artifacts, you will be able to quickly add widgets configured to show CERT-related data by using the following dashboard templates:

- SEI_CERT_C_Compliance.json (see CERT C Compliance)
- SEI_CERT_CPP_Compliance.json

SEI_CERT_CPP_Compliance.json

This file adds the CERT C++ Compliance dashboard template to DTP. See Custom Dashboard Templates for additional information about understanding dashboard templates.

CERT Compliance.json

This is the DTP Workflow you must install and deploy in Extension Designer. It extends DTP's data processing functionality to produce dashboard widgets and reports specific to CERT C++. It helps you track compliance status and document guideline enforcement, deviations, and rule re-categorization.

Model and Profiles

Profiles provide a range of functions in a DTP infrastructure, such as providing inputs for custom calculations executed by an extension and providing data for compliance reports. Profiles take their structure from models, which define fields, headers, or other components used in the profile. See Working with Model Profiles for information about understanding profiles in DTP Enterprise Pack. The following profiles are included with the CERT C++ artifact.

- **cert-cpp-compliance.json:** This model file describes how the cert-cpp-2018.json profile renders the data.
- **cert-cpp-2018.json**: This is the default profile that renders data according to the CERT Compliance.json model. This profile should be enabled to generate compliance audit reports.
- **cert-cpp-likelihood.json:** This profile provides metric information for key performance indicator (KPI) calculations. It renders data according to the KPI.json model.
- **cert-cpp-remediation-cost.json**: This profile provides metric information for KPI calculations. It renders data according to the KPI.json model.

KPI.json

This profile extends the Key Performance Indicator artifact so that metrics widgets can show metrics information related to CERT C++ guidelines. The profile renders the data calculated by the cert-cpp-likelihood.json and cert-cpp-remediation-cost.json profiles.

> **Key Performance Indicator Extension is Required**
>
> In order to leverage the metrics calculations enabled by the KPI assets, install and deploy the Key Performance Indicator artifact.

Cross-reference PDF

For your convenience, a PDF that shows the association between Parasoft rules and CERT guidelines is located in the <PACK>/rules/cpptest directory.

package.json

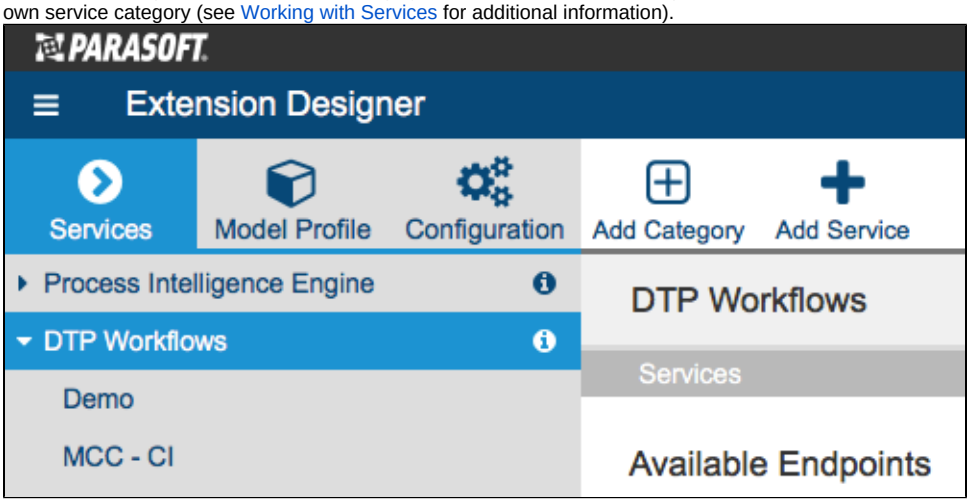
This file describes the contents of the extension.

Deploying the CERT C++ Compliance Assets

The CERT C++ Compliance files are installed as part of the Security Compliance Pack (see Installation for instructions). After installing the artifact, you must deploy the assets to your DTP environment.

> **CERT C and CERT C++**
>
> If you are already using the CERT C Compliance artifact, you do not need to perform this step. Both artifacts use the same DTP Workflow.

1. Choose **Extension Designer** from the DTP settings (gear icon) menu.
2. Click the **Services** tab and expand the **DTP Workflows** services category. You can deploy assets under any service category you wish, but we recommend using the DTP Workflows category to match how Parasoft categorizes the assets. You can also click **Add Category** to create your own service category (see Working with Services for additional information).



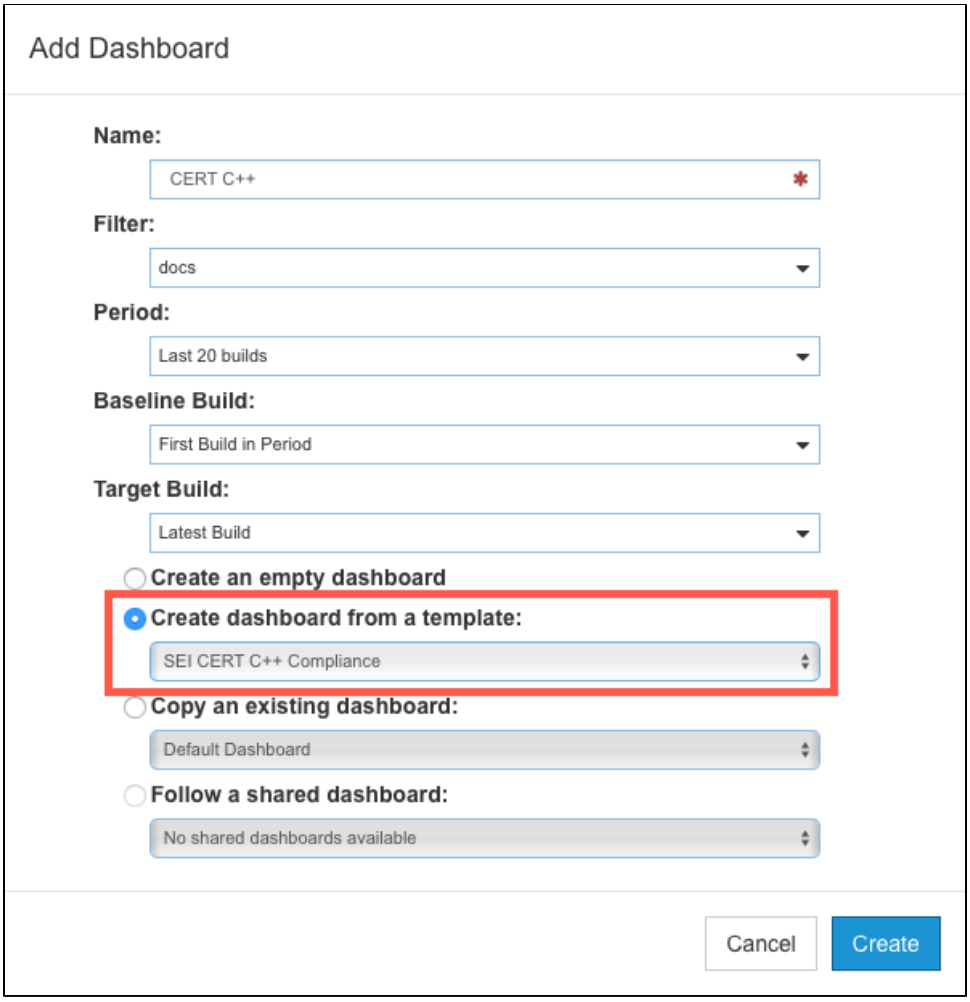
3. You can deploy the artifact to an existing service or add a new service. The number of artifacts deployed to a service affects the overall performance. See Extension Designer Best Practices for additional information. Choose an existing service and continue to step 5 or click **Add Service**.
4. Specify a name for the service and click **Confirm**.
5. The tabbed interface helps you keep artifacts organized within the service. Organizing your artifacts across one or more tabs does not affect the performance of the system. Click on a tab (or click the **+** button to add a new tab) and click the vertical ellipses menu.
6. Choose Import> Library> Workflows> Security> CERT Compliance and click anywhere in the open area to drop the artifact into the service.
7. Click **Deploy** and return to your DTP dashboard and refresh your browser.

You can now add CERT C++ Compliance widgets.

Adding the CERT C++ Compliance Dashboard

The CERT C++ dashboard template will be available after installing the Security Compliance Pack. If you do not see dashboard template, restart DTP (see Stopping DTP Services and Starting DTP Services).

1. Click **Add Dashboard** in the DTP toolbar and specify a name when prompted.
2. (Optional) You can configure the default view for the dashboard by specifying the following information:
    a. Choose the filter associated with your project in the filter drop-down menu. A filter represents a set of run configurations that enabled custom views of the data stored in DTP. See DTP Concepts for additional information.
    b. Specify a range of time from the Period drop-down menu.
    c. Specify a range of builds from the Baseline Build and Target Build drop-down menus.

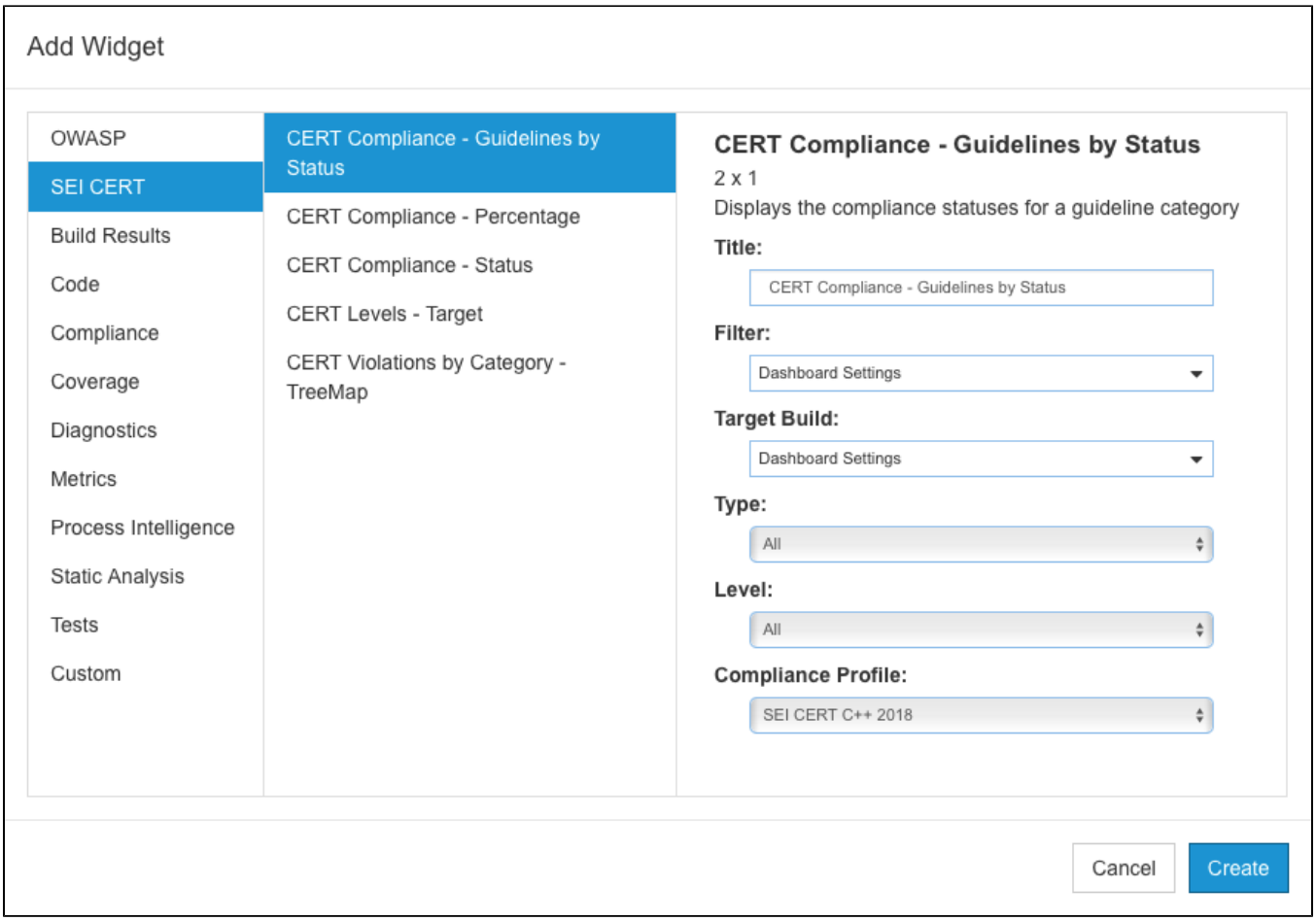


3. Enable the **Create dashboard from a template** option and choose the SEI CERT C++ Compliance template.
4. Click **Create** to finish adding the dashboard.

If you have already executed C/C++test on your project using the SEI CERT C++ test configuration, most widgets will render data as soon as the dashboard is added. You can immediately begin using these widgets and working with the data to help you track your compliance goals (see CERT C++ Compliance Widgets). Additional steps, however, are necessary to use the Remediation Cost and Likelihood Score widgets, which rely on calculations executed by the KPI extension. See Enabling the CERT KPI Widgets for instructions.

Manually Adding the CERT C++ Widgets

You can manually add the CERT C++ widgets to an existing dashboard. See Adding Widgets for general instructions on how to add widgets to a dashboard. After deploying the artifact, widgets will appear in the SEI CERT category.



The following configurations are available:

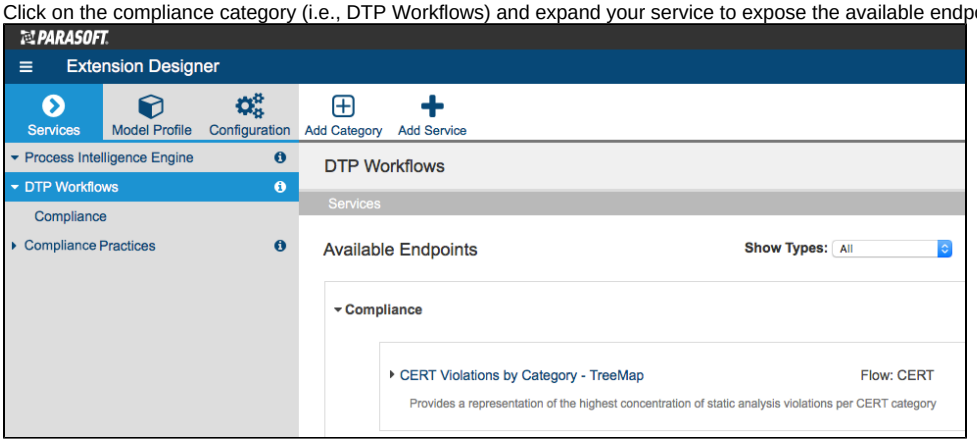
Title	You can rename the widget in the Title field. This setting is available for all widgets.
Filter	Choose a specific filter or Dashboard Settings from the drop-down menu. See Creating and Managing Filters for additional information. This setting is available for all widgets.
Target Build	Choose a specific build from the drop-down menu. The build selected for the entire dashboard is selected by default. See Using Build Administration for additional information about understanding builds. This setting is available for all widgets.
Type	This rule specifies which type of guideline you want to view in the widget. Choose either Rule, Recommendation, or All from the drop-down menu. See Background for additional information about guideline types. This setting is available for the following widgets: • CERT Compliance - Guidelines by Status • CERT Levels - Target • CERT Violations by Category - TreeMap
Level	This rule specifies which priority level you want to view in the widget. Choose either L1, L2, or L3 from the drop-down menu. See Back ground for additional information about guideline priorities. This setting is available for the following widgets: • CERT Compliance - Guideline by Status • CERT Compliance - Percentage • CERT Violations by Category - TreeMap
Compliance Profile	Specify the compliance profile you want to use to view the data. In most cases, this should be the default profile shipped with the extension (see CERT C++ Compliance Profile). This setting is available for all widgets.

Enabling the CERT KPI Widgets

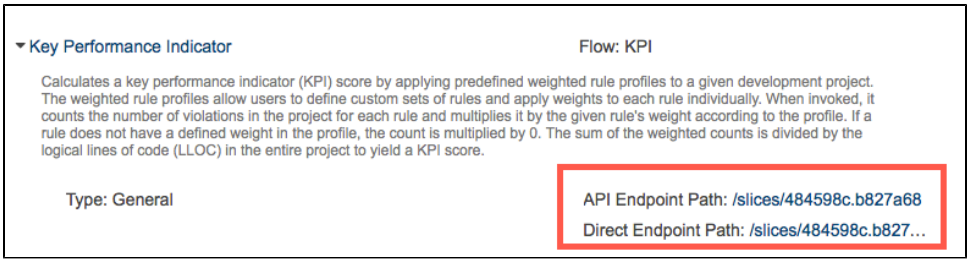
The Remediation Cost and Likelihood Score widgets are implementations of native Metrics - Summary DTP widgets configured to present CERT-specific metrics data. The KPI extension performs custom calculations on the metrics data sent to DTP according to the SEI CERT C++ Remediation Cost and SEI CERT C++ Likelihood KPI profiles. The processed metrics are reported in the widgets.

In order to use these widgets, you must deploy and execute the KPI extension. C/C++test must also be executed with the Metrics test configuration under the same build ID so that the build has both static analysis and metrics data. Refer to the C/C++test documentation for details about setting the build ID and executing the Metrics test configuration.

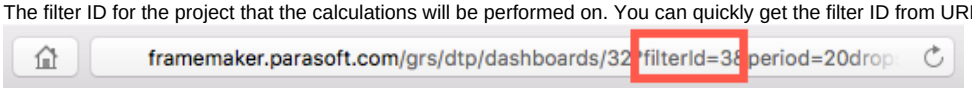
1. Choose **Extension Designer** from the DTP settings (gear icon) menu and click the **Services** tab.
2. Choose a service category and a service for the extension. We recommend deploying the KPI extension to the DTP Workflows category.
3. Open the vertical ellipses menu and choose **Import> Library> Workflows> Security> Key Performance Indicator**.
4. Click anywhere in the space to drop the flow into the service tab and click **Deploy**.
5. Click on the compliance category (i.e., DTP Workflows) and expand your service to expose the available endpoints.



6. Expand the Key Performance Indicator section and copy the endpoint. Extension Designer presents two paths for the endpoint. The API Endpoint Path includes all API directories and can be used for exercising the endpoint in most cases. The Direct Endpoint Path is the direct path to the endpoint on the server and can be used if the API endpoint path is blocked or inaccessible, such as in some third-party integrations that require authentication.



7. Send a REST request to the endpoint along with the required parameters. You can execute the request in a browser, using a cURL command, or add it to a script. The following table describes the required parameters:

filterId	The filter ID for the project that the calculations will be performed on. You can quickly get the filter ID from URL of your dashboard.  You can also get the filter ID from the the Filters settings in DTP administration (see Creating and Managing Filters).
profile	Profile name with the rules and weights to use for the calculations.
buildId	The build id for which the calculations will be performed on. If no build ID is provided, this parameter defaults to the latest build.

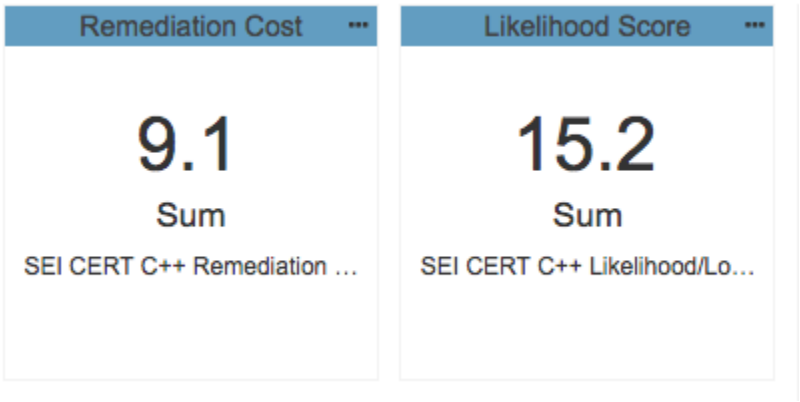
Example API Call URL

```
http://framemaker:8314/categories/5ae39f928550880f5026fc80?filterId=3&profile=SEI%20CERT%20C%2B%2B%20Likelihood
```

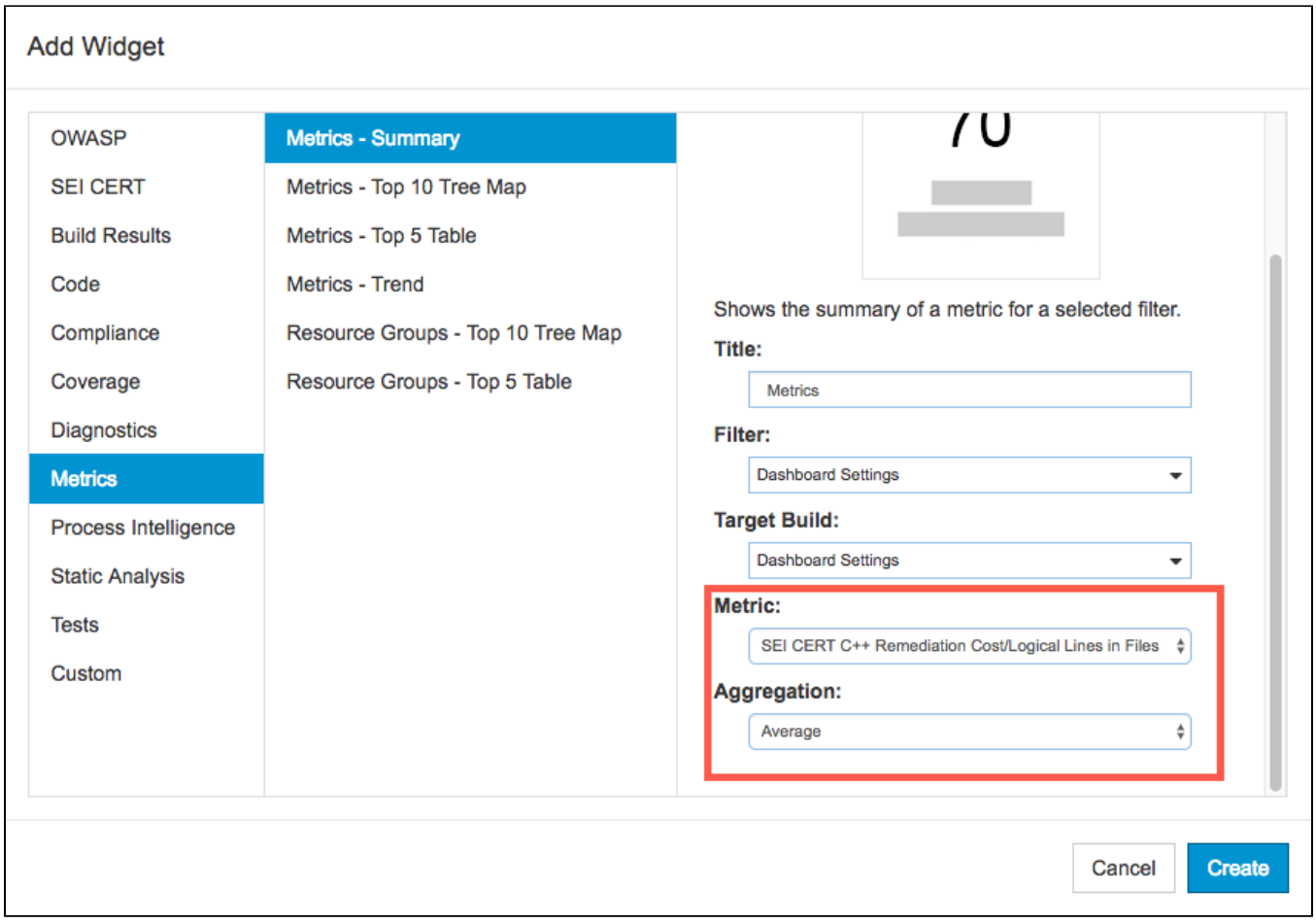
Example of Successful Response

```
{{success: {title: "KPI", message: "Calculation has successfully started for filter 'docs' using profile 'SEI CERT C++ Likelihood'."}}
```

Metrics-related calculations are long-running processes and may take several minutes to execute depending on how much data you have to process. After the calculation completes, refresh the widgets (if already on your dashboard) to view the data. The KPI extension only needs to be deployed once, but you must invoke the API separately for each profile, i.e., SEI CERT C++ Likelihood and SEI CERT C++ Remediation Cost.



If you are not using the CERT C++ dashboard template or want additional views of the metrics, you can manually add instances of the native Metrics - Summary DTP widget to your dashboard and configure them to use the SEI CERT C++ Likelihood and SEI CERT C++ Remediation Cost metrics, as well as set the aggregation value:



You can click on a widget to open the Single Metric Overview Report.

CERT C++ Compliance Widgets

The following widgets are shipped with the CERT C++ Compliance DTP Workflow to help you achieve CERT C++ Compliance goals.

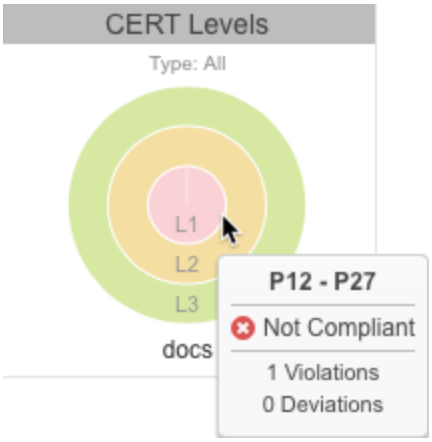
CERT Compliance - Status

This widget provides an overview of the project's CERT compliance status.

By default, the widget shows Rules and Recommendations, as well as all priority levels. You can add multiple instances of the widget and configure different combinations to create robust views of the compliance status. Click on the widget to open the CERT C++ Compliance Report.

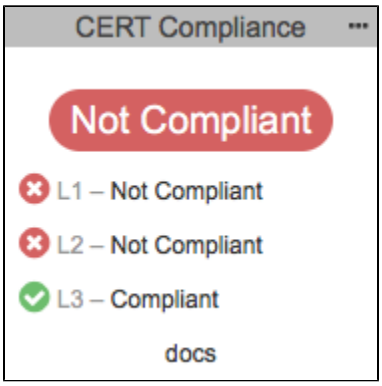
CERT Levels - Target

This widget shows the highest concentration of static analysis violations per CERT category. It provides an overview of the compliance status, as well as applicable deviations, in the tooltip. Click on the widget to open the CERT C++ Compliance Report.



CERT Compliance - Status

The widget shows the overall compliance status, as well as the compliance status for each CERT level. You can add multiple instances of the widget configured to use a different profile, e.g., a profile with disabled guidelines, to view your current compliance status. Click on the widget to open the CERT C++ Compliance Report.

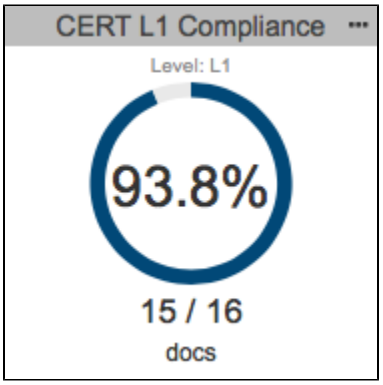


The code can be compliant with deviations and violations that have been deemed acceptable. See Deviation Report for additional information about deviations.

The status will be set to Not Compliant if Parasoft code analysis rules documented in your profile were not included in the specified build or if unacceptable violations have been reported. Make sure all rules are enabled in C/C++test and re-run analysis.

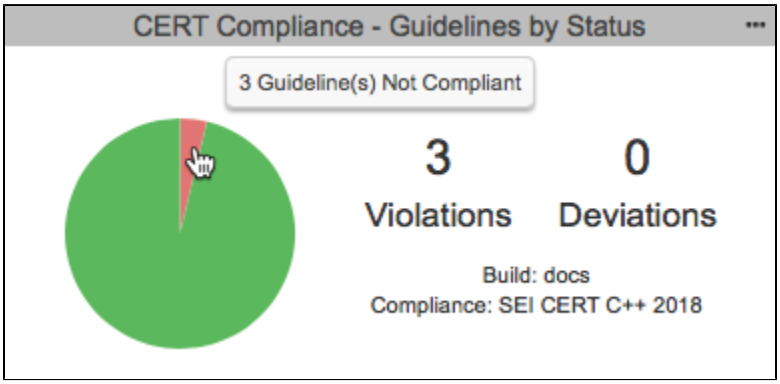
CERT Compliance - Percentage Widget

This widget shows the completeness of CERT compliance as a percentage. Completeness is based on the number of guidelines being enforced in the profile. The CERT C++ dashboard includes three instances of this widget, one for each level. Click on the widget to open the CERT C++ Compliance Report.



CERT Compliance - Guidelines by Status

This widget shows the compliance status for a specific Rule or Recommendation per priority level.



You can add multiple instances of the widget configured to different type/priority level combinations to help you understand your compliance status from different perspectives. The pie chart can represent up to four different guideline statuses for the selected category:

Green	Guidelines your code is in compliance with for the selected type and level.
Yellow	Guidelines that your code is deviating from but are still considered compliant. A deviation is when the guideline is not being followed according to the Parasoft static analysis rule, but is considered acceptable because it does not affect the safety of the software. Deviations represent Parasoft static analysis rules that have been suppressed.
Orange	Guidelines that your code is considered compliant with, even though the static analysis rules that enforce them contain violations. **Only Recommendations can have this status.**
Red	Guidelines that your code is not compliant with.

You can perform the following actions:

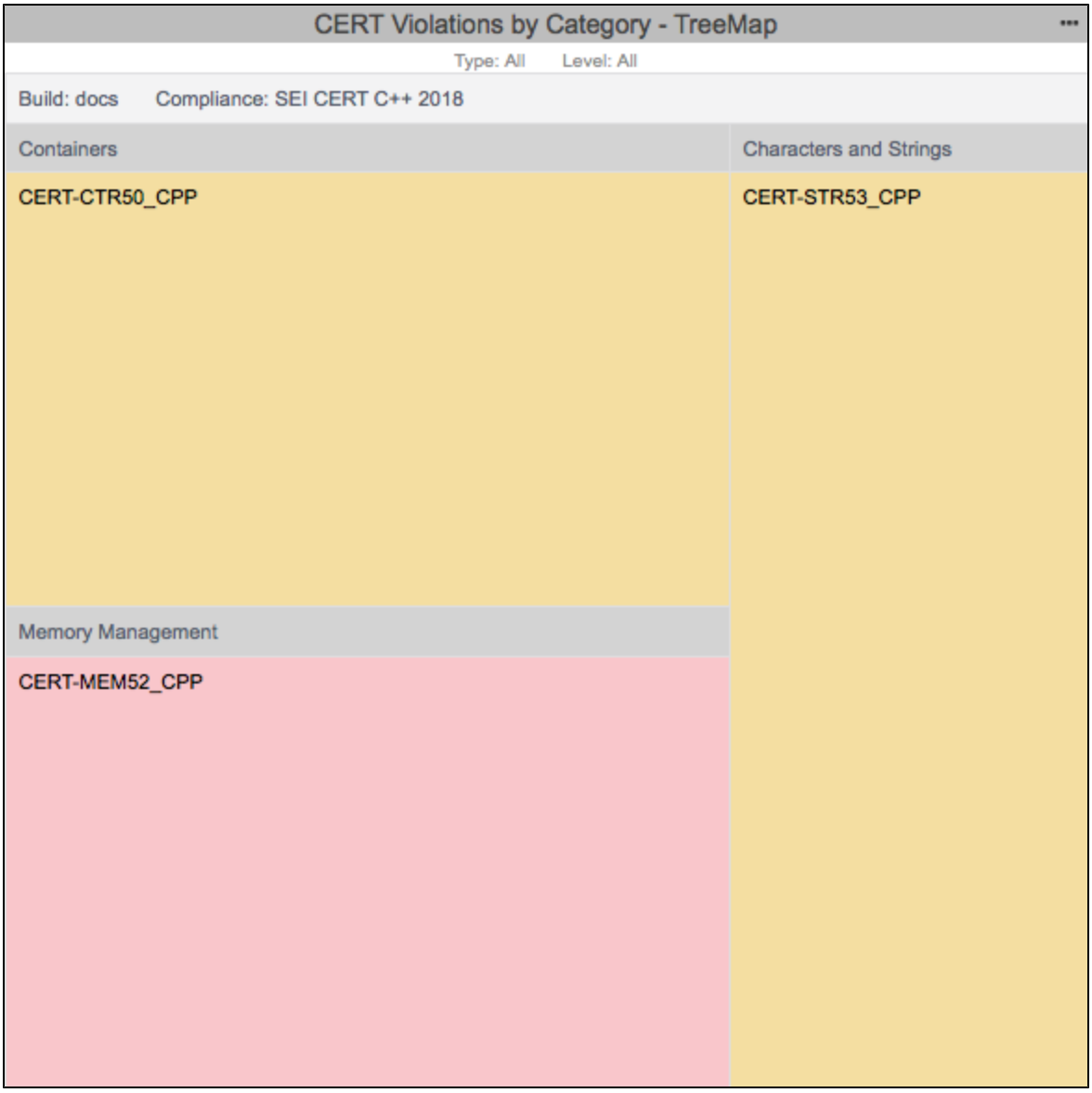
- Mouse over a pie slice to view details.
- Click on a section to open the CERT C++ Compliance Report filtered by the type, priority, and compliance status.
- Click on the number of violations counter to open the CERT C++ Compliance Report filtered by the type, priority, and compliance status.
- Click on the number of deviations counter to open the Deviation Report filtered by the type and priority.

CERT Violations by Category - TreeMap Widget

This widget provides a representation of the highest concentration of static analysis violations per type and priority level. Tiles are color-coded according the priority level:

- red tiles represent L1 violations
- yellow tiles represent L2 violations
- green tiles represent L3 violations

The Parasoft rule(s) enforcing violations are also presented. Tiles are proportional to the number of static analysis violations reported for each rule.

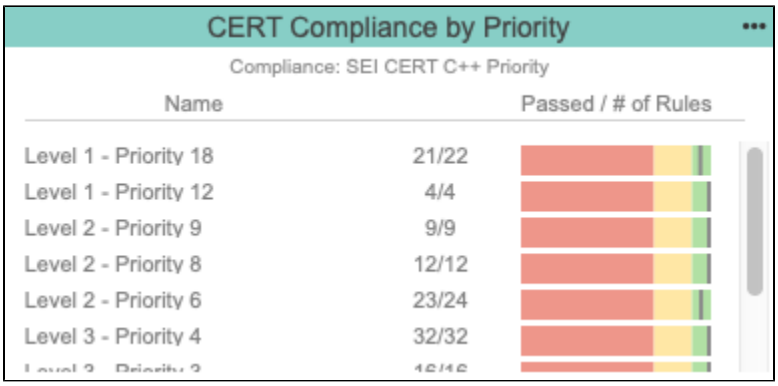


The widget uses the hierarchy established in the model profile to correlate Parasoft rules with CERT rules, recommendations, and priorities. You can mouse over a tile in the widget to view the number of violations associated with each rule-guidline-category.

Click on a rule to see the violation in the Violations Explorer.

CERT Compliance by Priority

This widget is an implementation of the standard Compliance By Category widget shipped with DTP. It shows the number and percentage of rules in compliance grouped by rule categories.



Click on an entry in the table to open the Violations by Compliance Category report.

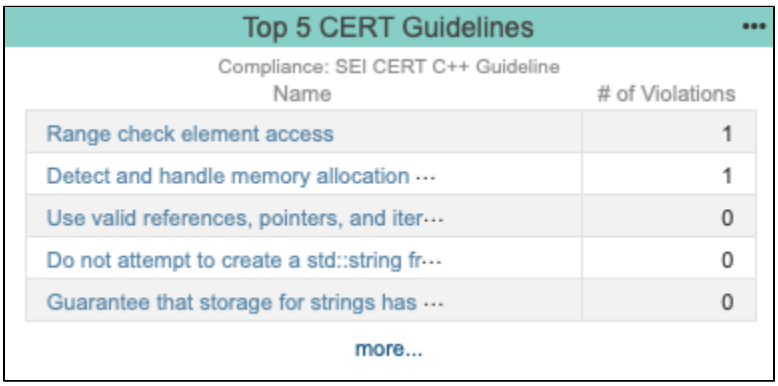
Top 5 CERT Categories

This widget is an implementation of the standard Categories - Top 5 Table widget shipped with DTP. It shows the five CERT guideline categories with the most violations.

Click on a link in the **Name** column or the **more...** link to open the Violations by Compliance Category report.

Top 5 CERT Guidelines

This widget is an implementation of the standard Categories - Top 5 Table widget shipped with DTP. It shows the five CERT guidelines with the most violations.



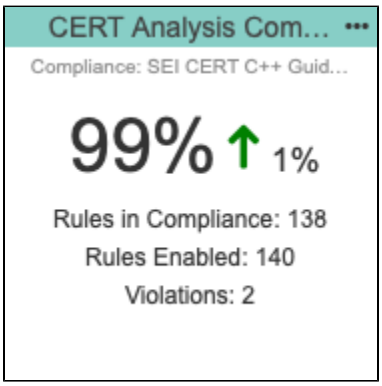
Click on a link in the **Name** column or the **more...** link to open the Violations by Compliance Category report.

CERT Analysis Compliance

This widget is an implementation of the standard Rules in Compliance - Summary widget shipped withe DTP. This widgets shows the following information:

- how many static analysis rules for the selected compliance standard were enabled during code analysis
- how many violations were reported
- the overall percentage of rules that did not report violations

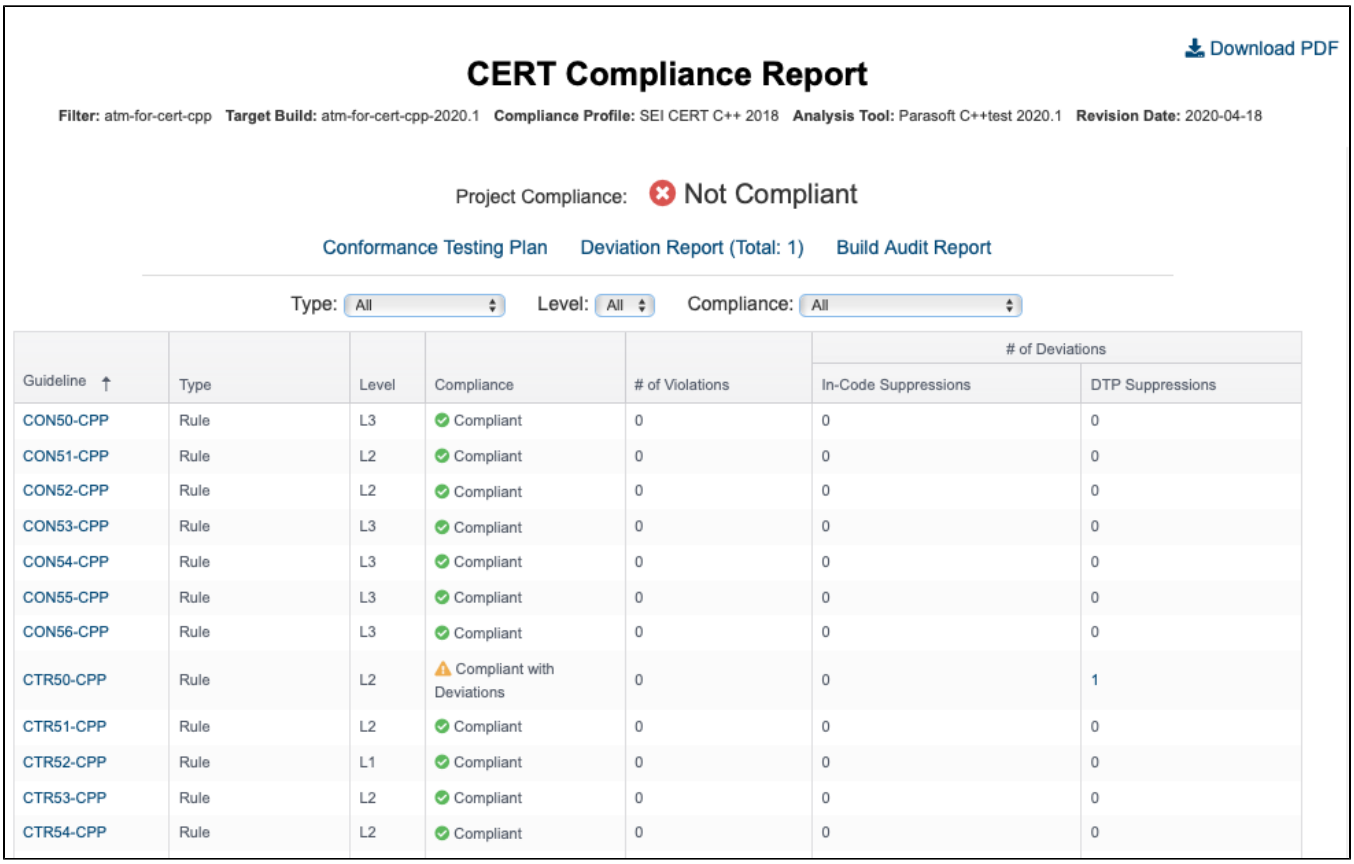
- the change in number of violations from the baseline build to the target build as a percentage (if applicable)



Click on the widget to open the Violations by Compliance Category report.

CERT C++ Compliance Reports

The CERT Compliance Report provides an overview of your CERT compliance status and serves as the primary document for demonstrating compliance.



Guideline	Type	Level	Compliance	# of Violations	# of Deviations	
					In-Code Suppressions	DTP Suppressions
CON50-CPP	Rule	L3	Compliant	0	0	0
CON51-CPP	Rule	L2	Compliant	0	0	0
CON52-CPP	Rule	L2	Compliant	0	0	0
CON53-CPP	Rule	L3	Compliant	0	0	0
CON54-CPP	Rule	L3	Compliant	0	0	0
CON55-CPP	Rule	L3	Compliant	0	0	0
CON56-CPP	Rule	L3	Compliant	0	0	0
CTR50-CPP	Rule	L2	Compliant with Deviations	0	0	1
CTR51-CPP	Rule	L2	Compliant	0	0	0
CTR52-CPP	Rule	L1	Compliant	0	0	0
CTR53-CPP	Rule	L2	Compliant	0	0	0
CTR54-CPP	Rule	L2	Compliant	0	0	0

You can perform the following actions:

- Use the drop-down menus to sort by the following criteria:
  - Guideline type: Rule, Recommendation, or All
  - Priority level: L1, L2, L3, or All
  - Compliance status: All, No Rules Enabled, Compliant, Compliant With Deviations, Compliant With Violations, Not Compliant, Missing Rule(s) in Analysis
- Click on a link in the # of Violations, In-Code Suppression, or DTP Suppressions column to view the violations in the Violations Explorer.
- Open one of the CERT Compliance sub-reports.
- Click **Download PDF** to download a printer-friendly PDF version of the report data. If you added a custom graphic to DTP as described in Adding a Custom Graphic to the Navigation Bar, the PDF will also be branded with the graphic.

The CERT Compliance Report contains four supporting reports:

Conformance Testing Plan

The Conformance Testing Plan cross-references CERT guidelines with Parasoft static analysis rules using the data specified in the compliance profile. You can change the severity, likelihood, remediation cost, and other values to meet your project goals by configuring the profile. Click on a guideline to view the CERT documentation on the CERT website.

CERT Conformance Testing Plan

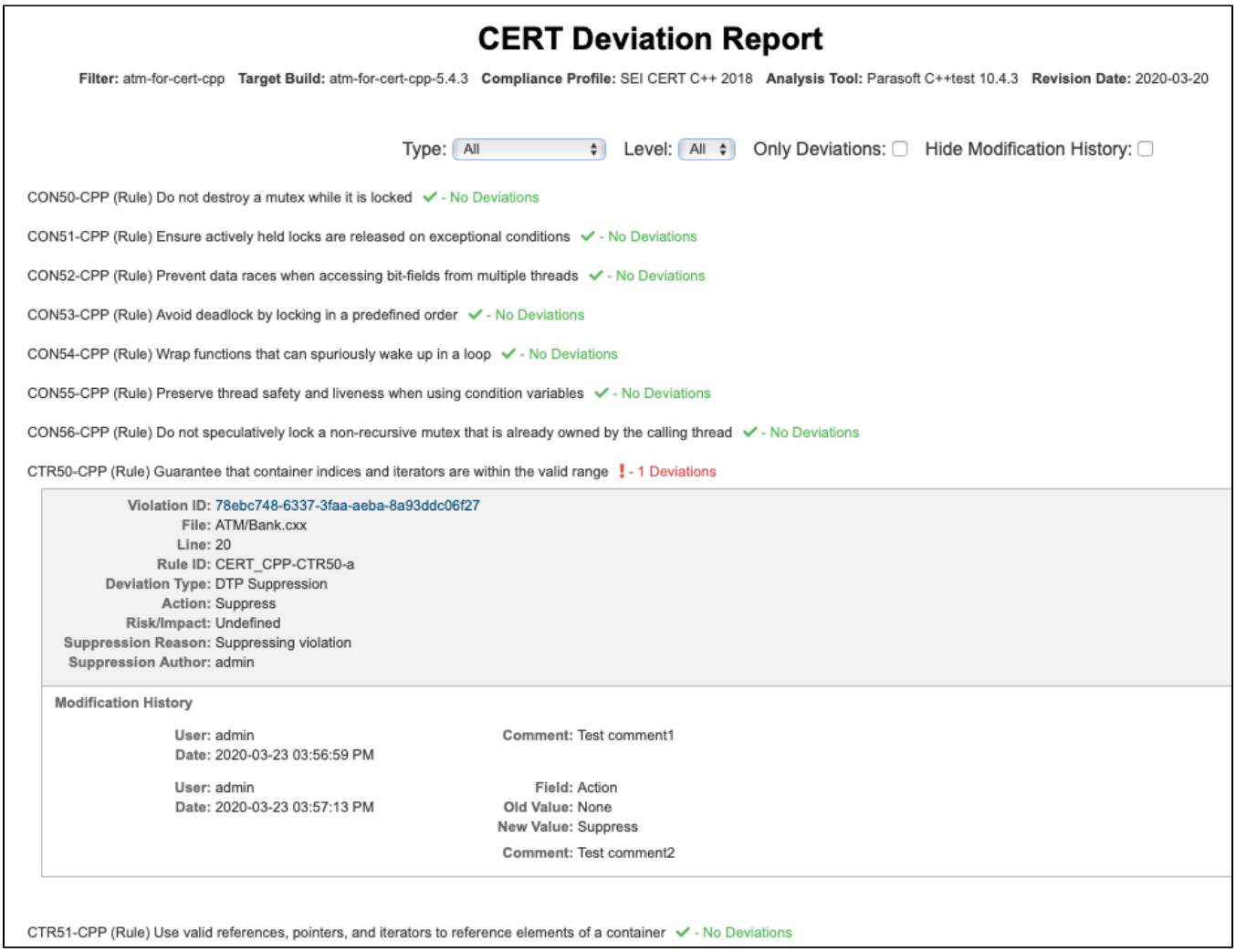
Compliance Profile: SEI CERT C++ 2018 **Analysis Tool:** Parasoft C++test 2020.1 **Revision Date:** 2020-04-18

Guideline	Type	Description	Category	Severity	Likelihood	Remediation Cost	Priority	Level	Parasoft Rule Ids
CON50-CPP	Rule	Do not destroy a mutex while it is locked	Concurrency	Medium	Probable	High	P4	L3	CERT_CPP-CON50-a
CON51-CPP	Rule	Ensure actively held locks are released on exceptional conditions	Concurrency	Low	Probable	Low	P6	L2	CERT_CPP-CON51-a
CON52-CPP	Rule	Prevent data races when accessing bit-fields from multiple threads	Concurrency	Medium	Probable	Medium	P8	L2	CERT_CPP-CON52-a
CON53-CPP	Rule	Avoid deadlock by locking in a predefined order	Concurrency	Low	Probable	Medium	P4	L3	CERT_CPP-CON53-a
CON54-CPP	Rule	Wrap functions that can spuriously wake up in a loop	Concurrency	Low	Unlikely	Medium	P2	L3	CERT_CPP-CON54-a
CON55-CPP	Rule	Preserve thread safety and liveness when using condition variables	Concurrency	Low	Unlikely	Medium	P2	L3	CERT_CPP-CON55-a
CON56-CPP	Rule	Do not speculatively lock a non-recursive mutex that is already owned by the calling thread	Concurrency	Low	Unlikely	High	P1	L3	CERT_CPP-CON56-a

Deviation Report

Your code can contain violations and still be CERT-compliant as long as the deviations from the standard are documented and that the safety of the software is unaffected. Deviations are code analysis rules that have been suppressed either directly in the code or in the DTP Violations Explorer. See the C/C++test documentation for details on suppressing violations in the code. See Suppressing Violations in the Violations Explorer documentation for information about suppressing violations in DTP.

Click on the **Deviation Report** link in the CERT Compliance Report to open the Deviation Report.

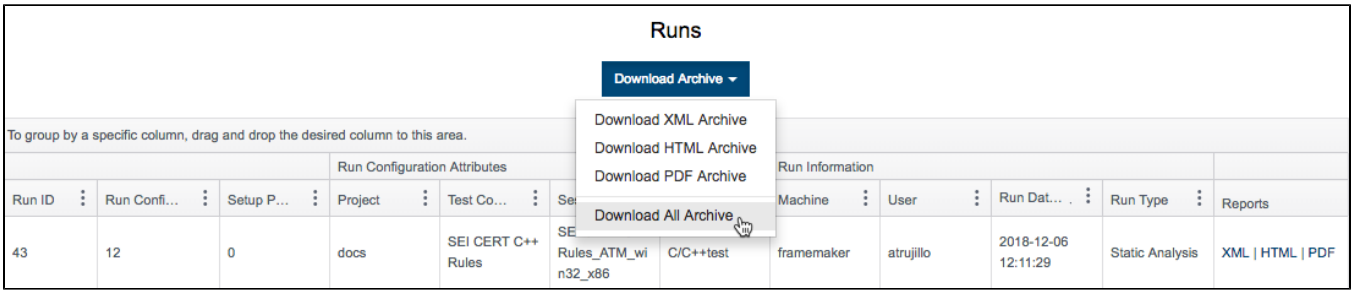


The Deviations Report shows all guideline IDs and headers, but guidelines that have been suppressed will show additional information. You can perform the following actions:

1. Filter the report by type (Rule, Recommendation, All)
2. Filter the report by level (L1, L2, L3)
3. Enable the **Only Deviations** option to only show deviations
4. Enable the **Hide Modification History** option to exclude the modification history for deviations

Build Audit Report

The Build Audit Report is native functionality in DTP. It shows an overview of code analysis violations, as well as test results and coverage information, associated with the build. This report also allows you to download an archive of the data, which is an artifact you can use to demonstrate compliance with CERT during a regulatory audit.



In order to download an archive, the build has to be locked. See Build Audit Report for additional details about this report.

Profiles

The Security Compliance Pack includes a profile associated with the core CERT C++ workflow and a set of profiles associated with calculating the SEI CERT C++ Remediation Cost and SEI CERT C++ Likelihood KPI metrics.

CERT C++ Compliance Profile

The CERT C++ Compliance DTP Workflow ships with a default profile that includes information necessary for generating CERT compliance reports. The default profile shows the correlation between CERT guidelines and Parasoft code analysis rules and is suitable for most normal use cases.

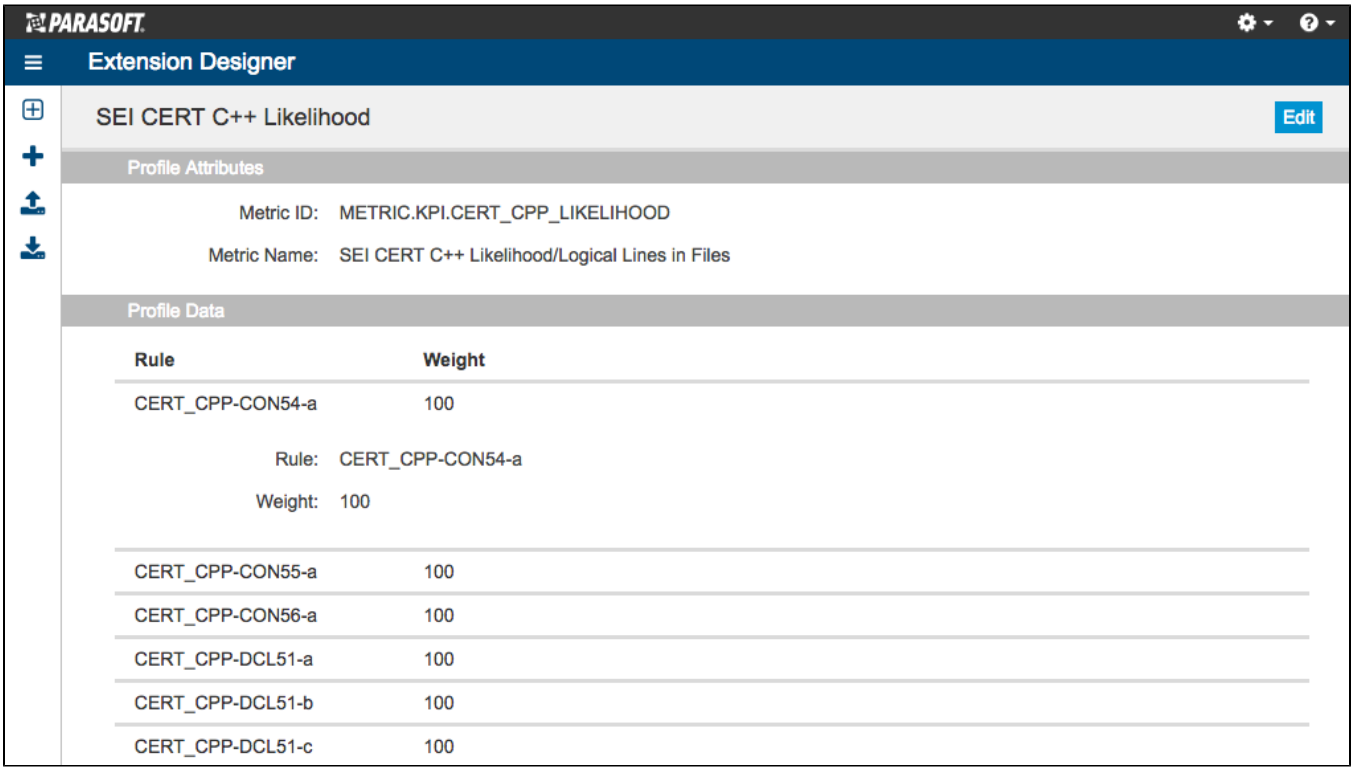
> **Do not modify the CERT profile**
>
> We strongly advise you to avoid changing the default CERT C++ 2018 profile because doing so will affect any reports you may need to generate for auditing purposes.

If necessary, you can make a copy of the default profile and adjust the correlation between Parasoft code analysis rules and CERT C++ guidelines to achieve your software quality and compliance goals.

1. Open Extension Designer and click the **Model Profile** tab.
2. Expand the CERT Compliance model and choose the **SEI CERT C++ 2018** profile.
3. Click **Export Profile** to download a copy.
4. Rename the copy and click **Import Profile**.
5. Browse for the copy and confirm to upload.
6. Rename the profile and click **Edit.**
7. Make your adjustments and click **Save.**

CERT C++ KPI Profiles

The KPI artifact shipped with the Security Compliance Pack includes the SEI CERT C++ Likelihood and SEI CERT C++ Remediation Cost profiles. The profiles assign weights to the metrics analysis rules in order to calculate a KPI value for the build.



The default profile is suitable for most normal usage, but you can adjust the weights for each metrics rule if necessary.

1. Open Extension Designer and click the **Model Profile** tab.
2. Expand the KPI model and choose either the **SEI CERT C++ Likelihood** or **SEI CERT C++ Remediation** profile.
3. Click **Export Profile** to download a copy.
4. Rename the copy and click **Import Profile**.
5. Browse for the copy and confirm to upload.
6. Rename the profile and click **Edit.**
7. Make your adjustments and click **Save.**