

# JIRA Traceability Report

In this section:

- [Introduction](#)
- [Requirements](#)
- [Installing the Workflow](#)
- [Preparation](#)
- [Flow Configuration](#)
- [Reviewing Results](#)

## Introduction

This artifact helps you demonstrate traceability between requirements stored in JIRA and test, static analysis, and build review data generated using Parasoft tools. It includes a widget and drill-down report that display test data correlated with existing JIRA requirements. In addition to configuring and deploying this artifact, you must also ensure that the test and analysis tools used in your project are properly configured and that tests executed with the tool are associated with the requirements. Requirements traceability must also be enabled in DTP. The overall process is described in more detail in [Enabling Requirements Traceability](#). You should also consult the documentation for the Parasoft testing tool used in your project.

## Requirements

- DTP Enterprise Pack 5.3.3.
- Parasoft DTP 5.3.3.
- At least one project set up in a JIRA.
- At least one DTP filter configured to receive dynamic analysis details associated with one or more requirements from the system with the correct ID. The builds displayed in the filter must also contain static analysis data for the report to render properly (see [Using Build Administration](#)).



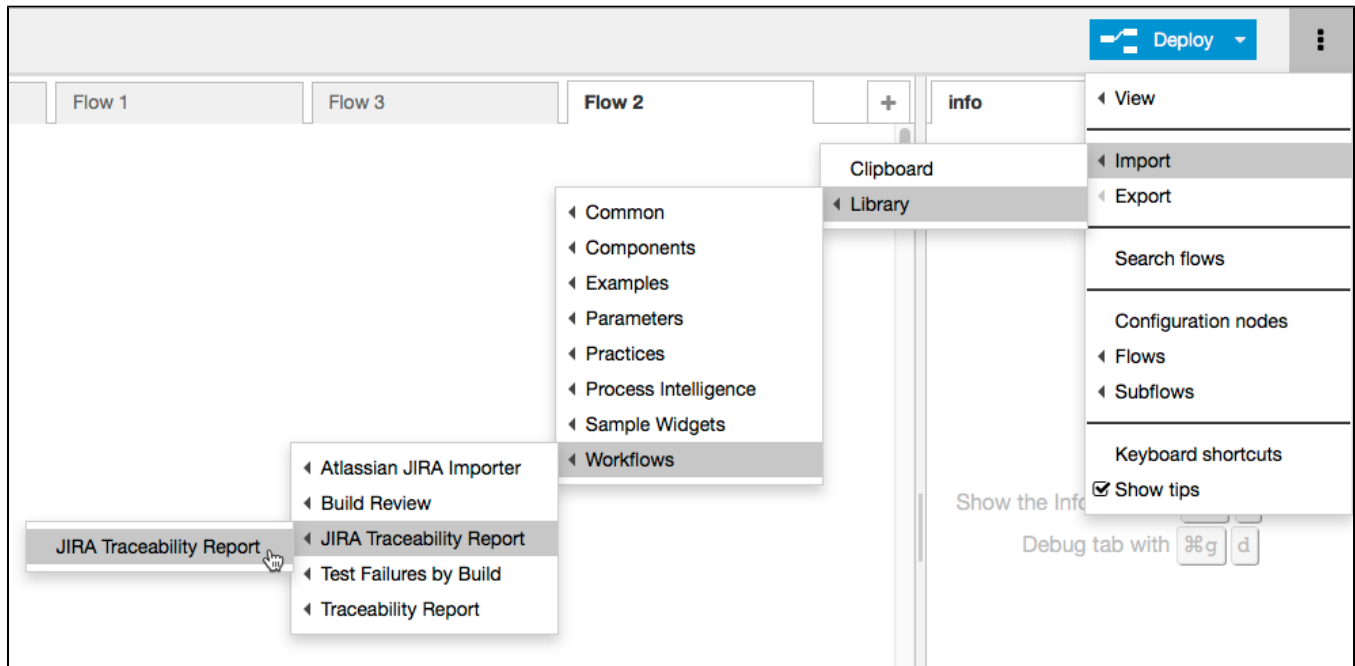
### JIRA Limits Results to 1000 Items

By default, the search API for JIRA will return a maximum of up to 1000 issues. This is an intentional limitation set by Atlassian to avoid any possible OutOfMemory exceptions. See the following JIRA documentation for information on changing this limit:

<https://confluence.atlassian.com/jirakb/exporting-a-filter-result-containing-more-than-1000-issues-191500982.html>

## Installing the Workflow

See [Downloading and Installing Artifacts](#). Import the JIRA Traceability Report workflow into your service(s) in order to use the widget.



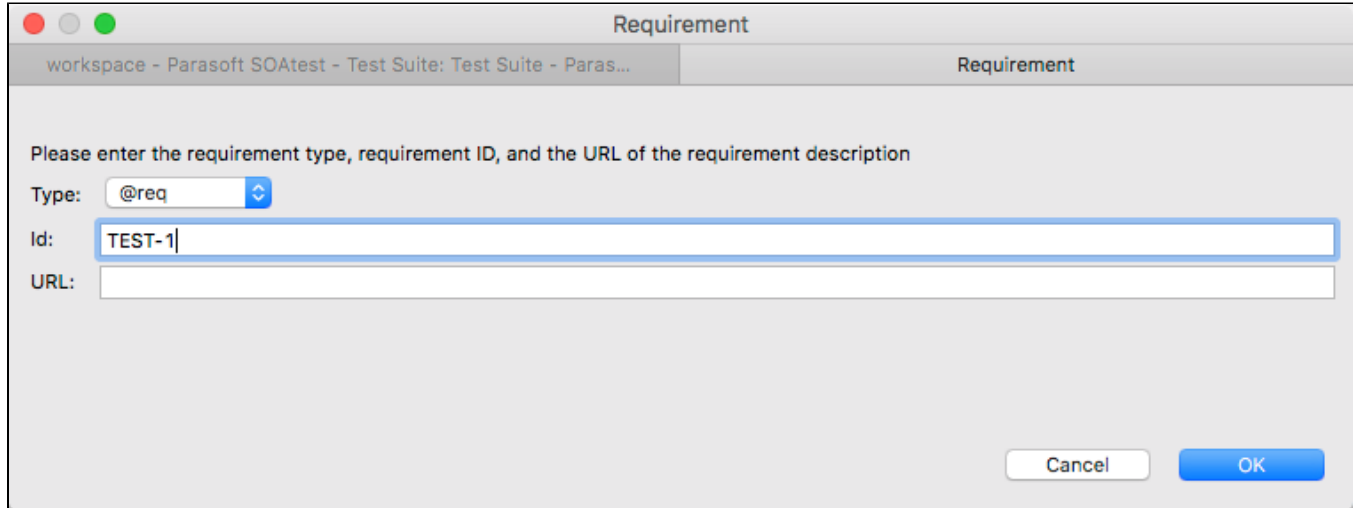
# Preparation

The following configurations must be made to enabling requirements traceability.

## Associating Tests with Requirements

You must associate tests with existing JIRA requirements to use the workflow. Test-to-requirement associations are made with the Parasoft code analysis and test execution tool you are using to verify and validate the requirements.

You can make the associations in the Parasoft tool UI by choosing `@req` as the association type and entering the JIRA requirement ID:



You can also correlate a unit test with a requirement by adding the `@req` tag and JIRA ID to a comment in the test class:

### Example: Correlating a unit test with a requirement

```
/**
 * @req TEST-1
 */
@Test
public void testSomething(){
    . . .
}
```

See the documentation for your test and code analysis tool for details on how to associate tests with requirements.

## Setting Build and Filter IDs

You must also ensure that the analysis tool used to execute tests and send data to DTP tags the data with the correct build and filter IDs. The IDs can be set locally in several ways:

- Locally the tool's `.properties` (see the documentation for your tool)
- In DTP on a per project case (see [Creating and Managing Projects](#))
- Globally in DTP for all projects (see [Configuring Parasoft Test for All Projects](#))

## Associating Requirements with Files in DTP

You must create a document that maps source files to requirement IDs and execute a script that ships with DTP to enable requirements traceability. This process is described in [Associating Requirements with Files](#).

## Verifying that Build ID Has Data

Builds used for this artifact must have test and static analysis data, otherwise the report will not be able to render the data properly. See [Using Build Administration](#) for information about checking build IDs.


## Flow Configuration



Double-click the **Set JIRA URL** node and specify the location of you JIRA server.


The screenshot shows the 'Edit change node' dialog for a node named 'Set JIRA URL'. The dialog has a title bar with 'Delete', 'Cancel', and 'Done' buttons. The 'Name' field contains 'Set JIRA URL'. Under the 'Rules' section, there are two rule entries. The first rule is 'Set' with a dropdown menu set to 'flow. jiraURL'. The 'to' field for this rule contains 'http://jira-demo.parasoft.com:8720', which is highlighted with a red box. The second rule is 'Set' with a dropdown menu set to 'msg. payload'. The 'to' field for this rule contains 'JIRA URL Configured'.


Double-click the **Get JIRA Projects** node and click the edit button for the JIRA REST API Configuration field.

**Edit http request node**


 Name


 Configuration  

 Method

 Endpoint

---

 Response

 Return

Tip: If the JSON parse fails the fetched string is returned as-is.

Enter your JIRA credentials and click **Update**.

http request > **Edit http-request-config node**

Name

URI

Username

Password

## Reviewing Results

This artifact ships with two widget that drill down into a requirements traceability report:

- [JIRA Requirements Widget](#): Shows the number of requirements associated with the project
- [JIRA Requirements - Pie Widget](#): Shows the status of testing activities (static analysis, tests, reviews) associated with project's requirements.

## Adding and Configuring the Widgets

See [Adding Widgets](#) for instructions on adding the JIRA Requirements Traceability Widget. After deploying the artifact, the widget will appear in a separate JIRA category.

### Add Widget

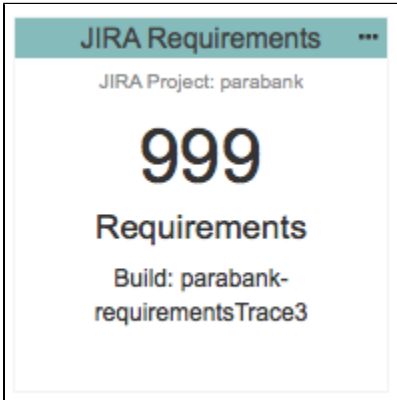
JIRA	JIRA Requirements	JIRA Requirements
Build Results	JIRA Requirements - Pie	<b>JIRA Requirements</b> 1 x 1 A summary widget showing the number of JIRA requirements for the selected project. This widget drills down to a JIRA requirement traceability report.
Code		<b>Title:</b> <input type="text" value="JIRA Requirements"/>
Compliance		<b>Filter:</b> <input type="text" value="Dashboard Settings"/>
Diagnostics		<b>Target Build:</b> <input type="text" value="Dashboard Settings"/>
Metrics		<b>JIRA Project:</b> <input type="text"/>
Policies		<b>JIRA Query:</b> <input type="text" value="issuetype = Story"/>
Process Intelligence		
Static Analysis		
Tests		
Custom		

You can configure the following settings:

<b>Title</b>	Enter a new title to replace the default title that appears on the dashboard.
<b>Filter</b>	Choose Dashboard Settings to use the dashboard filter or choose a filter from the drop-down menu.
<b>Target Build</b>	You can use the build specified in the dashboard settings, the latest build, or a specific build from the drop-down menu. Also see <a href="#">Configuring Dashboard Settings</a> .
<b>Type</b>	<i>JIRA Requirements - 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.
<b>JIRA Project</b>	Choose a project in JIRA from the drop-down menu.
<b>JIRA Query</b>	Specify a JIRA search string to filter the data shown in the widget. By default, the widget searches for JIRA stories (requirements). Do not specify a project when entering your own search string. See the JIRA documentation for using JIRA Query Language (JQL).

## JIRA Requirements Widget

The widget shows the number of requirements associated with the build and filter IDs.



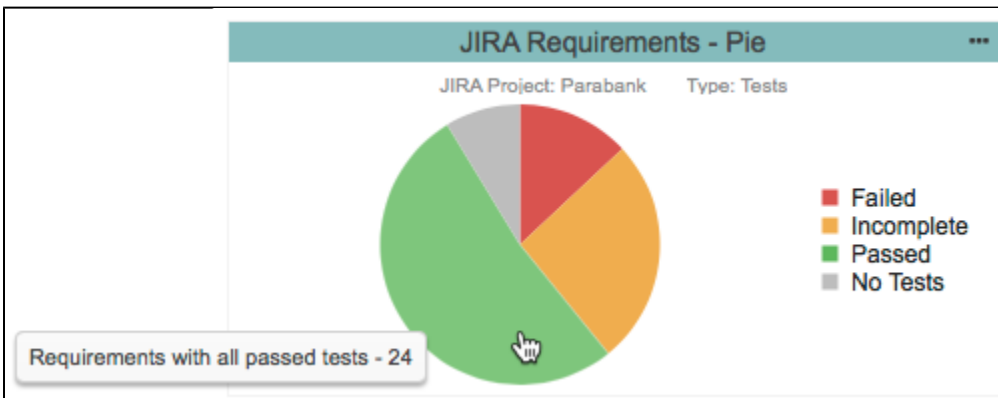
Click on the widget to open the JIRA Requirement Traceability report.

## JIRA Requirements - Pie Widget

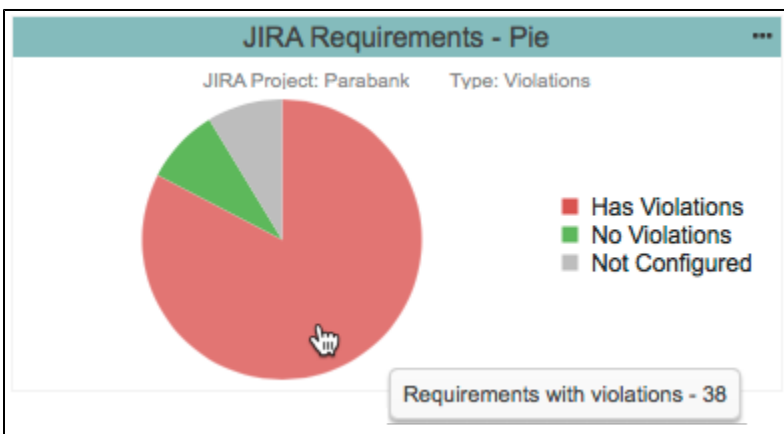
Static analysis, unit testing, functional testing, and peer reviews are common activities for verifying that requirements have been properly and thoroughly implemented. This widget shows the overall status of the project requirements in the context of those software quality activities. You can add a 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 JIRA Requirement Traceability report filtered by that type.

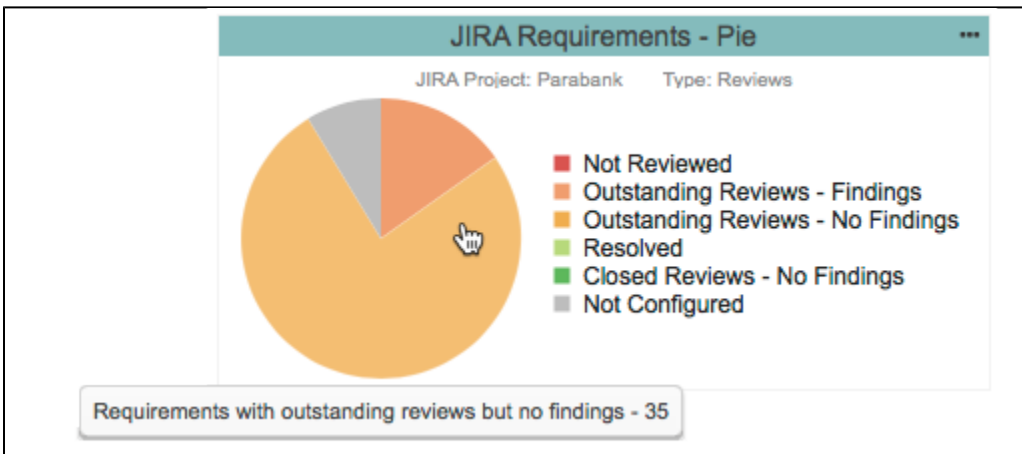
### Requirements Implementation Status by Tests



### Requirements Implementation Status by Violations



### Requirements Implementation by Reviews



## Understanding the JIRA Requirement Traceability

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

JIRA Requirement Traceability											
Filter: parabank Build: parabank-requirementsTrace3											
Type: All											
JIRA Requirement		Tests					Files		Reviews		
Key	Summary	Success %	Total	✓	✗	⚠	📄	🔍	📄	🗨	🗨
PAR-1231	User Story 979	70.00%	10	7	1	2	1	111	2 / 2	0 / 0	
PAR-1220	User Story 968	60.00%	5	3	2	0	1	76	2 / 2	0 / 0	
PAR-10	Search for transactions	100.00%	2	2	0	0	0	0	0 / 0	0 / 0	
PAR-1039	User Story 787	100.00%	2	2	0	0	2	0	0 / 0	0 / 0	
PAR-250	Staff panel is hidden for non-staff users	100.00%	2	2	0	0	2	724	0 / 0	0 / 0	
PAR-270	User Story 18	100.00%	2	2	0	0	1	22	0 / 0	0 / 0	
PAR-290	User Story 38	100.00%	2	2	0	0	1	0	0 / 0	0 / 0	
PAR-590	User Story 338	100.00%	2	2	0	0	0	0	0 / 0	0 / 0	
PAR-280	User Story 28	100.00%	2	2	0	0	1	79	0 / 0	0 / 0	
PAR-632	User Story 380	100.00%	1	1	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 [JIRA Requirement Details report](#) for the requirement.
- Click on a link in the Summary column or one of the Test columns to view the test-related information associated with the requirement in the JIRA Requirement Details Report.
- Click on a link in one of the Files columns to view the static analysis-related information associated with the requirement in the JIRA Requirement Details Report.
- Click on a link in one of the Reviews columns to view the change review-related information associated with the requirement in the JIRA Requirement Details Report.

## JIRA Requirements Traceability Report by Type

Clicking on the a section of the JIRA 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.

## JIRA Requirement Traceability

Filter: Parabank-v3 Build: PARABANK3-20170619

Type: Tests Category: Incomplete

JIRA Requirement		Tests				Files	Reviews		
Key	Summary	Success %	Total						
PAR-11	Admin Page: implement REST API with JSON payload	92.31%	13	12	0	1	16	10	2 / 2
PAR-12	Admin Page: implement JDBC Data Access mode backend	92.86%	14	13	0	1	26	6	2 / 2
PAR-15	Admin Page: Provide Swagger doc for REST API	85.71%	7	6	0	1	17	2	2 / 2
PAR-16	Admin Page: Provide SOAP Endpoint link form	86.67%	15	13	0	2	15	0	2 / 2
PAR-18	Admin Page: Create LoanProcessor and provide link to WSDL and endpoint.	85.71%	7	6	0	1	27	9	2 / 2
PAR-21	Account Service - A customer can open Savings account	91.67%	12	11	0	1	27	2	2 / 2
PAR-24	Account Services - a customer can see specific account activity	66.67%	3	2	0	1	16	1	2 / 2

## Understanding the JIRA Requirement Details Report

The JIRA Requirement Details report provides additional information about the files and tests associated with the JIRA requirement. You can open this report by click on a requirement in the JIRA Requirement Traceability report.

### JIRA Requirement Details

Filter: parabank Requirement: PAR-250 Build: parabank-requirementsTrace3

PAR-250: Staff panel is hidden for non-staff users

Test Success %	Total			Incomplete				
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 provides the file paths and test results. Click on a test name to view the test in the [Test Explorer](#).

### JIRA Requirement Details

Filter: parabank Requirement: PAR-250 Build: parabank-requirementsTrace3

PAR-250: Staff panel is hidden for non-staff users

Test Success %	Total			Incomplete				
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

25 items per page 1 - 2 / 2 items

The second tab shows the static analysis violations associated with the files. Click on a link in the Violations column to view the violations in the [Violations Explorer](#).



## JIRA Requirement Details

Filter: parabank Requirement: PAR-874 Build: parabank-requirementsTrace3

🔗 PAR-874: User Story 622

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

⏪ ⏩ 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](#).