Integrating with TeamForge

In this section:

- Introduction
- Requirements
- Configuration
- Usage

Introduction

TeamForge is a popular browser-based platform for managing and tracking requirements, defects, and other work items. Parasoft DTP integrates with TeamForge, providing the following functionality:

- Ability to manually create defects and issues in TeamForge from the Violations Explorer view
- Ability to manually create defects and issues in TeamForge from the Test Explorer view.
- Ability to send, view, and update Parasoft test results in TeamForge.
- Traceability from TeamForge work items to tests, static analysis results, and code reviews (see Viewing the Traceability Report).

Requirements

- Tests executed by C/C++test Professional, dotTEST, Jtest, and SOAtest 10.4.3 and later are supported.
- To send test and analysis data to TeamForge, you should have already created work items in TeamForge. Parasoft can associate multiple tests with the following types TeamForge work items:
  - Story
  - Test

Configuration

The configuration is performed by the Parasoft administrator and only needs to be set up once. Developers, testers, and other DTP end users should review the Usage section for instructions on how to use Parasoft with TeamForge.

Connecting DTP to TeamForge Server

1. Choose **Report Center Settings** from the settings (gear icon) drop-down menu.
2. Choose **External Application** from the Administration sidebar and choose **TeamForge** from the Application Type drop-down menu.
3. Enable the **Enabled** option.
4. Enter a name for your instance of TeamForge in the Name field. The name is required, but it does not affect the connection settings or render in any other interfaces.
5. Enter the TeamForge URL in the Application URL field.
6. The Display URL field is rendered in DTP interfaces when links to your TeamForge system are created.
7. Enter login credentials in the Username and Password/API Tokens fields. The login must have sufficient privileges to create issues in the TeamForge projects specified in the Project Associations section.
8. Click **Test Connection** to verify your settings and click **Save**.

Associating Parasoft Projects with TeamForge Projects

Create links between Parasoft and TeamForge projects so that defects created in the Violations or Test Explorer view are created in the correct project in TeamForge. The association is also important when using the the Sending Test Data to External Application flow.

1. Click **Create Project Association** and choose a project from the DTP Project drop-down menu.
2. Enter the name of a TeamForge project in the External Project field and click **Create** to save the association.

Click the trash icon to remove a project association and deactivate links that may have been created. Removing the project association does not remove links between a DTP explorer view and the work item. If a new association is created, existing links between violations and TeamForge issues will be reactivated.

You can associate multiple projects in DTP with a project in TeamForge, but you cannot associate the same DTP project with more than one TeamForge project.

Enabling the Traceability Report

You can configure DTP to generate widgets and reports that help you demonstrate traceability between the work items stored in TeamForge and the test, static analysis, and build review data sent to DTP from Parasoft tools (C/C++test, dotTEST, Jtest, SOAtest).

If you want the Traceability Report to include code review and static analysis information, you must associate your source code files with requirements stored in TeamForge. See Associating Requirements with Files for instructions.

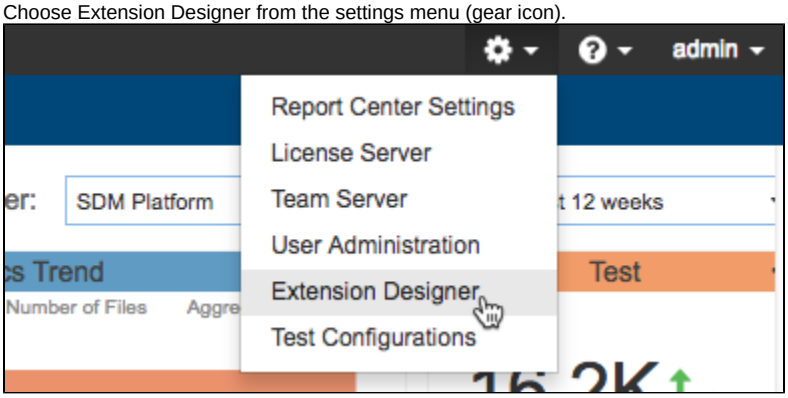
DTP interfaces that display and track traceability are enabled by deploying the External Application Traceability Report artifact shipped with the Traceability Pack. The Traceability Pack also includes the Sending Test Data to External Application flow, which automates part of the traceability workflow. Refer to the Traceability Pack documentation for additional information about the pack.

Use DTP Extension Designer to deploy the External Application Traceability Report and the Sending Test Data to External Application flow to your environment. Verify that DTP is connected to TeamForge as described in the Connecting DTP to TeamForge Server section before deploying the artifact.

Installing the Traceability Pack

The first step is to install the Traceability Pack. The artifact is a collection of configuration files and assets that enable traceability.

1. Choose Extension Designer from the settings menu (gear icon).

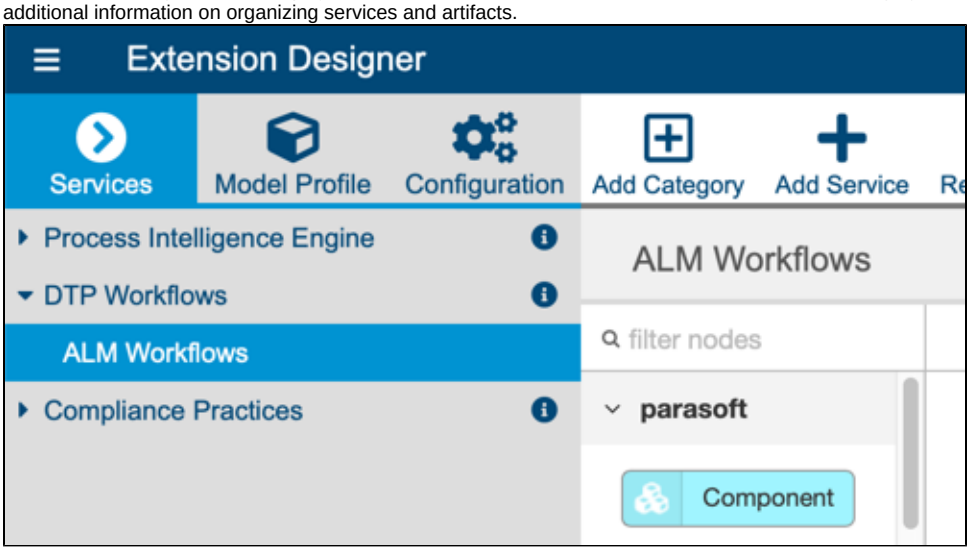


2. Click the **Configuration** tab to open Artifact Manager.
3. Click **Upload Artifact** and browse for the external-app-traceability-report-<version>.zip archive (also see Downloading and Installing Artifacts).
4. Click **Install** and a collection of assets and configuration files for enabling traceability will be installed.

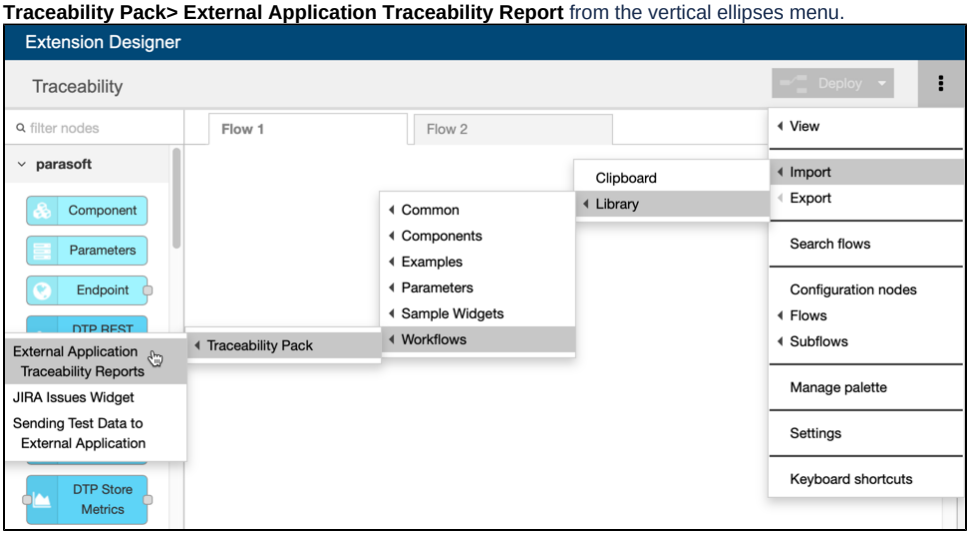
Deploying the External Application Traceability Report

Deploy the External Application Traceability Report after installing the Traceability Pack.

1. Open Extension Designer and click on the **Services** tab.
2. Choose an existing service to deploy the artifact or create a new service in the DTP Workflows category. Refer to Working with Services for additional information on organizing services and artifacts.



3. If you are adding the artifact to an existing service, add a new Flow tab (see Working with Flows) and choose **Import> Library> Workflows> Traceability Pack> External Application Traceability Report** from the vertical ellipses menu.



4. Click inside the Flow tab to drop the nodes into the service and click **Deploy**.

Deploying the External Application Traceability Report adds new widgets to Report Center, as well as a drill-down report. See Viewing the Traceability Report for instructions on adding the widgets and viewing the report.

Deploying the Sending Test Data to External Application Flow

This artifact sends test data to TeamForge when DTP Data Collector retrieves test results from a Parasoft tool. This artifact ships with the Traceability Pack, which must be installed as described in Installing the Traceability Pack before deploying the flow.

1. Open Extension Designer and click on the **Services** tab.
2. Choose an existing service to deploy the artifact or create a new service in the DTP Workflows category. Refer to Working with Services for additional information on organizing services and artifacts.
3. If you are adding the artifact to an existing service, add a new Flow tab (see Working with Flows) and choose **Import> Library> Workflows> Traceability Pack> Sending Test Data to External Application** from the vertical ellipses menu.
4. Click inside the Flow tab to drop the nodes into the service and click **Deploy**.

Advanced Configuration

You can modify the ExternalAppsSettings.properties configuration file located in the <DTP_DATA_DIR>/conf directory to change the default behavior of the TeamForge integration. The out-of-the-box configuration uses default or commonly-used fields and work item types. If you customized your TeamForge system, however, then you can configure the following settings to align data in DTP with your custom configuration.

teamForge.defect.status	Specifies the status of defects that are created in TeamForge when creating work items in the DTP Violations Explorer and Test Explorer views. Default: `Open`
teamForge.tasks.status	Specifies the status of tasks that are created in TeamForge when creating work items in the DTP Violations Explorer and Test Explorer views. Default: `Not Started`
teamForge.workItemType.defect	Specifies the type of work item to create in TeamForge when creating new defects from the DTP Violation Explorer and Test Explorer. This enables you to associate custom defect trackers you may have configured in TeamForge with work items created from DTP. By default, the property is not set. As a result, defect work items created in DTP are associated with task work items in TeamForge.
teamForge.workItemType.task	Specifies the type of work item to create in TeamForge when creating new tasks from the DTP Violation Explorer and Test Explorer. This enables you to associate custom task trackers you may have configured in TeamForge with work items created from DTP. By default, the property is not set. As a result, task work items created in DTP are associated with task work items in TeamForge.
teamforgeIssueUrl	Specifies the URL template for linking work items created in the DTP Violation Explorer and Test Explorer to work items in TeamForge. Default: `teamforgeIssueUrl=<TEAMFORGE_URL>/sf/go<ID>`

teamForge.trackerType.requirement.name	Specifies the name of TeamForge work item types that should take the role of requirements in Parasoft. The work items are also used in the Traceability Report. Default: `Stories`
teamForge.trackerType.test.name	Specifies the name of TeamForge work item types that should take the role of tests in Parasoft. The work items are also used in the Traceability Report. Default: `Tests`
teamForge.trackerType.test.status.pass	Specifies the test run status name in TeamForge to assign to passing test results sent from DTP. This enables you to set custom statuses you may have configured in TeamForge for test results in DTP. Default: `Passed`
teamForge.trackerType.test.status.fail	Specifies the test run status name in TeamForge to assign to passing test results sent from DTP. This enables you to set custom statuses you may have configured in TeamForge for test results in DTP. Default: `Failed`

Usage

After configuring the integration with TeamForge, developers, testers, and other users can leverage the functionality enabled by the integration.

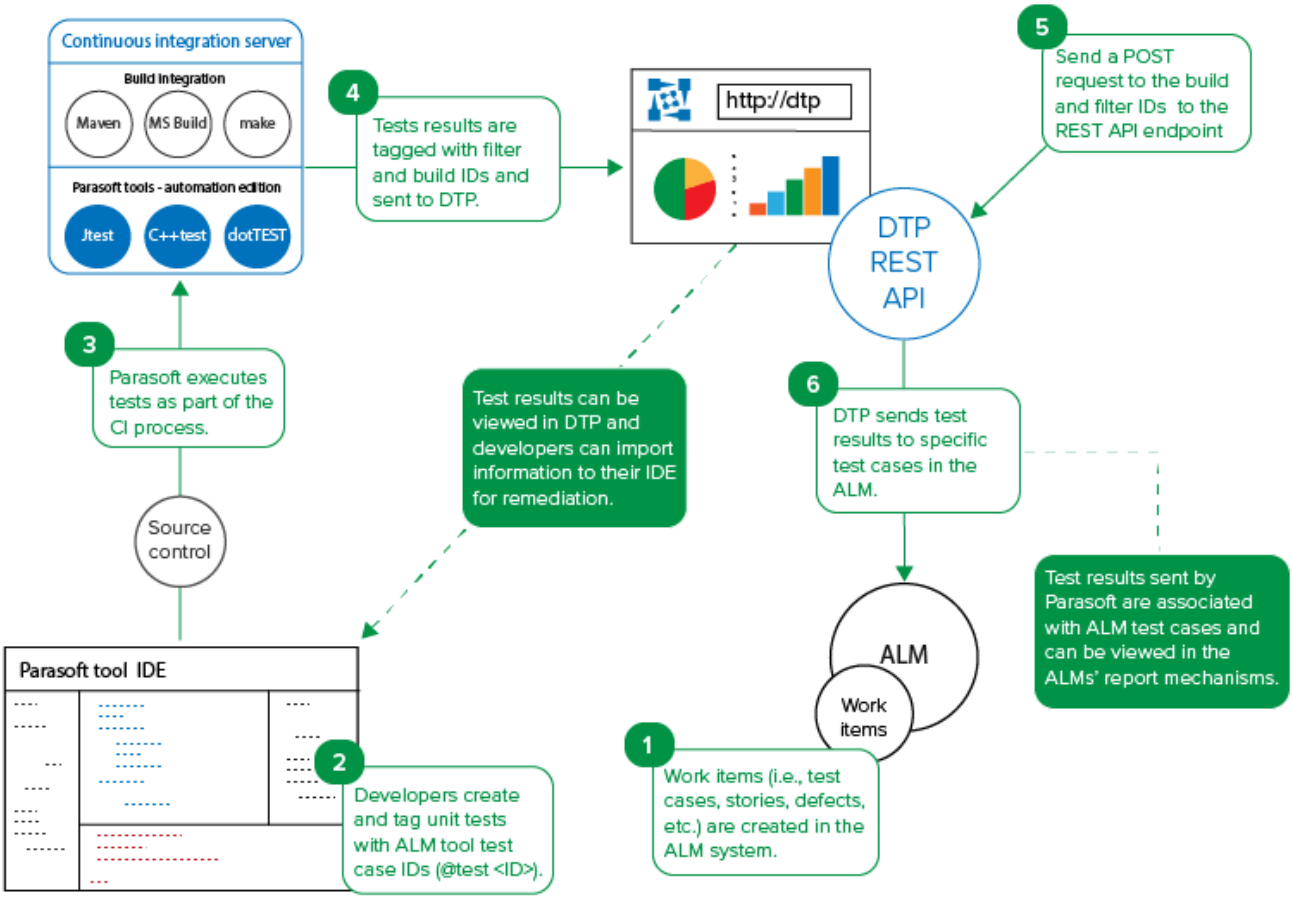
Manually Creating Defects and Tasks in TeamForge

The Test Explorer and Violations Explorer views enable you to create defects and tasks for any test and violation, respectively, regardless of status. Refer to the following sections for details on creating TeamForge assets in explorer views:

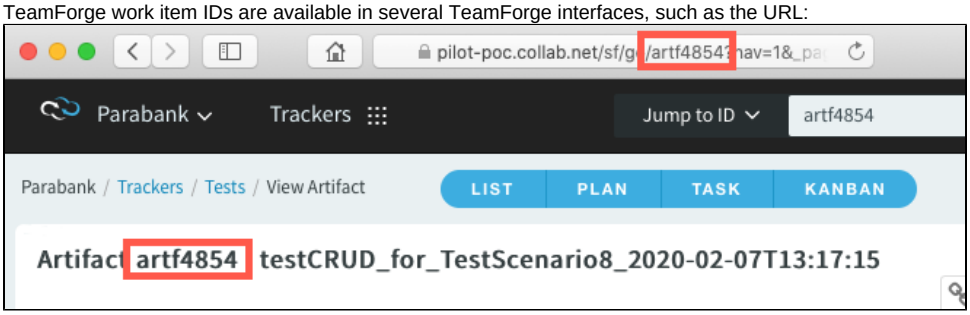
- See Creating an Issue in a Third-party System for instructions on how to manually create defects and tasks in TeamForge from the Violations Explorer view.
- See Creating an Issue in a Third-party System for instructions on how to manually create defects and tasks in TeamForge from the Test Explorer view.

Sending Test Data to TeamForge

TeamForge work items are assets that represent a story, defect, or set of tests. Annotate the test code executed by your Parasoft tool with the TeamForge work item ID using `@test` or `@req` annotation. By default, the `@test` annotation will associate Parasoft tests with TeamForge tests and the `@req` annotation will associate Parasoft tests with TeamForge stories (see Advanced Configuration for instructions on how to change these settings). Refer to your Parasoft tool documentation for details on adding associations and the TeamForge documentation for information about getting work item IDs. The following diagram shows how you could implement an automated infrastructure for integrating Parasoft into your TeamForge environment:



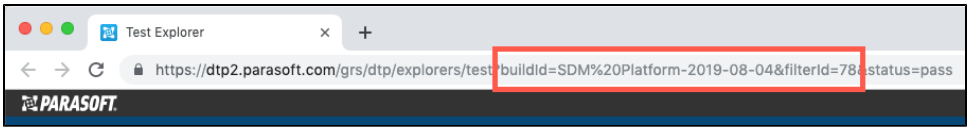
1. Create work items in TeamForge to associate with tests executed by Parasoft tools.
2. In your test file, add the TeamForge work item IDs using the `@test` or `@req` annotation:
   - Use the `@test <TeamForge test ID>` annotation in your tests to associate them with TeamForge tests.
   - Use the `@req <TeamForge story ID>` annotation in your tests to associate them with TeamForge stories.
   - TeamForge work item IDs are available in several TeamForge interfaces, such as the URL:



3. Execute your tests as part of the CI process. You can also manually execute the tests from your tool's desktop.
4. As part of the test execution, the tool will tag the results with the filter and build IDs and send the data to DTP. You can verify the results in DTP by adding Test Widgets to your DTP dashboard and setting the filter and build ID. Developers can download the test execution data from DTP into their IDEs so that they can address any failed tests.
5. If you deployed the Sending Test Data to External Application flow (see Deploying the Sending Test Data to External Application Flow), then unit and functional testing results will automatically be sent to TeamForge when Data Collector receives the data from the Parasoft tool.
   You can also manually send a POST request to the DTP REST API endpoint to send results from the DTP database to TeamForge. Pass the DTP filter and build IDs as URL parameters in the API call:

```
curl -X POST -u <username>:<password> "http://<host>:<port>/grs/api/v1.7/linkedApps/configurations/1
/syncTestCases?filterId=<filterID>&buildId=<buildID>"
```

   The filter and build IDs are available in the Test Explorer URL:



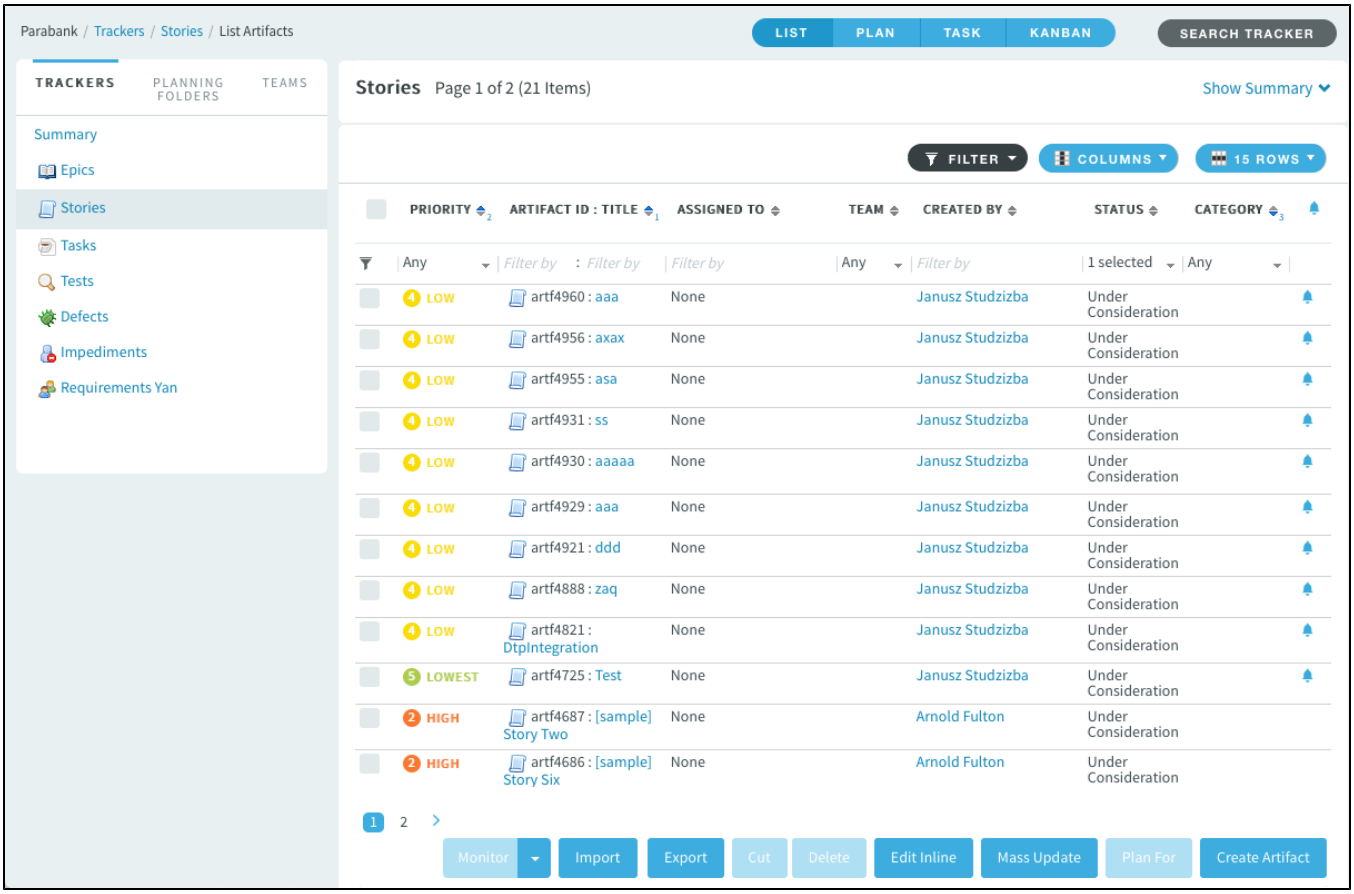
6. DTP will locate the test results that match the `filterId` and `buildId` parameters and send the data to the TeamForge work items. You should expect the following response:
    - When DTP locates results with an `@test <ID>`, it will search for unit test cases with a matching ID in TeamForge and update the item. No action will be taken if the unit test case IDs do not exist in TeamForge.
    - When DTP locates results with an `@req <ID>`, it will search for work items with a matching ID in TeamForge and update associated children unit test cases. If no unit test cases exist for the requirement IDs, unit test cases will be created. Unit test cases will also be created if the requirement IDs are not found.
    - An external-app-sync.log file will also be written to the the <DTP_INSTALL>/logs directory. This log file contains progress information about sending test results from DTP to TeamForge.

After DTP processes the report and sends results to TeamForge, you should expect a response similar to the following:

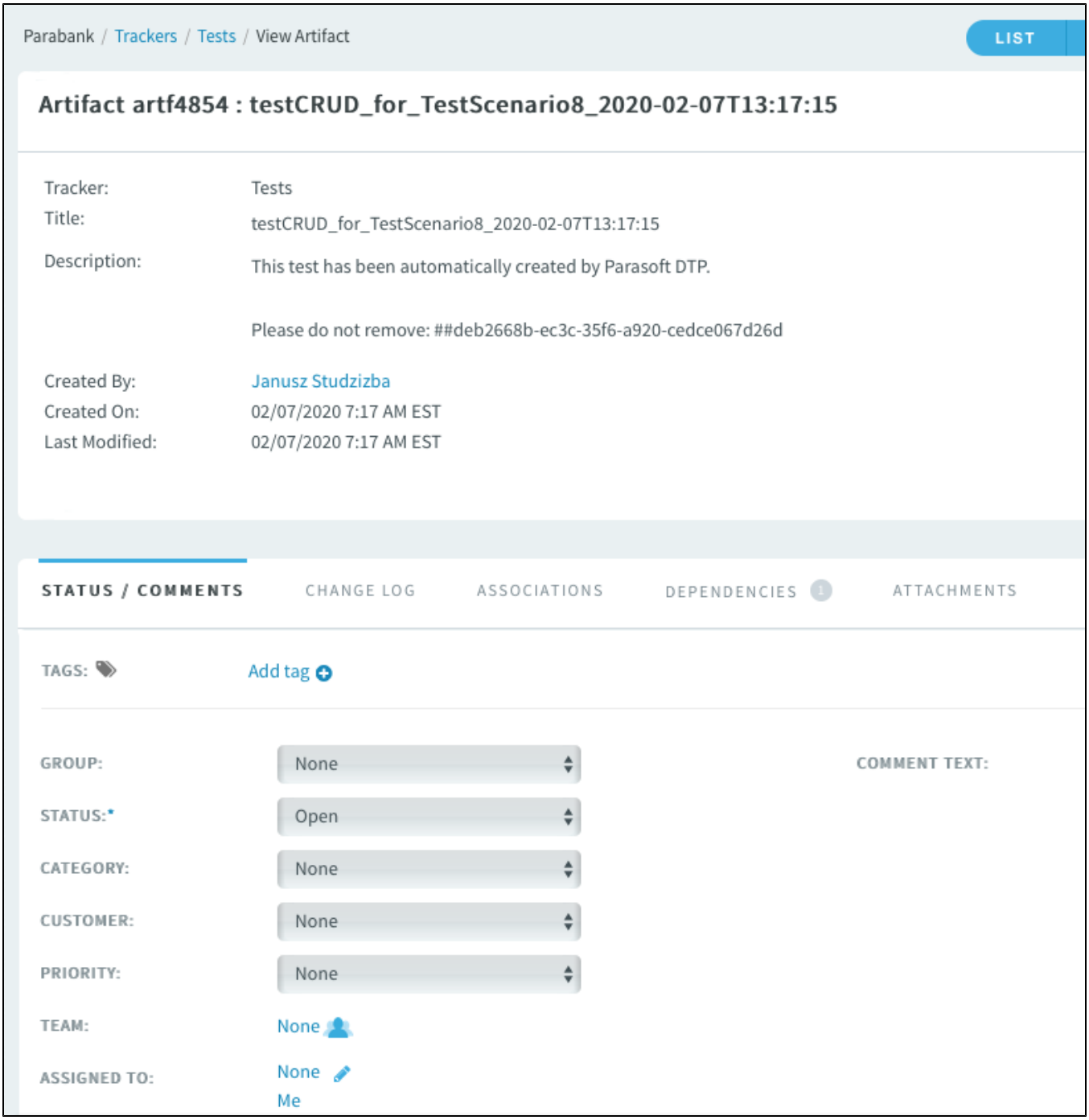
```
{
        "createdTestSession":"DTPP-521",
        "created" : [
                "DTPP-519, testName = testBagSumAdd"
    ],
    "updated" : [
        "Test:1545 for AT-01053, testName = test_quoteGhsLine_Exp_Act_3",
        "Test:1546 for AT-01054, testName = test_quoteGhsLine",
        "Test:1554 for AT-01056, testName = test_quoteGhsLine_Exp_Act_10",
        "Test:7177 for S-01045, testName = test_quoteGhsLine_moreThanOne"
    ],
    "ignored" : [
                "MAGD-567, testName = testBagNegate",
        "QAP-512, testName = testTryThis3",
        "QAP-512, testName = testTryThis4",
        "MAGD-567, testName = testBagMultiply"
    ]
}
```

Viewing Results in TeamForge

After successfully sending the test data to from DTP, you will be able to view results TeamForge assets



You can drill down into results to see details about the work item in DTP, which includes build and authorship information, as well as links back to the test or violation in DTP.



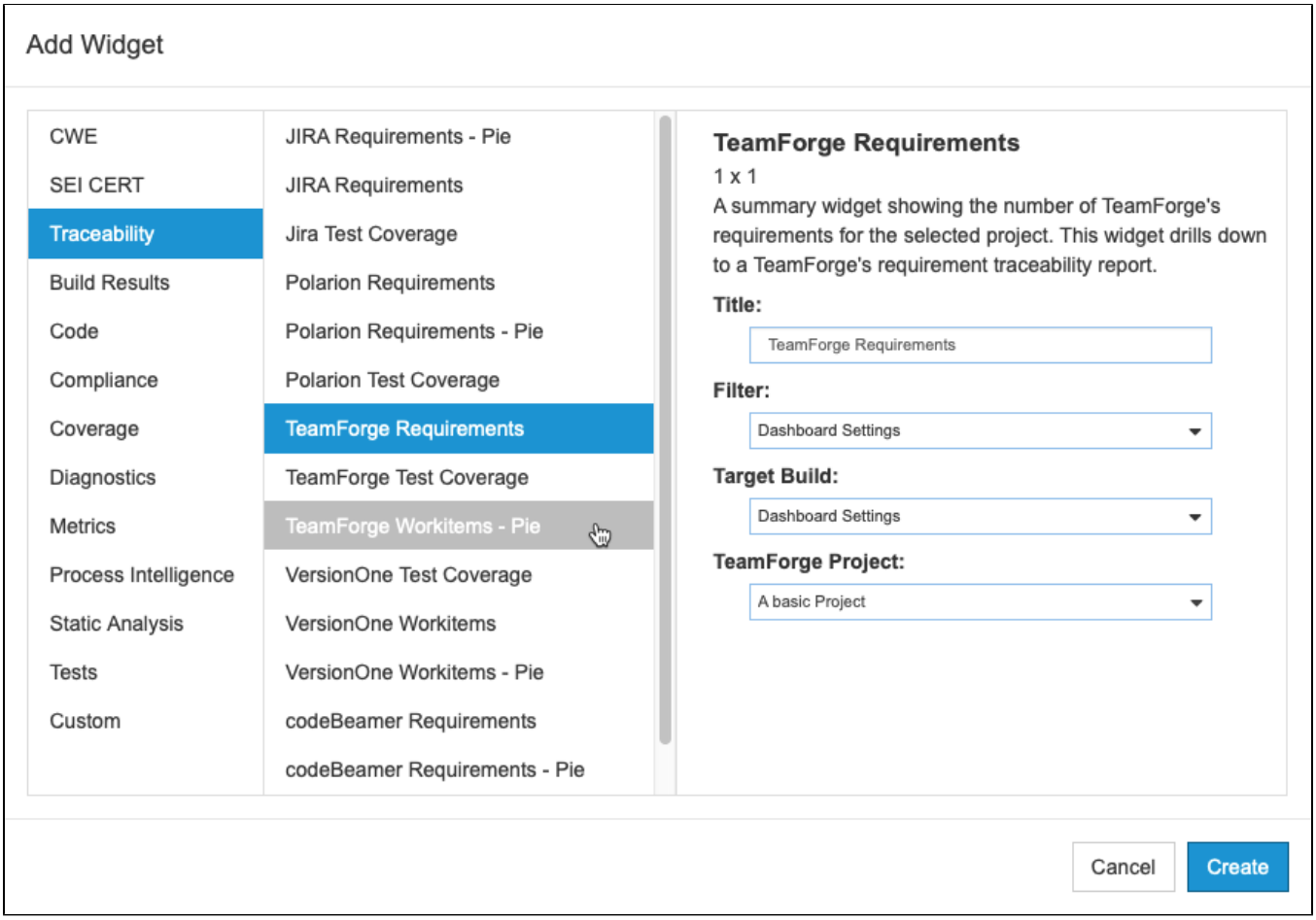
Refer to the TeamForge documentation for details on using TeamForge interfaces.

Viewing the Traceability Report

If the External Application Traceability Report has been deployed to your system (see Enabling the Traceability Report), you can add widgets to your dashboard to monitor traceability from work items to tests, static analysis, code reviews for your project. The widgets also drill down to a report that includes additional details.

Adding and Configuring the Widgets

The widgets will appear in a separate Traceability category when adding widgets to your DTP dashboard. See Adding Widgets for general instructions on adding widgets.



You can configure the following settings:

Title	You can enter a new title to replace the default title that appears on the dashboard.
Filter	Choose Dashboard Settings to use the dashboard filter or choose a filter from the drop-down menu. See Creating and Managing Filters for additional information about filters.
Target Build	This should be set to the build ID you executed the tests and code analysis under. You can use the build specified in the dashboard settings, the latest build, or a specific build from the drop-down menu. Also see Configuring Dashboard Settings.
Type	*Pie widget only*. Choose either a Tests, Violations, or Reviews from the drop-down menu to show a pie chart detailing the status by type. Add instances of the widget configured to each type for a complete overview in your dashboard.
Project	Choose a TeamForge project from the drop-down menu.

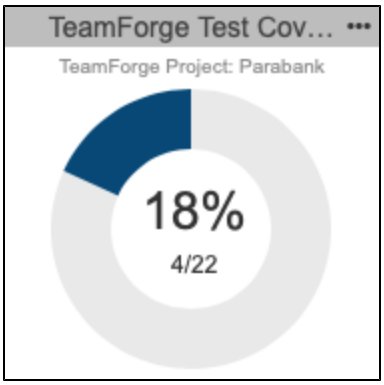
Requirements Widget

This widget shows the number of work items from the specified TeamForge project.

Click on the widget to open the Requirement Traceability report.

Test Coverage Widget

This widget shows the percentage of requirements covered by tests against all requirements in the project.



Click the center of the widget to open the main Requirement Traceability report.

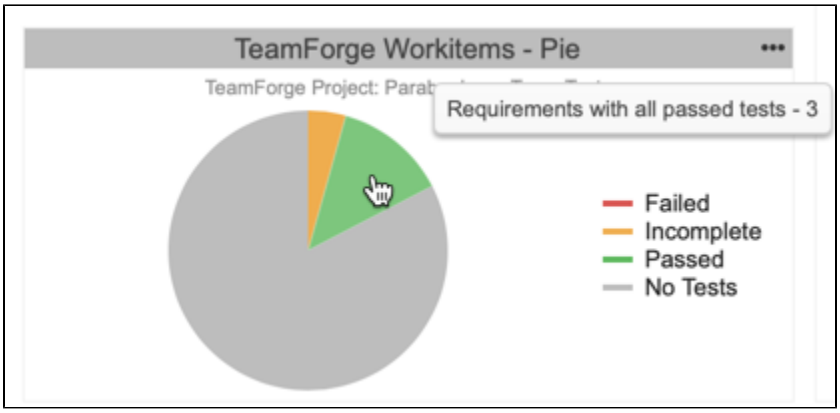
The colored-in segment represents the requirements covered by tests. Click on the segment to open the Requirement Traceability report filtered to the **With Tests** category.

Pie Widget

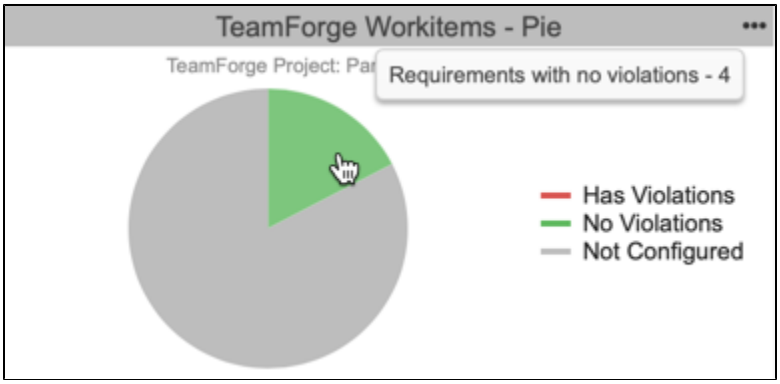
Unit testing, functional testing, static analysis, and peer reviews are common activities for verifying that work items have been properly and thoroughly implemented. This widget shows the overall status of the project work items in the context of those software quality activities. You can add a widget for each type of quality activity (tests, static analysis violations, reviews) to monitor the progress of work item implementation for the project.

Mouse over a section of the chart to view details about quality activity type status. Click on the widget to open the Requirement Traceability report filtered by the selected type.

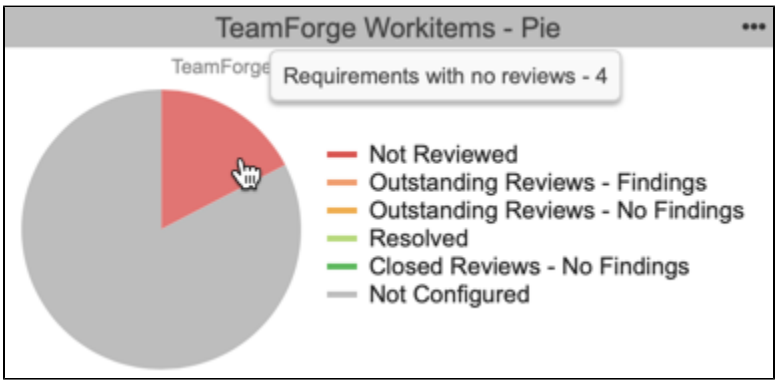
Requirements Implementation Status by Tests



Requirements Implementation Status by Violations

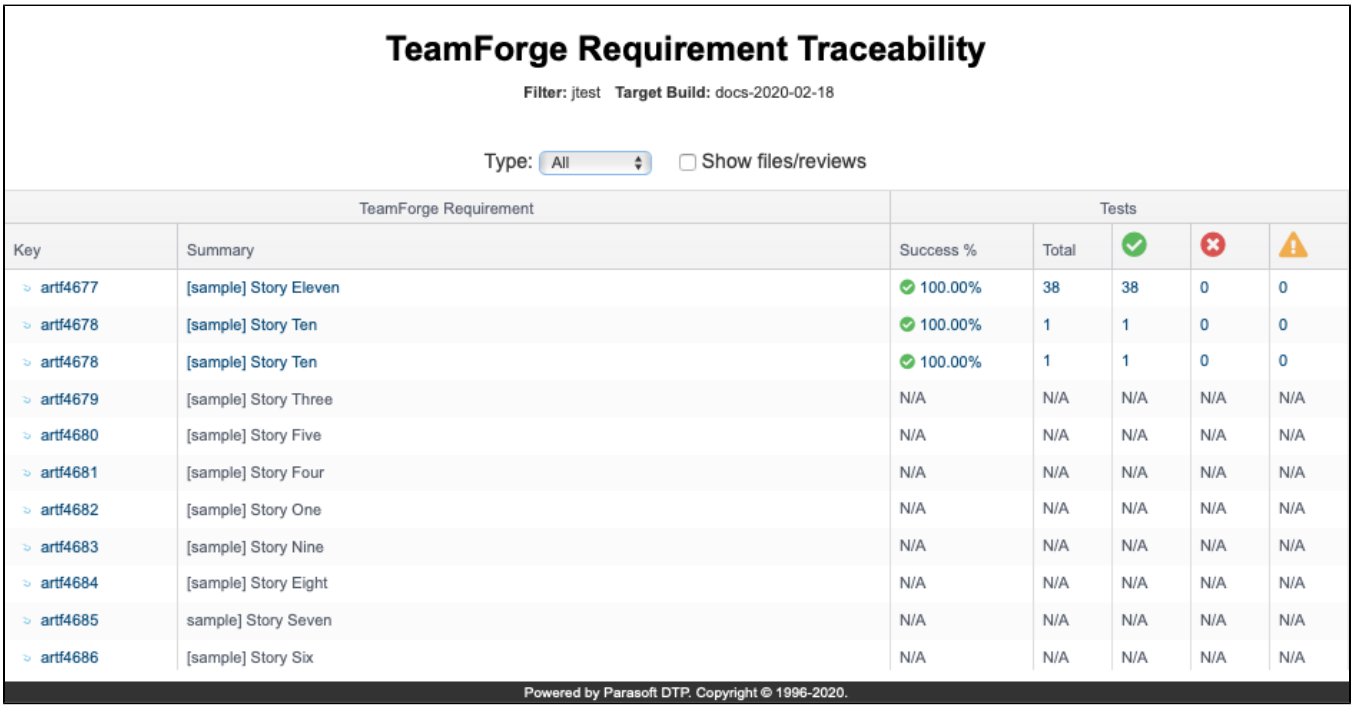


Requirements Implementation by Reviews



Understanding the Requirement Traceability

The report lists the TeamForge project work items and data associated with them.



TeamForge Requirement Traceability

Filter: jtest **Target Build:** docs-2020-02-18

Type: All Show files/reviews

TeamForge Requirement		Tests				
Key	Summary	Success %	Total	✓	✗	⚠
artf4677	[sample] Story Eleven	100.00%	38	38	0	0
artf4678	[sample] Story Ten	100.00%	1	1	0	0
artf4678	[sample] Story Ten	100.00%	1	1	0	0
artf4679	[sample] Story Three	N/A	N/A	N/A	N/A	N/A
artf4680	[sample] Story Five	N/A	N/A	N/A	N/A	N/A
artf4681	[sample] Story Four	N/A	N/A	N/A	N/A	N/A
artf4682	[sample] Story One	N/A	N/A	N/A	N/A	N/A
artf4683	[sample] Story Nine	N/A	N/A	N/A	N/A	N/A
artf4684	[sample] Story Eight	N/A	N/A	N/A	N/A	N/A
artf4685	sample] Story Seven	N/A	N/A	N/A	N/A	N/A
artf4686	[sample] Story Six	N/A	N/A	N/A	N/A	N/A

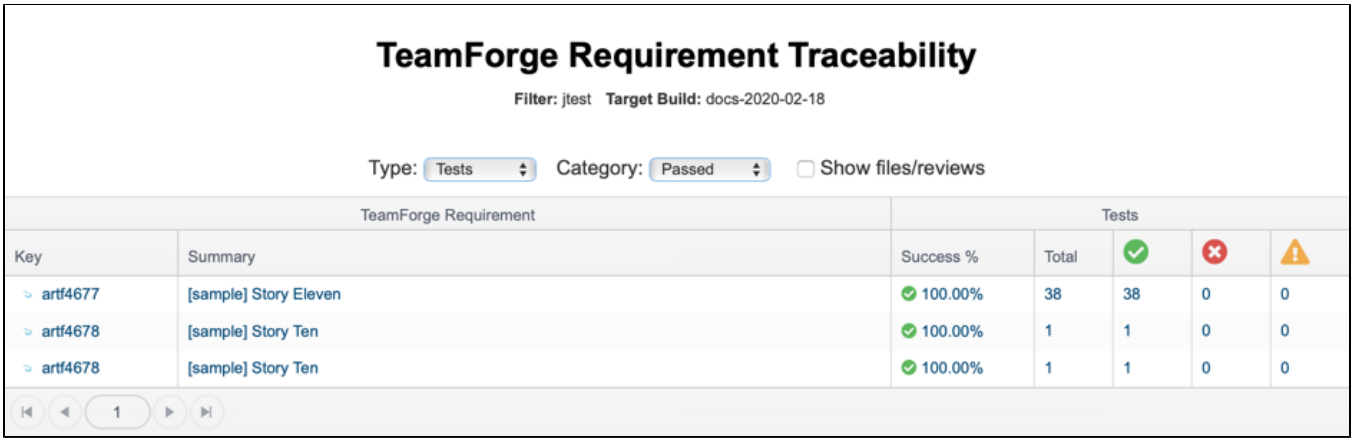
Powered by Parasoft DTP. Copyright © 1996-2020.

You can perform the following actions:

- Disable or enable the **Show files/reviews** option if you want to hide the Files and Reviews columns in the report. The Files and Reviews columns will only contain data if the requirements have been mapped to source files files (see Enabling the Requirements Traceability Report). Disabling the Files and Reviews columns on this screen hides the related tabs in the Requirement Details report.
- Click on a link in the Key column to view the work item in TeamForge.
- Click on a link in the Summary column or one of the Test columns to view the test-related information associated with the work item in the Requirement Details Report.
- Click on a link in one of the Files columns to view the static analysis-related information associated with the work item in the Requirement Details Report.
- Click on a link in one of the Reviews columns to view the change review-related information associated with the work item in the Requirement Details Report.

Requirement Traceability Report by Type

Clicking on a section of the Pie widget opens a version of the report that includes only the quality activity type selected in the widget. You can use the drop-down menus to switch type and status.

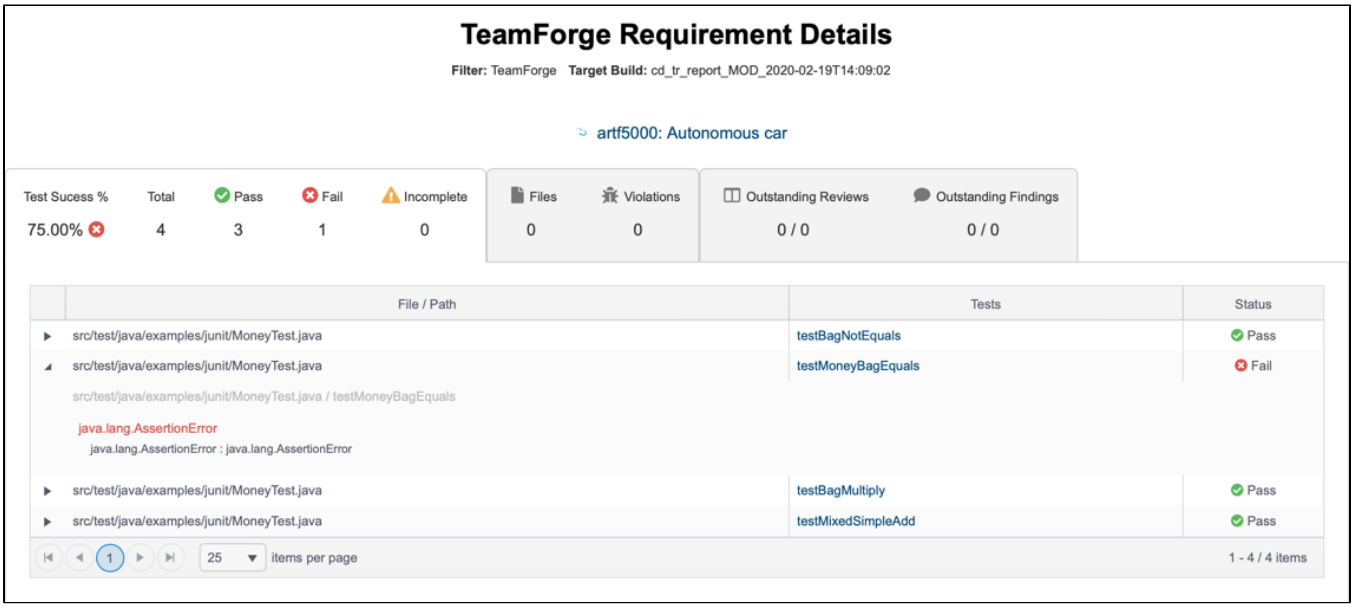


Understanding the Requirement Details Report

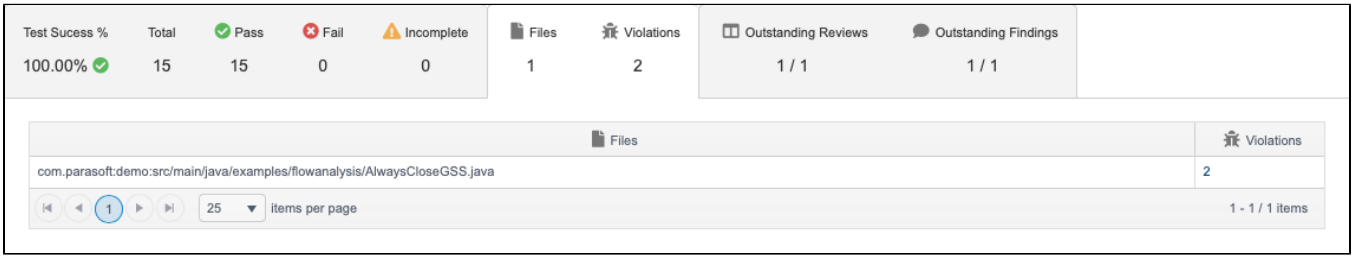
The Requirement Details report provides additional information about the files, static analysis findings, and tests associated with a specific TeamForge work item. Stories marked as `Dead` will not appear in the Traceability Report. The Traceability Report shows information about the following TeamForge work items:

- Requirements
- Defects

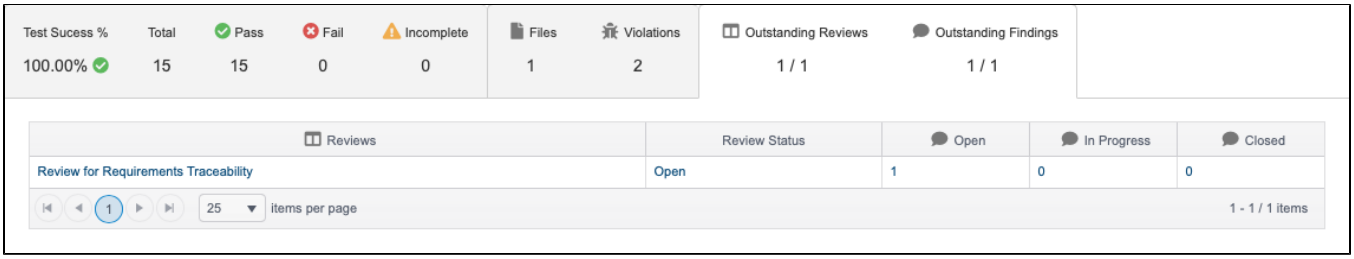
You can open this report by clicking on a work item in the main Requirement Traceability report.



The first tab shows the results of the tests that were executed to verify the specific work item. Click on a test name to view the test in the Test Explorer.



The second tab shows the files associated with the specific requirement, as well as the static analysis violations detected in the files. You can click the link the Violations column to view the violations in the Violations Explorer, which provides additional details about the violations.



If the files include any change reviews or review findings, they will be shown in the third tab with links to view them in the Change Explorer.