

Code Review Introduction

This topic provides an overview of the code review functionality available across Parasoft Test products.

Sections include:

- [Code Review Overview](#)
- [Workflow Overview](#)
- [Built-In Code Review Test Configurations](#)
- [Code Review Task / Status Reporting](#)

Code Review Overview

Parasoft's Code Review module is designed to make peer reviews more practical and productive by automating preparation, notification, and tracking. It automatically identifies updated code, matches it with designated reviewers, and tracks the progress of each review item until closure. This allows teams to establish a bulletproof review process that ensures the designated items are reviewed and all identified issues are resolved.

Parasoft provides built-in support for the following typical code review flows:

- **Post-commit:** This mode is based on automatic identification of modifications in a source repository via custom source control interfaces. Review tasks are created based on a preconfigured mapping of changed code to reviewers.
- **Pre-commit:** Users can initiate a code review from the desktop by selecting a set of files to distribute for the review, or automatically identify all locally changed source code.

Code Review requires Team Server and at least one Parasoft Test Server Edition.

Workflow Overview

There are two key workflow variations: default vs. restricted and pre-commit vs. post-commit.

Default vs. Restricted

In a restricted workflow, the reviewer must accept each issue. If the author does not agree with the reviewer's suggestion, he needs to discuss this with the reviewer.

The default workflow is more open: when the author receives comments from a reviewer, he can either apply them to the code, or leave the code as is.

Pre-Commit vs. Post-Commit

Pre-Commit

Pre-commit code reviews are for teams who want to review code before adding it to source control. When developers are ready to have a piece of new/modified code reviewed, they run a Code Review Test Configuration from the Parasoft Test UI, then the reviewer is automatically notified about the required review.

Post-Commit

Post-commit code reviews are for teams who want to review code after code is committed to source control. A Code Review Test Configuration is typically scheduled to run automatically on a regular basis (for example, every 24 hours). It scans the designated source control repository to identify code that requires review, then sends this information to Team Server, which then distributes it to the designated reviewer. In this scenario, the code authors do not need to perform any special actions to have their code reviewed; simply committing it to source control is sufficient. After the Code Review Test Configuration runs, the designated reviewers are automatically notified about the required reviews.

In a pre-commit process, a code review package is created for code that needs to be accepted before it is committed to source control. In this type of process, the author should always see that packages are finally accepted. At that point, he can commit the files and then close the package.

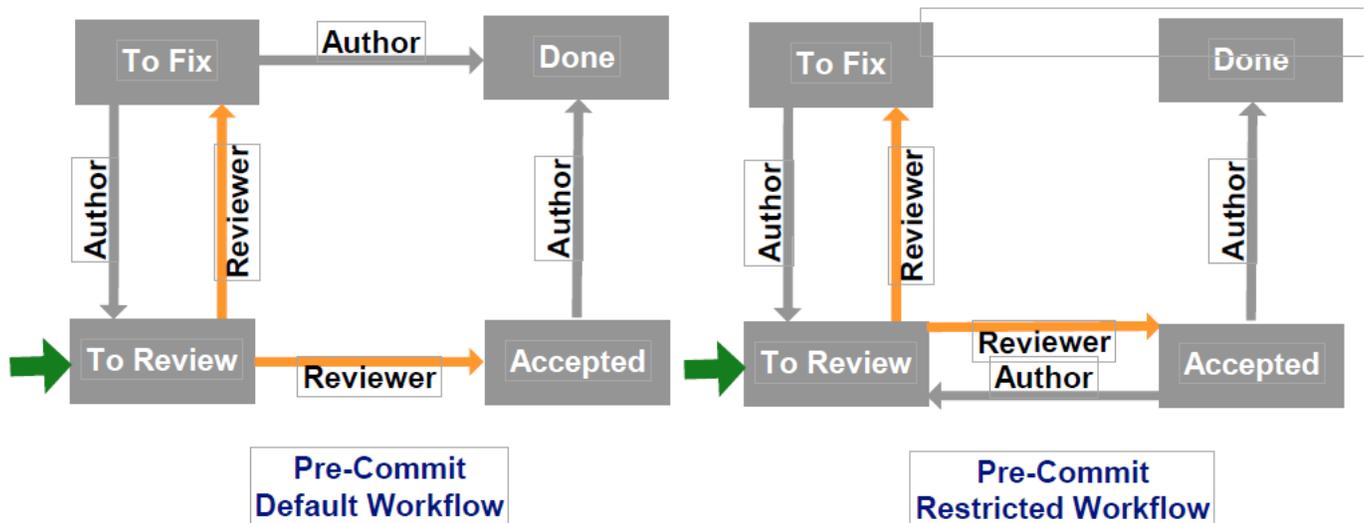
Some teams who submit code for review via a pre-commit process also like to perform a post-commit nightly scan to:

- Generate emails notifying authors and reviewers about their assigned code review tasks.

- Identify any code changes that were committed to source control, but were not submitted for review using the pre-commit procedure.

Workflow Details

The following diagrams illustrate the key workflows available:



Built-In Code Review Test Configurations

- **Pre-Commit:** This Test Configuration is for teams who want to review code before it is committed to source control (see [Pre-Commit vs. Post-Commit](#)). It scans files added or modified locally. To use this Test Configuration, the Code Review Preference **Show user assistant during scanner run** setting must be enabled so that the author can designate the appropriate reviewer(s).
- **Post-Commit (Template):** This Test Configuration is for teams who want to review code after it is committed to source control (see [Pre-Commit vs. Post-Commit](#)). It scans all project files modified in the previous day. This Test Configuration must be duplicated and customized prior to use (e.g, to specify author-reviewer mappings). See [Configuring and Running Post-Commit Code Review Scans](#) for details.
- **Post-Commit (Assign All):** This Test Configuration is for teams who want to review code after it is committed to source control (see [Pre-Commit vs. Post-Commit](#)). It scans all project files modified in the previous day. It can be used without customization. It includes a mapping for the local code review user; it assigns all revisions found in scope (for any author) to the current user.

Code Review Task / Status Reporting

Code Review tasks are reported in the GUI as described in [Working with the Code Review UI](#). Code Review details are also provided in the Code Review section of reports generated from command line or GUI analysis. These reports provide an overview of code review tasks and issues per project and per author. It also provides details on pending issues and code review tasks.

For example:

CODEREVIEW

Project Name	Tasks		
	new reviews	pending reviews	pending issues
aneta_JtestExample91	1	1	1

Author	Tasks				
	new check-ins	new reviews	pending reviews	pending issues	monitored
aneta	0	1	1	1	0
monika	1	0	0	0	0

Pending Issues: 1 aneta [1] ^

/aneta_JtestExample91//examples/bank/General Issue Code review-3

To Review from monika (OWNER) at 2012/07/17 16:03

Task: 2

Modified:

doc/aneta/aneta_JtestExample91/examples/bank/BankAccount.java

To Review from monika (OWNER) at 2012/07/17 16:03

Task: 2

Modified:

doc/aneta/aneta_JtestExample91/examples/bank/BankAccount.java

Checked Files (Details)

```

+ [1/1/1] New / Reviews / Issues
+ [1/1/1] aneta_JtestExample91
+ [0/0/1] /examples/bank
+ [0/0/1] General Issue
  /aneta_JtestExample91//examples/bank/General Issue Code review-3
  To Review from monika (OWNER) at 2012/07/17 16:03
  Task: 2
  Modified:
  doc/aneta/aneta_JtestExample91/examples/bank/BankAccount.java
  To Review from monika (OWNER) at 2012/07/17 16:03
  Task: 2
  Modified:
  doc/aneta/aneta_JtestExample91/examples/bank/BankAccount.java
    
```