

Integrating with CodeBeamer ALM

In this section:

- [Introduction](#)
- [Requirements](#)
- [Configuration](#)
- [Usage](#)

Introduction

CodeBeamer ALM is a popular browser-based platform for managing requirements. Parasoft DTP integrates with codeBeamer ALM, providing the following functionality:

- Ability to manually create bugs and tasks in codeBeamer ALM from the [Violations Explorer](#) view.
- Ability to manually create bugs and tasks in codeBeamer ALM from the [Test Explorer](#) view.
- Ability to send, view, and update Parasoft test results in codeBeamer work items ([Sending Test Data to CodeBeamer](#)).
- Traceability from codeBeamer requirements to tests, static analysis results, and code reviews (see [Viewing the Traceability Report](#)).

Requirements

The following requirements are only applicable if you are going to send test results to codeBeamer.

- Tests executed by the following Parasoft tools:
 - C/C++test Professional, dotTEST, or Jtest 10.4.3 +
 - Selenic 2020.1 +
 - SOAtest 9.10.8 +
- You should have already created requirements in codeBeamer.

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 codeBeamer ALM.

Connecting DTP to CodeBeamer ALM Server

1. Choose **Report Center Settings** from the settings (gear icon) drop-down menu.
2. Choose **External Application** and choose codeBeamer from the Application Type drop-down menu.
3. Enable the **Enabled** option.
4. Enter a name for your instance of codeBeamer ALM in the Name field. The name is required but does not affect the connection settings or render in any other interfaces.
5. Enter the URL for the codeBeamer server in the Application URL field. The URL should include the protocol, host, and port number. Do not include paths or parameters.
6. The Display URL field is rendered in DTP interfaces when links to your codeBeamer system are created. This URL should include additional paths that may be necessary to access codeBeamer in a browser.
7. Enter login credentials in the Username and Password/API Tokens fields. The login must have sufficient privileges to create issues in the codeBeamer projects specified in the Project Associations section.
8. Click **Test Connection** to verify your settings and click **Save**.

Associating Parasoft Projects with CodeBeamer ALM Projects

Creating links between Parasoft with codeBeamer at the project level enables defects created in the Violations or Test Explorer view to be created in the correct project in codeBeamer.

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

Click the trash icon to remove a project association. Deleting the project association does not remove links in DTP explorer views to defects in codeBeamer. If an association is deleted and recreated later, existing links between violations and codeBeamer issues will be reactivated.

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

Enabling the Requirements Traceability Report

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

Your source code files must be associated with the requirements stored in codeBeamer. See [Associating Requirements with Files](#) for instructions.

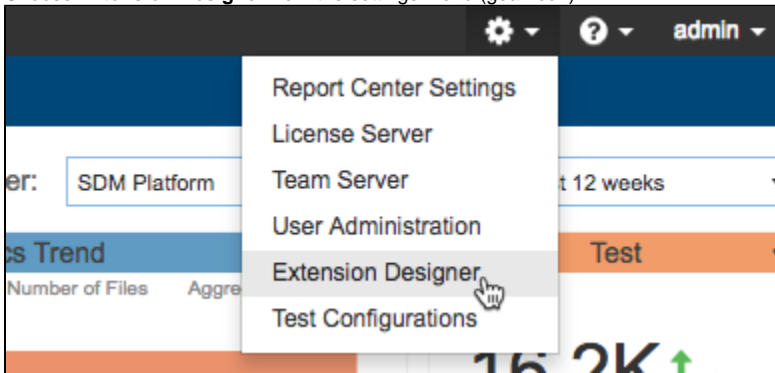
DTP interfaces that display and track traceability are enabled by deploying the External Application Traceability Report artifact shipped with the Traceability Pack. The Traceability Pack also includes the Sending Test Data to External Application flow, which automates part of the requirements traceability workflow. Refer to the [Traceability Pack](#) documentation for instructions on how to install 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 codeBeamer as described in the [Connecting DTP to CodeBeamer ALM Server](#) section before deploying the artifacts.

Installing the Traceability Pack

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

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

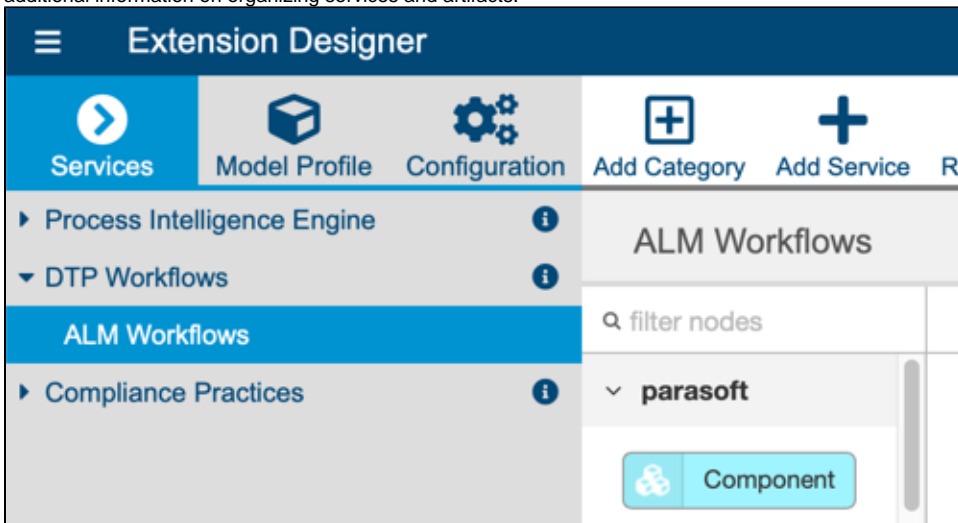


2. Click the **Configuration** tab to open Artifact Manager.
3. Click **Upload Artifact** and browse for the traceability-pack-<version>.zip archive (also see [Downloading and Installing Artifacts](#)).
4. Click **Install** and a collection of assets and configuration files for enabling traceability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

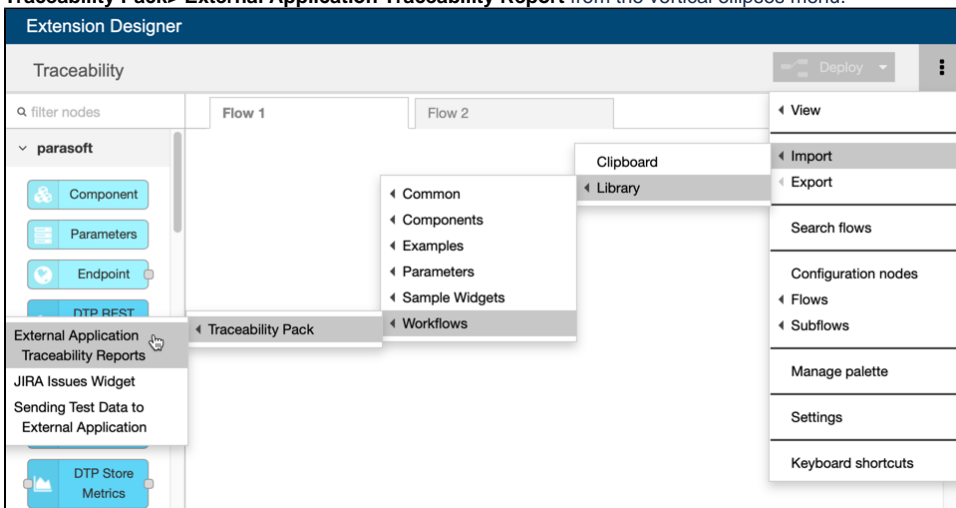
Deploying the Traceability Report

Deploy the report components to your DTP environment 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.



- 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.

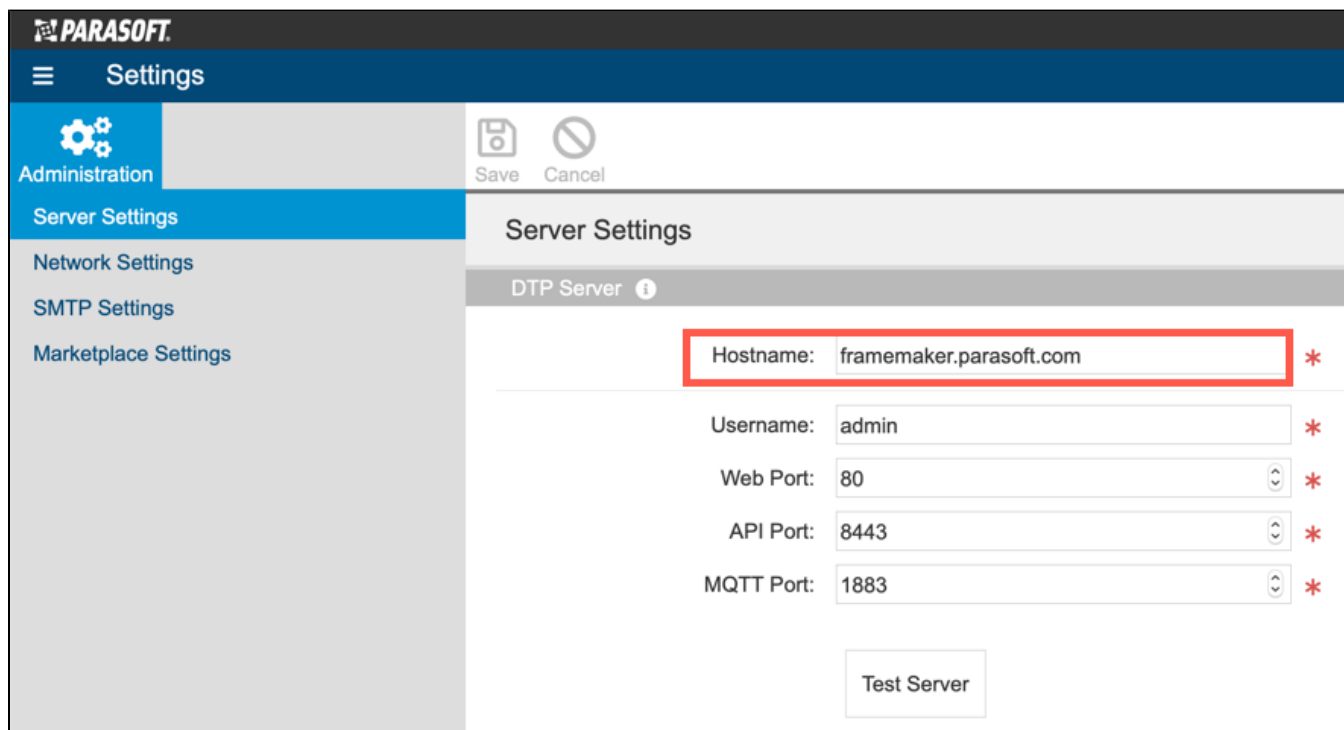


- Click inside the Flow tab to drop the nodes into the service and click **Deploy**.

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 codeBeamer 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. You should also verify that the DTP Enterprise Pack connection is to DTP is configured with the host name of the server running DTP.



By default, Enterprise Pack points to localhost. See [Server Settings](#) for additional information.

- Open Extension Designer and click on the **Services** tab.
- 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.
- 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.
- Click inside the Flow tab to drop the nodes into the service and click **Deploy**.

Usage

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

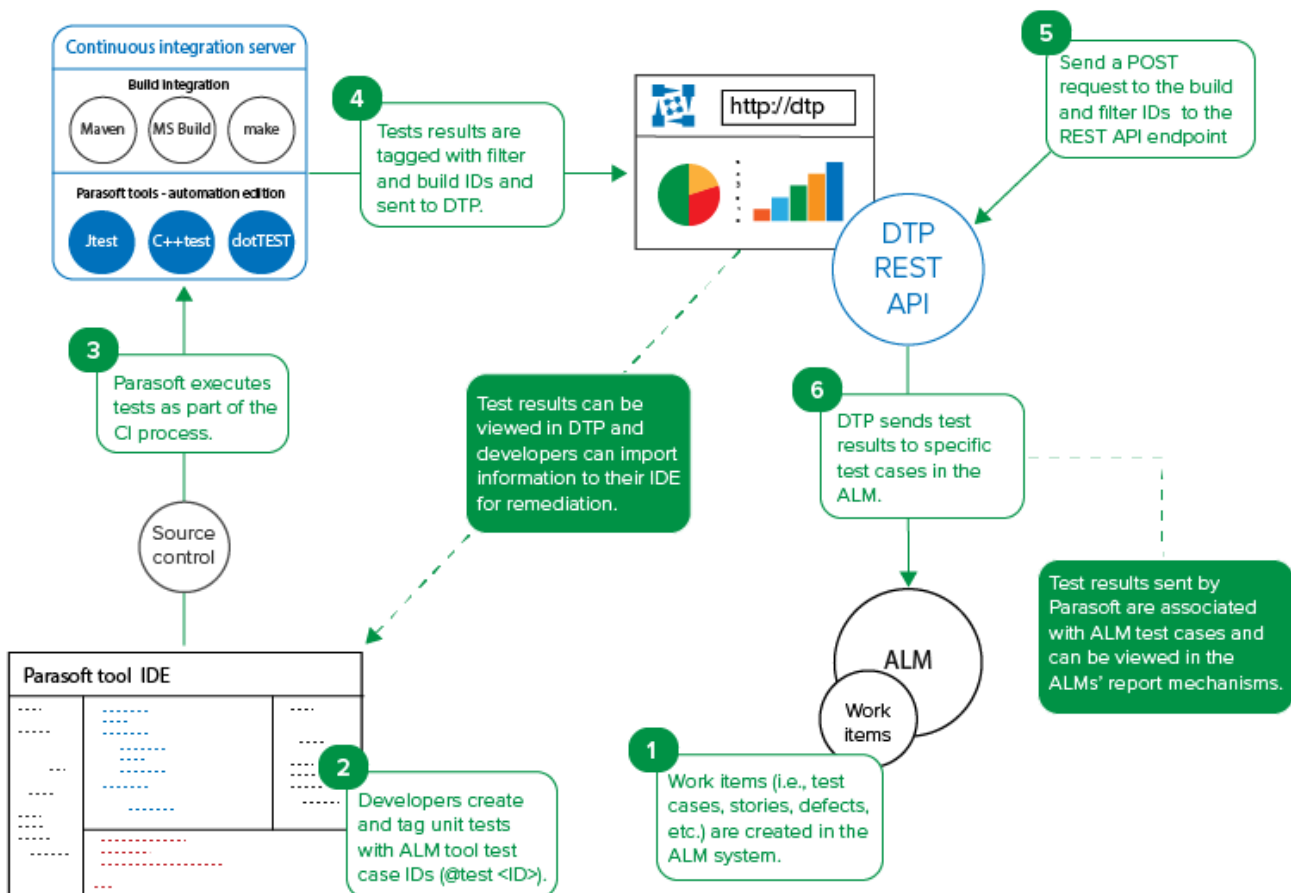
Manually Creating Bugs and Tasks in CodeBeamer ALM

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

- See [Creating an Issue in a Third-party System](#) for instructions on how to manually create bugs and tasks in codeBeamer ALM from the Violations Explorer view.
- See [Creating an Issue in a Third-party System](#) for instructions on how to manually create bugs and tasks in codeBeamer ALM from the Test Explorer view.

Sending Test Data to CodeBeamer

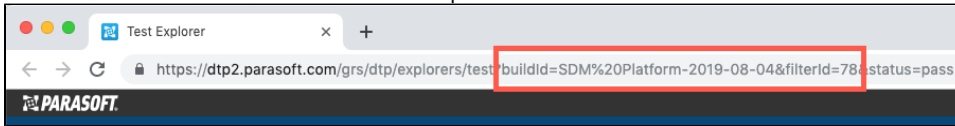
The following diagram shows how you could implement an automated infrastructure for integrating Parasoft DTP and Parasoft test execution tools into your codeBeamer ALM environment:



1. Create items in codeBeamer trackers. The items will be associated with tests executed by Parasoft C/C++test, dotTEST, or Jtest. You can create requirements (i.e., items in a System Requirements Specifications tracker) or test cases (i.e., items in a Test Cases tracker), for example.
2. In your test file, add the codeBeamer test case or requirement IDs using the @test or @req annotation. See the C/C++test, dotTEST, or Jtest documentation for details on adding annotations.
 - Use the @test <codeBeamer test ID> annotation to associate tests with items in a codeBeamer Test Cases tracker.
 - Use the @req <codeBeamer System Requirements Specifications ID> annotation to associate tests with items in a codeBeamer System Requirements Specifications tracker.
3. Execute your tests as part of the CI process. You can also manually execute the tests from the IDE.
4. As part of the test execution, Parasoft test execution tools 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 codeBeamer 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 codeBeamer. Pass the DTP filter and build IDs as URL parameters in the API call:

```
curl -X POST -u <username>:<password> "https://<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 codeBeamer work items.

- When DTP locates results with an `@test <ID>`, it will search for items in Test Case trackers with a matching ID in codeBeamer and update the item. No action will be taken if the unit test case IDs do not exist in codeBeamer.
- When DTP locates results with an `@req <ID>`, it will search for items in System Requirements Specifications trackers with a matching ID in codeBeamer. If a match is found, Test Runs will be added to the Test Case associated with the requirement. If there are no Test Cases for the requirement ID, then Test Case will be created and a Test Run will be added.
- An external-app-sync.log file will also be written to the `<DTP_INSTALL>/logs` directory. This log file contains progress information about sending test results from DTP to codeBeamer.

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

```
{
  "createdTestSession": "DTPP-521",
  "created": [
    "DTPP-519, testName = testBagSumAdd"
  ],
  "updated": [
    "DTPP-519, testName = testBagSumAdd",
    "DTPP-518, testName = testBagSimpleAdd"
  ],
  "ignored": [
    "MAGD-567, testName = testBagNegate",
    "QAP-512, testName = testTryThis3",
    "QAP-512, testName = testTryThis4",
    "MAGD-567, testName = testBagMultiply"
  ]
}
```

Viewing Results in CodeBeamer

You will be able to view results in codeBeamer after sending the test data. The following image shows a System Requirement Specifications tracker in codeBeamer. The tracker contains several test cases.

DTP » Trackers » System Requirement Specifications
Req for demo #4403 HEAD / v1

No suitable Reference Item found

Generate Risk Define Approve Transitions

Tags: not added yet

Tracker: System Requirement Specifications Business Value: -- Risk: --
 Status: **NEW** Type: -- Complexity: --
 Release: -- Submitted by: *devtest* Today 07:19 Modified by: *devtest* Today 07:19
 Assigned to: -- Customer Requirement: -- Team: --
 Story Points: -- Color: --

Traceability

Description

Details

Comments & Attachments Associations Children Downstream References (21) SCM Commits History (1) Baselines All (1)

Tracker Type: Test Cases (21) Status: Unresolved Total 0 resolved/closed 21 unresolved

Summary	BV	Status	Release	Submitted at	Submitted by	Assigned to
[TESTCASE-4426] testvar23_19-09-04T14:22:38 ↳ SRS-4403		NEW		Today 07:40	<i>devtest</i>	--
[TESTCASE-4425] testvar20_19-09-04T14:22:38 ↳ SRS-4403		NEW		Today 07:40	<i>devtest</i>	--
[TESTCASE-4424] testvar19_19-09-04T14:22:38 ↳ SRS-4403		NEW		Today 07:40	<i>devtest</i>	--
[TESTCASE-4423] testvar18_19-09-04T14:22:38 ↳ SRS-4403		NEW		Today 07:40	<i>devtest</i>	--

You can drill down into a test case to view details, such as test runs for the test case.

DTP > Trackers > Test Cases
testvar23_19-09-04T14:22:38 #4426 HEAD / v1

Upstream References [SRS-4403](#) ☆☆☆☆

Design Accept

Tags: not added yet

Tracker: Test Cases Priority: Normal Status: **NEW**

Type: Automated Submitted by: *devtest* Today 07:40 Modified by: *devtest* Today 07:40

Assigned to: --

Verifies: [Req for demo](#)

Pre-Action: --

Post-Action: --

Reusable: False

Traceability

Test Parameters (0)

Description

Details

Test Steps	Test Runs	Reported bug(s)	Comments & Attachments	Associations	Children	History (1)	Baselines	All (1)
Release: <input type="text" value="Any"/> Configuration: <input type="text" value="Any"/> Run by: <input type="text" value="Any"/> After: <input type="text"/> Before: <input type="text"/> <input type="button" value="GO"/>								
	[TESTRUN-4450] Run of testvar23_19-09-04T14:22:38	--	--	PASSED	--	0s	Today 07:40	

One Test Run found.

Click on a test run to view execution details, including test file, build ID, and test author.

DTP » Trackers » Test Runs » Results from Parasoft DTP - demo19-09-04T14:22:38
Run of testvar23_19-09-04T14:22:38 #4450 HEAD / v2

Upstream References TESTCASE-4426 Level 1 SRS-4403

Tags: not added yet

Test Results: Results from Parasoft DTP - demo19-09-04T14:22:38

Tracker: Test Runs	Test Set: --	Priority: --
Status: FINISHED	Result: PASSED	Completed at: Today 07:40
Release: --	Test Configuration: --	Build: --
Submitted by: devtest Today 07:40	Modified by: devtest Today 07:40	Assigned to: --
Running Time: 0s	Reported bug(s): --	

[TESTCASE-4426] testvar23_19-09-04T14:22:38 v 1

Traceability

Conclusion

[View Test in Parasoft DTP](#)

Test File Name: MoneyTest.java
Test Name: testvar23_19-09-04T14:22:38
Build Id: demo19-09-04T14:22:38
Author: kprzybyla
Location: src/test/java/examples/junit/MoneyTest.java

Details

Comments & Attachments	Associations	History (2)	Baselines	All (2)						
<table border="1"> <thead> <tr> <th>Type</th> <th>Submitted</th> <th>Comment</th> </tr> </thead> <tbody> <tr> <td></td> <td> devtest Today 07:40</td> <td> Finish Version 2: Status: Finished, Result: Passed, Description: [View Test in Parasoft DTP]http://leto.parasoft.com.pl:8080/grs/dtp/explorers/test?filterId=16&testCaseId=0fbdfa44-a4e5-37a4-b72c-4130f7620ee8__ Test File Name: __ MoneyTest.java__ Test Name: __ testvar23_19-09-04T14:22:38__ Build Id: __ demo19-09-04T14:22:38__ Author: __ kprzybyla__ Location: __ src/test/java/examples/junit/MoneyTest.java </td> </tr> </tbody> </table>					Type	Submitted	Comment		devtest Today 07:40	Finish Version 2 : Status: Finished, Result: Passed, Description: [View Test in Parasoft DTP] http://leto.parasoft.com.pl:8080/grs/dtp/explorers/test?filterId=16&testCaseId=0fbdfa44-a4e5-37a4-b72c-4130f7620ee8 __ Test File Name: __ MoneyTest.java__ Test Name: __ testvar23_19-09-04T14:22:38__ Build Id: __ demo19-09-04T14:22:38__ Author: __ kprzybyla__ Location: __ src/test/java/examples/junit/MoneyTest.java
Type	Submitted	Comment								
	devtest Today 07:40	Finish Version 2 : Status: Finished, Result: Passed, Description: [View Test in Parasoft DTP] http://leto.parasoft.com.pl:8080/grs/dtp/explorers/test?filterId=16&testCaseId=0fbdfa44-a4e5-37a4-b72c-4130f7620ee8 __ Test File Name: __ MoneyTest.java__ Test Name: __ testvar23_19-09-04T14:22:38__ Build Id: __ demo19-09-04T14:22:38__ Author: __ kprzybyla__ Location: __ src/test/java/examples/junit/MoneyTest.java								

Viewing the Traceability Report

If the External Application Traceability Report has been deployed to your system (see [Enabling the Requirements Traceability Report](#)), you can add widgets to your dashboard to monitor traceability from requirements 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.

Add Widget

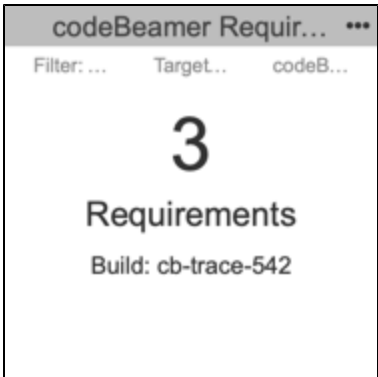
AUTOSAR	JIRA Requirements - Pie	codeBeamer Requirements - Pie 2 x 1 A Pie Chart widget showing requirements data relative to the selected type: Test, Violation, or Review Title: <input type="text" value="codeBeamer Requirements - Pie"/> Filter: <input type="text" value="Dashboard Settings"/> Target Build: <input type="text" value="Dashboard Settings"/> Type: <input type="text" value="Tests"/> codeBeamer Project: <input type="text"/>
MISRA	JIRA Requirements	
PCI DSS	Polarion Requirements	
Traceability	Polarion Requirements - Pie	
Build Results	VersionOne Requirements	
Code	VersionOne Requirements - Pie	
Compliance	codeBeamer Requirements	
Coverage	codeBeamer Requirements - Pie	
Diagnostics		
Metrics		
Process Intelligence		
Static Analysis		
Tests		
Custom		

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	<i>Pie widget only</i> . 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 codeBeamer project from the drop-down menu.

Requirements Widget

This widget shows the number of requirements from the specified codeBeamer project.



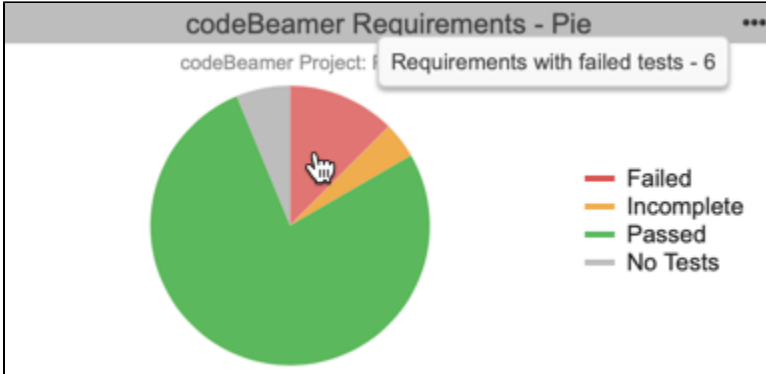
Click on the widget to open the [Requirement Traceability report](#).

Pie Widget

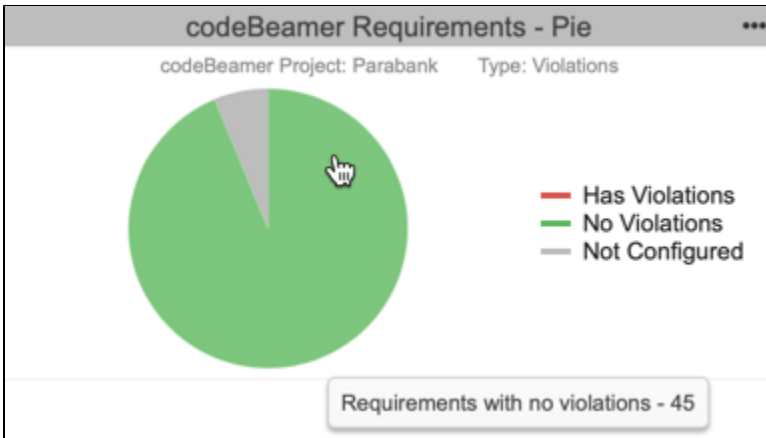
Unit testing, functional testing, static analysis, and peer reviews are common activities for verifying that requirements have been properly and thoroughly implemented. This widget shows the overall status of the project requirements in the context of those software quality activities. You can add a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

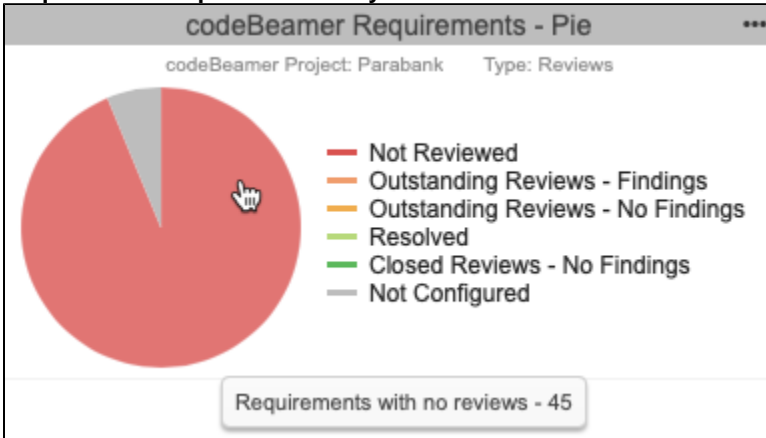
Requirements Implementation Status by Tests



Requirements Implementation Status by Violations



Requirements Implementation by Reviews



Understanding the Requirement Traceability Report

The report lists the codeBeamer requirements and data associated with them.

codeBeamer Requirement Traceability

Filter: Parabank-codeBeamer Target Build: PARABANK3-20180510

Type: All

codeBeamer Requirement		Tests					Files		Reviews	
Key	Summary	Success % ↑	Total	✓	✗	⚠	📄	🔍	🗉	💬
2201	Customer Requirement Specifications	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
2202	System Requirement Specifications	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
4502	Admin Page: provide REST Endpoint form	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
3242	A customer may login to Parabank application	✗ 50.00%	8	4	4	0	0	0	0 / 0	0 / 0
6350	Create bookstore SOAP example service	✗ 54.55%	55	30	25	0	0	0	0 / 0	0 / 0
3158	Create a basic Parabank application	✗ 61.11%	54	33	21	0	0	0	0 / 0	0 / 0
6014	ATM Service - create a SOAP Service to Withdraw funds	✗ 72.22%	18	13	5	0	0	0	0 / 0	0 / 0
5090	Account Services - a customer can see specific account activity	⚠ 75.00%	4	3	0	1	0	0	0 / 0	0 / 0
6434	About Us page to explain ParaBank is not a real bank	✗ 75.00%	4	3	1	0	0	0	0 / 0	0 / 0
3494	Admin page: admin can initialize database	✗ 76.92%	26	20	6	0	0	0	0 / 0	0 / 0
5342	Account Services - A customer can search for transaction by transaction ID	⚠ 90.91%	11	10	0	1	0	0	0 / 0	0 / 0
3410	A customer can retrieve login information	✓ 100.00%	19	19	0	0	0	0	0 / 0	0 / 0
4334	Admin Page: Provide Swagger doc for REST API	✓ 100.00%	7	7	0	0	0	0	0 / 0	0 / 0
4448	Admin Page: Provide SOAP Endpoint list form	✓ 100.00%	15	15	0	0	0	0	0 / 0	0 / 0

You can perform the following actions:

- Click on a link in the Key column to view the tracker in codeBeamer ALM.
- Click on a link in the Summary column or one of the Test columns to view the test-related information associated with the tracker in the codeBeamer Requirement Details Report.
- Click on a link in one of the Files columns to view the static analysis-related information associated with the tracker in the codeBeamer Requirement Details Report.
- Click on a link in one of the Reviews columns to view the change review-related information associated with the tracker in the codeBeamer Requirement Details Report.

Requirement Traceability Report by Type

Clicking on a section of the codeBeamer Requirements - 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.

codeBeamer Requirement Traceability

Filter: Parabank-codeBeamer Target Build: PARABANK3-20180510

Type: Tests Category: Passed

codeBeamer Requirement		Tests					Files		Reviews	
Key	Summary	Success %	Total							
3326	A customer can register to Parabank application	100.00%	11	11	0	0	0	0	0/0	0/0
3410	A customer can retrieve login information	100.00%	19	19	0	0	0	0	0/0	0/0
3578	Admin Page: admin can clean the database	100.00%	7	7	0	0	0	0	0/0	0/0
3662	Admin Page: admin can start or shutdown JMS service	100.00%	12	12	0	0	0	0	0/0	0/0
3746	Admin Page: admin and tester can choose different data access mode	100.00%	11	11	0	0	0	0	0/0	0/0
3830	Admin Page: implement SOAP data access mode	100.00%	7	7	0	0	0	0	0/0	0/0
3914	Admin Page: Implement REST API with XML payload	100.00%	11	11	0	0	0	0	0/0	0/0
3998	Admin Page: implement REST API with JSON payload	100.00%	13	13	0	0	0	0	0/0	0/0
4082	Admin Page: implement JDBC Data Access mode backend	100.00%	14	14	0	0	0	0	0/0	0/0
4166	Admin Page: Provide WDSL Endpoint	100.00%	5	5	0	0	0	0	0/0	0/0
4250	Admin Page: Provide WADL link	100.00%	5	5	0	0	0	0	0/0	0/0

Understanding the Requirement Details Report

The codeBeamer Requirement Details report provides additional information about the files, static analysis findings, and tests associated with a specific codeBeamer requirement. You can open this report by clicking on a requirement in the codeBeamer Requirement Traceability report.

codeBeamer Requirement Details

Filter: Parabank-codeBeamer Target Build: PARABANK3-20180510

3326: A customer can register to Parabank application

Test Success %	Total	Pass	Fail	Incomplete	Files	Violations	Outstanding Reviews	Outstanding Findings
100.00%	2	2	0	0	2	77	0/0	0/0

File / Path	Tests	Status
test/com/parasoft/parabank/dao/jdbc/JdbcPositionDaoTest.java	testUpdatePosition	Pass
test/com/parasoft/parabank/dao/jdbc/JdbcPositionDaoTest.java	testCreatePosition	Pass
test/com/parasoft/parabank/dao/jdbc/JdbcPositionDaoTest.java / testCreatePosition		

25 items per page 1 - 2 / 2 items

The first tab shows the results of the tests that were executed to verify the specific requirement. Click on a test name to view the test in the [Test Explorer](#).

codeBeamer Requirement Details

Filter: Parabank-codeBeamer Target Build: PARABANK3-20180510

🔗 3326: A customer can register to Parabank application

Test Success %	Total	✔ Pass	✘ Fail	Incomplete	📁 Files	🚫 Violations	📄 Outstanding Reviews	🗨 Outstanding Findings
100.00% ✔	2	2	0	0	2	77	0 / 0	0 / 0

Files	Violations
com.parasoft.parabank:parabank:src/com/parasoft/parabank/domain/validator/PayeeValidator.java	51
com.parasoft.parabank:parabank:src/com/parasoft/parabank/service/ParaBankServiceConfiguration.java	26

⏪ ⏩ 1 25 items per page 1 - 2 / 2 items

The second tab shows the files associated with the specific requirement, as well as the static analysis violation 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.

codeBeamer Requirement Details

Filter: Parabank-codeBeamer Target Build: PARABANK3-20180510

🔗 3326: A customer can register to Parabank application

Test Success %	Total	✔ Pass	✘ Fail	Incomplete	📁 Files	🚫 Violations	📄 Outstanding Reviews	🗨 Outstanding Findings
N/A	0	0	0	0	3	486	1 / 1	0 / 0

Reviews	Review Status	🗨 Open	🗨 In Progress	🗨 Closed
Baseline: parabank-requirementsTrace, Target: parabank-requirementsTrace3	Open	0	0	0

⏪ ⏩ 1 25 items per page 1 - 1 / 1 items

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](#).