

# Indicating Code and Test Correlations

This topic explains how to correlate code and test correlations in order to configure change-based testing and requirements/defects traceability.

For the correlations to be used on, you need to connect to DTP (see [Connecting to DTP](#)).

Sections include:

- [Correlating Code During Implementation](#)
- [Correlating C++test Tests](#)
- [Using Custom Defect/Issue Tracking Tags](#)
- [Linking to Correlated Artifacts in Reports](#)

## Correlating Code During Implementation

### Correlation Upon Checkin

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 C++test Tests

Parasoft DTP can correlate unit test cases with PRs (problem reports) and FRs (feature requests) defined in a bug-tracking system or in a requirements management system. This allows DTP to track whether each item has an associated test, and whether that test is passing or failing.

### Parasoft C++test Test Correlations

In Parasoft C++test, you can specify correlations in test names or in comments.

#### Using Test Names

If you want DTP to correlate test cases with tasks, requirements, PRs, and FRs, then add the following to the end of the test case name:

- `_PRxxxxx` - Connects the test case to problem report xxxxx.
- `_FRxxxxx` - Connects the test case to feature request xxxxx.
- `_TASKxxxxx` - Connects the test case to task xxxxx.
- `_REQxxxxx` - Connects the test case to requirement xxxxx.

For example, if a test case verifies that PR# 12345 is resolved, you would name it `test_case_PR12345`. If a test case verifies that task #1 is implemented properly, you would name it `test_case_TASK1`.

#### Using Comments

To use comments to indicate test correlations:

- Place the appropriate comment between `CPPTTEST_TEST_CASE_BEGIN` and `CPPTTEST_TEST_CASE_END` test case markers.

The following markers are supported:

- @PR
- @FR
- @TASK
- @REQ
- @ASSET
- @TEST

Do not include other code on the line with the comment, and use only one type of marker per line. Use a comma or a space to separate markers.

For example, the following test will be connected with PRs: 123, 456 and TASK 789.

```
/* CPPTTEST_TEST_CASE_BEGIN test_foo */
// @PR 123, 456
// @TASK 789
void TestSuite::test_foo()
{
    //test case body
}
/* CPPTTEST_TEST_CASE_END test_foo */
```

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



#### Note

You can use any marker/tag your prefer, as long as it is specified as a custom tag under **Preferences> Issue Tracking Tags**. For information on how to use a custom tag to mark defects, see [Using Custom Defect/Issue Tracking Tags](#).

## 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 (for example, Bugzilla). 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.

If you do not want to use the default defect/issue tracking tags, 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.

Builtin Tracking Tags
@asset
@fr
@pr
@req
@task
@test

  

User Issue Tracking Tags

### Customization Example1

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

Also, assume the following Javadoc comment:

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

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

```
2453
462345
2342
742725
87766
```

The following FR properties will be sent to Parasoft DTP(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 Parasoft DTP (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).

Assume the following test case comments:

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

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

```
123
456
789
1357
```

The following REQ property will be sent to Parasoft DTP:

```
7799
```

# Linking to Correlated Artifacts in Reports

If you want 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}](http://bugzilla.company.com/show_bug.cgi?id=${id}) means that the URL for PR1024 is [http://bugzilla.company.com/show\\_bug.cgi?id=1024](http://bugzilla.company.com/show_bug.cgi?id=1024)
  - Mapping @task to [http://concerto.company.com:8080/grs/jsf/planning/task/edit\\_task.jsf?entityId=\\${id}](http://concerto.company.com:8080/grs/jsf/planning/task/edit_task.jsf?entityId=${id}) means that the URL for task 1215 is [http://concerto.company.com:8080/grs/jsf/planning/task/edit\\_task.jsf?entityId=1215](http://concerto.company.com:8080/grs/jsf/planning/task/edit_task.jsf?entityId=1215)