

# Indicating Code and Test Correlations

This topic explains how to correlate code with tests in order to configure change-based testing and requirements/defects traceability through DTP. In order to process correlations, SOAtest must be connected to DTP (see [Connecting to Parasoft Development Testing Platform](#)).

Sections include:

- [Correlating Code During Implementation](#)
- [Correlating SOAtest Tests](#)
- [Correlating Load Test Scenarios](#)
- [Using Custom Defect/Issue Tracking Tags](#)
- [Linking to Correlated Artifacts in Reports](#)

## Correlating Code During Implementation

### Report Center Task Correlations

When the development team works with tasks, the source code that is modified when a task is "active" will be correlated with that task—and any associated requirements or defects specified for that task in Report Center.

### Check-in Correlations

If code is not correlated to a task with associated requirements or defects, then the related requirement or defect should be specified upon check-in.

- **Requirements:** Include the following text in a revision comment when committing a file into the repository (where "NNN" is the requirement ID: @req NNN).
- **Defects:** Include the following text in a revision comment when committing a file into the repository (where "NNN" is the defect tracking ID: @PR NNN).
- **Enhancements:** Include the following text in a revision comment when committing a file into the repository (where "NNN" is the feature request ID: @FR NNN).

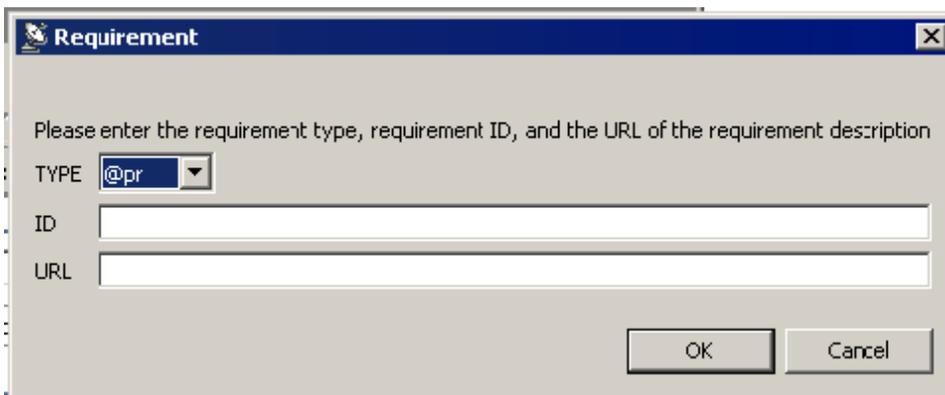
For information on how to use a custom tag to mark defects, see [Using Custom Defect/Issue Tracking Tags](#) below.

## Correlating SOAtest Tests

The **Requirements and Notes** tab of SOAtest's test suite configuration panel allows you to identify requirements/tasks/bugs associated with the tests in the test suite. The requirements you define will appear in Structure reports (and also in Report Center), allowing managers and reviewers to determine whether the specified test requirements were completed.

To configure correlations:

1. Select a node from the test suite tree within the **Requirements and Notes** tab.
2. Click the **Add** button.
3. In the **Type** box, select a requirement type. DTP will use this information to associate the test suite's test cases to the specified element type. For instance, if it is associated with a specific bug, information about the test case's status will be considered for Report Center's bugs graphs. Custom tags can be added as described in [Using Custom Defect/Issue Tracking Tags](#). Default tags are:
  - @pr and @bugzilla: for bugs.
  - @fr: for feature requests.
  - @req: for requirements.
  - @task: for tasks.
4. Enter an **ID** and a **URL** for the requirement and click **OK**.



The screenshot shows a dialog box titled "Requirement" with a close button in the top right corner. The main text inside the dialog reads: "Please enter the requirement type, requirement ID, and the URL of the requirement description". Below this text are three input fields: "TYPE" with a dropdown menu currently showing "@pr", "ID" with an empty text box, and "URL" with an empty text box. At the bottom right of the dialog are two buttons: "OK" and "Cancel".

The requirement you defined will display in the **Requirements** table within the **Requirements and Notes** tab and correspond to the test suite node you selected and all of its child nodes.

5. (Optional) If you want to enter notes for the test suite, enter a description in the **Notes** field. This option is useful in that it enables you to view a quick description of the test suite purpose.

## Correlating Load Test Scenarios

See [Reviewing and Customizing Load Test Results](#).

## Using Custom Defect/Issue Tracking Tags

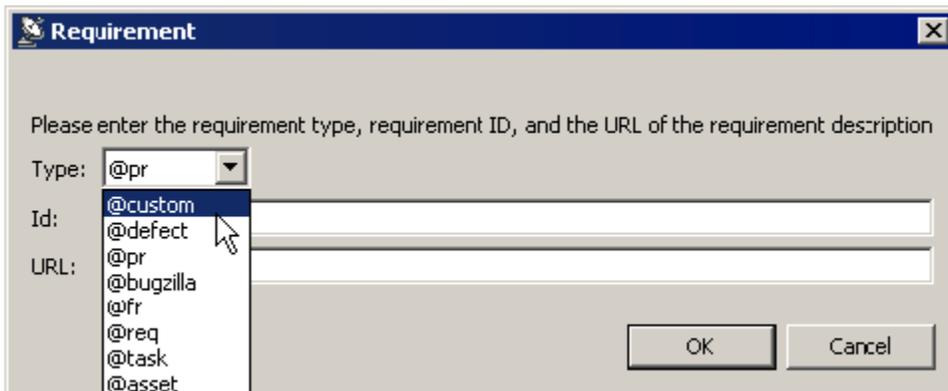
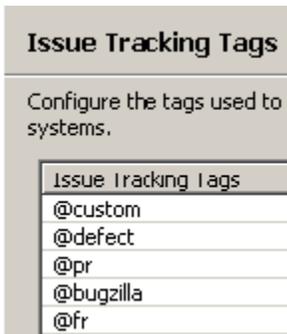
You can customize the tags your team can use to associate a test case with an issue from an issue/feature/defect tracking system (e.g, JIRA, Bugzilla, etc.). This allows you to configure the tag to match the language that your organization uses to refer to defects. For instance, some organizations use the term "PR" to refer to a defect. Others use "defect" and would thus prefer to use a custom @defect tag rather than the default @pr tag.

The default set of tags is @pr and @bugzilla.

If you do not want to use the default defect/issue tracking tags (@pr, @bugzilla), specify aliases for the tags you prefer to use as follows:

1. Choose **Parasoft> Preferences**.
2. Select **Parasoft> Issue Tracking Tags**.
3. Use the available controls to add, modify, or delete tags.
  - A tag is followed by one or more values that must be represented by either a single string or a number of strings separated by separator tokens.
  - Separator token values can be any permutation of comma-separated lists of strings and whitespace-separated lists of strings.

The tags listed here will be available in SOAtest when you configure the Requirements Tracking options in the test suite editor (when you go to **Add** a new requirement correlation). See [Correlating SOAtest Tests](#) for details.



### Customization Example1

Assume that your team uses the issue tracking tags @bugzilla and @defect (a custom issue tracking tag that is configured in SOAtest).

Also assume the following Javadoc comment:

```
/**
 * @pr 2453, 462345
 * @bugzilla 2342
 * @defect 87766
 */
void testFoo1() {}
```

The following PR properties will be sent to Project Center (as separate items):

```
2453
462345
2342
742725
87766
```

The following FR properties will be sent to Project Center (as separate items):

```
8766
996686776
```

## Example 2

Assume the following Javadoc comment:

```
/**
 * @pr 2453, 462345 2342
 * 8766 996686776 - these numbers are not be counted as a part of @pr tag
 * @pr 87766
 */
void testFoo1() {}
```

The following PR properties will be sent to Project Center (as separate items):

```
2453
462345
2342
742725
87766
```

## Example 3

Assume that your team uses the issue tracking tags @bugzilla and @defect (a custom issue tracking tag that is configured in SOAtest).

Assume the following test case comments:

```
/* TEST_CASE_BEGIN test_foo */ // @pr 123, 456
// @bugzilla 789
// @defect 1357
// @req 7799
void TestSuite::test_foo()
{
//test case body
}
/* TEST_CASE_END test_foo */
```

The following PR properties will be sent to DTP (as separate items):

```
123
456
789
1357
```

The following REQ property will be sent to DTP:

```
7799
```

# Linking to Correlated Artifacts in Reports

If you want SOAtest reports to link to correlated artifacts, you need to specify URL templates that explain how links to these artifacts are formatted. To do this:

1. Choose **Parasoft> Preferences**.
2. Select **Parasoft> Issue Tracking Tags**.
3. In the panel's top table, specify how links to each artifact type are formatted. For example:
  - Mapping @pr to `http://bugzilla.company.com/show_bug.cgi?id=${id}` means that the URL for PR1024 is `http://bugzilla.company.com/show_bug.cgi?id=1024`
  - Mapping @task to `http://host.company.com:8080/grs/jsf/planning/task/edit_task.jsf?entityId=${id}` means that the URL for task 1215 is `http://host.company.com:8080/grs/jsf/planning/task/edit_task.jsf?entityId=1215`