

General Code Review Configuration

This section explains general code review configuration options (options that apply to both pre-commit and post-commit processes).

Sections include:

- [Configuring Code Review Preferences](#)
- [Configuring Test Configurations for Scanning](#)
- [Hiding Differences for Specific Pieces of Code](#)
- [Troubleshooting Unpublished or Skipped Reviews](#)
- [Understanding How Code Review Packages are Created](#)
- [Importing Code Review Data from Team Server to the DTP Report Center Database](#)

Configuring Code Review Preferences

To configure Code Review preferences on every team Para installation:

1. Choose **Parasoft> Preferences** to open the Preferences dialog.
2. If you have not already done so, configure source control preferences as follows:
 - a. Select the **Parasoft> Scope and Authorship** category in the left pane of the Preferences panel.
 - b. Enable **Use source control (modification author) to compute scope**.
 - c. Select the **Parasoft> Source Control** category in the left pane of the Preferences panel.
 - d. Specify your team's source control repository. See [Connecting to Source Control](#) for details.
3. If you have not already done so, configure Team Server preferences under **Parasoft> DTP> Team Server**.

Team Server Configuration Note

If your Parasoft solution uses Team Server Named Accounts, ensure that Team Server user accounts can access the Team Server 'Code Review' directory. For details on opening the appropriate path permissions, see the 'Named Accounts' section of the PST Admin Guide.'

4. To store code review data in Report Center (recommended for more efficient storage), specify the settings under **Parasoft> DTP> Project Center** (see [Connecting to Project Center](#) for details).
5. (Desktop installations only) Select the **Parasoft> Code Review** category in the left pane, then complete the Code Review settings as follows:
 - **DTP Settings**
Displays the DTP server host name you have configured. You can disable the **Default** checkbox to modify the port number.
 - **General Settings**
 - **User name:** Enter a unique Code Review name for the current user. The same user name specified here should also be specified in the Code Review Test Configuration.
 - **Show user assistant during scanner run:** This enables a user assistant, which allows you to specify a task identifier, enter notes, specify review priority, and enter a specific reviewer or monitor for each Code Review run. This is intended for pre-commit code reviews.
 - **Notify me about new or updated reviews every ___ minutes:** Select this if you want to be alerted when you need to review code and/or when code that you authored has been reviewed. This option is recommended.
 - **Show completed tasks by:** If you want the code review tasks tree to show completed tasks as well as active ones, enable this option and specify the range of completed tasks you want shown. For example, if you want to see all tasks completed within the past week, choose 7 days.
 - **Compare Editor**
 - **Reuse compare editor:** If want Code Review to open each revision in the same editor, enable this option.
 - **Close compare editors on commit review action:** If you want the compare editor to be closed when a revision is committed, enable this option.
 - **Show structural changes:** If you want the compare editor to show structural changes, enable this option.
 - **Show suppressed parts:** If you want the compare editor to show details from code excluded from analysis (as described in [Hiding Differences for Specific Pieces of Code](#)), enable this option.
 - **Show the newer source on the left | right:** Specifies where the most recent source is positioned in the comparison editor.
 - **Opening Local Sources in Existing Projects**
 - **Use source control to recognize local sources:** If you want to allow Parasoft Test to use source control data in order to better recognize the sources found, enable this option.
 - **Always open without asking when path of a local source is different than remote path (single source matches only):** If you want to force Parasoft test to always apply a local source path (if it is different than the remote one), enable this option.
 - **Show warning when the file has been changed since issue was created:** If you want to receive warnings if the Code Review task becomes outdated (i.e., because the current source code has been modified since the Code Review task was created), enable this option.
 - **Appearance:** Allows you to customize what labels are used in the code review task tree and determine whether tasks for multiple revisions of a file are merged.
6. (Desktop installations only) Select the **Parasoft> Code Review> Team** category in the left pane, then complete the Code Review settings as follows:
 - **Workflow:** Indicate whether you want to use the default workflow or the restricted workflow. For an overview of these options, see [Workflow Overview](#). If you prefer, you can enable a restricted workflow directly on the Team Server: Use the property key path `/usr/{user}/codereview/workflow`, and the property value `restrict`. If you later want to revert to a default workflow policy, change the property value to `default`.
 - **Import tasks from Team Server to Report Center:** Optimizes performance by storing network data in the DTP database instead of in flat files on Team Server. See [Importing Code Review Data from Team Server to the DTP Report Center Database](#) for details.
 - **Task categories:** This allows you to customize the list of categories that reviewers can use to categorize their comments.

Migrating Data Previously Stored on Team Server

Team Server storage is deprecated. To migrate data from Team Server to the current storage method (Report Center):

- a. Open the TCM Data Storage page in a web browser and go to the root folder for the tcm user account that was being used for code review.
- b. In the 'codereview' subfolder, set or change the value of the 'storage.type' property to 'db.storage.type'.
- c. Restart the IDE.
- d. In the preferences page described here (**Preferences > Parasoft > Code Review > Team**) click the **Import** button to transfer data from Team Server to Report Center.

7. Click **Apply** to apply your settings.
8. Click **OK** to set and save your settings.

Configuring Test Configurations for Scanning

Test Configurations control how code reviews are prepared, and the appropriate configuration varies depending on whether your team has a pre-commit or post-commit code review process.

- For details on configuring Test Configurations for **pre-commit** code review processes, see [Configuring and Running Pre-Commit Code Review Scans](#).
- For details on configuring Test Configurations for **post-commit** code review processes, see [Configuring and Running Post-Commit Code Review Scans](#).

Hiding Differences for Specific Pieces of Code

If you do not want the compare editor to show diffs that occur within specific blocks of code:

1. Add the `codereview-begin-suppress` comment at the point in the code where you want to stop showing differences.
2. Add the `codereview-end-suppress` comment at the point in the code where you want to resume reporting of differences.

By default, suppressed code will not be shown in diffs for related code review tasks (for example, a diff highlighting a code change immediately following or preceding suppressed code). If you later want to view the suppressed content in diffs:

1. Choose **Parasoft > Preferences** to open the Preferences dialog.
2. Select the **Parasoft > Code Review** category in the left pane.
3. Enable the **Show suppressed parts** option.

Ignoring Changes in within Suppress Comments

If you want the code review scan to ignore all differences that occur between "codereview-begin-suppress" and "codereview-end-suppress" markers, enable **Ignore changes within suppressed blocks** in the **Code Review > Filters** tab.

Troubleshooting Unpublished or Skipped Reviews

Unpublished Reviews

If the Test Progress tab reports that reviews were not published, this is a sign that your configuration has undefined authors.

For example, assume that your code review results are as follows:

Code Review

Files scanned: 2/2
Files skipped: 0
Locally removed: 0
Revisions accepted: 2/2
Revisions skipped: 0
Undefined authors: pietrek
Reviews to publish: 0/0
Reviews skipped: 0

0:00:01 

In this case, 2 files have been scanned and 2 revisions have been accepted—but there is no code review package to publish, because the scanned code had modifications made by "pietrek"—and the Test Configuration does not list pietrek in the **Code Review > Authors** tab.

To correct this problem:

1. Add the author to the Test Configuration's **Code Review > Authors** tab.
2. Either assign a reviewer to review all code by this author, or assign a reviewer to the area of source code where this author's revisions occur.
3. Run the Test Configuration again.

Skipped Reviews

If the results show that reviews were skipped (with undefined users and/or unmatched paths by author), this is a sign that the modified code is not properly assigned to a reviewer.

For example, assume that you have the following results

Code Review

Files scanned: 2/2
Files skipped: 0
Locally removed: 0
Revisions accepted: 2/2
Revisions skipped: 0
Undefined authors: 0
Reviews to publish: 0/1
Reviews skipped: 1
- no reviewers assigned: 1, authors: pietrek

0:00:12 

and that your report indicates the following:

12/04/08 13:36:19	Results from: Post Commit	
CODEREVIEW		
Author		cre
Unmatched Paths by Author		
pietrek: \com.parasoft.xtext.results.ap\src\com\parasoft\xtest\results\api		

You can resolve the problem by modifying the Test Configuration to either assign that area of code to a reviewer, or to assign a reviewer to all code modified by that author.

Understanding How Code Review Packages are Created

Each code review package is defined based on the author changes (commits).

- In post-commit code review, changes are collected when the automated scan runs each day.
- In pre-commit code review, the code author specifies which changes should be included in the package.

Any reviewers and monitors mapped to that author are then assigned to the package. If the package contains code from the project areas that other reviewers or monitors are set to review, those reviewers/monitors are also assigned to this package.

If a package is created and no reviewers or monitors are set to review the related author or project area, then no review tasks will be generated for that package.

Separate packages are not created for different paths (regions). As a result, reviewers assigned to different paths may be working on the same package. They should decide for themselves which paths they want to review. This maintains the integrity of tasks that may spread across multiple reviewers' assigned areas. If a reviewer wants to see which files in a package were matched to him because they belong to "his" assigned region, this information is available in the Code Review view.

Importing Code Review Data from Team Server to the DTP Report Center Database

It is more efficient to store network data in the DTP Report Center database instead of in flat files on Team Server. Team Server storage is supported for backward compatibility, but Report Center storage is strongly recommended.

To migrate code review data from Team Server to Report Center:

1. Choose **Parasoft> Preferences**.
2. Open **Parasoft> Code Review> Team**.
3. Click **Import**, then complete the dialog that opens. This allows you to import all code review data or only packages from specified data range. If existing items are imported, they will be duplicated.

The average time needed to import one Code Review package from Team Server to the DTP database is approximately 30 seconds. A safe estimate is to assume that you can import about 2880 packages per day.