

# Connecting to Source Control

This topic explains how to connect C++test to your source control repository.

Sections include:

- [About C++test's Source Control Support](#)
- [Enabling Source Control Support](#)
- [AccuRev Configuration](#)
- [ClearCase Configuration](#)
- [CVS Configuration](#)
- [Git Configuration](#)
- [Perforce Configuration](#)
- [Serena Dimensions Configuration](#)
- [StarTeam Configuration](#)
- [Subversion Configuration](#)
- [Synergy Configuration](#)
- [Team Foundation Server Configuration](#)
- [Visual SourceSafe Configuration](#)
- [Specifying Source Control Definitions via localsettings](#)

## About C++test's Source Control Support

Any source control system that plugs into your C++test environment can be used to manage your source and test files.

If your team is using a one of the specified supported source control system (see list below) and performs the necessary configurations (as described later in this topic), Parasoft can:

- Use file revision data from source control to determine authorship (for automatically assigning test failures and policy violations to the responsible team member as well as for restricting test scope by author and/or modification time). See [Configuring Task Assignment and Code Authorship Settings](#) for details.
- Update projects from source control before testing (if the Test **Configuration's Common**> **Source Control**> **Update projects** setting is enabled).
- Automate the preparation and tracking of code reviews (see [Code Review](#)).

Parasoft currently supports the following source control system.

- **AccuRev** - tested on 4.6, 5.x
- **ClearCase** - tested on 2003.06.00
- **CVS** - tested on 1.1.2
- **Git** - tested on version 1.7+
- **Perforce** - tested on 2006.2, 2015
- **Serena Dimensions** - tested on 9.1, 10.3 (2009 R2), 12.2
- **StarTeam** - tested on 2005, 2008, 2009
- **Subversion (SVN)** - tested on 1.2.x, 1.3.x, 1.4.x, 1.5.x, 1.6.x, 1.7.x, 1.8.x
- **Synergy** - tested on 6.4, 7.0, 7.1
- **Team Foundation Server** - tested on 2010, 2012, 2015
- **Visual SourceSafe** - tested on 5.0, 6.0 and 2005

In addition, you can use Parasoft's source control API to add seamless integration with any other source control tool your team happens to be using. For details, see [Adding a Custom Source Control Integration](#).



### Subclipse Support Notes

- Each Subclipse plugin version is compatible with specific Subversion versions. Ensure that your Subclipse plugin is compatible with a version of Subversion that Parasoft supports. For example, you should not install Subversion 1.3 and Subclipse plugin 1.2, which uses Subversion 1.4. Subclipse 1.4.x requires Subversion 1.5.0 version of JavaHL/SVNKit. Subclipse 1.4.5 already has subversion client adapter 1.5.2.
- Due to changes introduced in Subversion 1.4, Subversion clients earlier than 1.4 cannot work with working copies produced by Subversion 1.4. If you are using Subclipse plugin 1.2 (which includes Subversion 1.4), you might receive the following error message:  
svn: This client is too old to work with working copy '.'; please get a newer Subversion client  
This means that Parasoft is using a command-line client that is version 1.3 or older. The solution is to update your command-line SVN client to version 1.4. The client version can be verified by executing `svn --version`

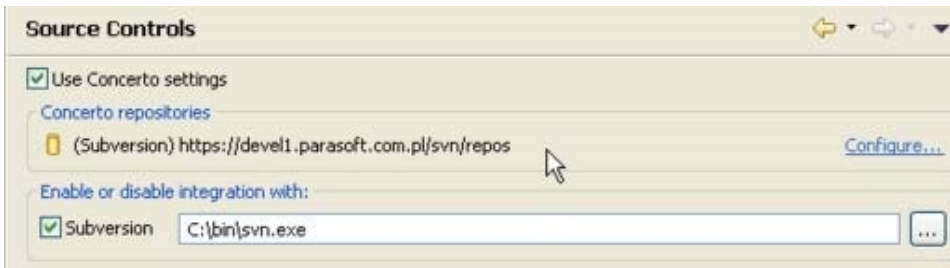
### Why do dialogs open when I try to modify files?

Some source controls (including ClearCase, Perforce, Synergy, and Visual SourceSafe) require users to mark (lock) sources before editing them. If you are using one of these source control systems and you prompt C++test to perform an operation that involves editing a "read-only" file in source control, it will first open a dialog asking you whether you want to make the file writable and lock it. Click **OK**, then provide your source control username and password in the next dialog that opens; this allows it to access the source control system and set the lock.

# Enabling Source Control Support

To enable support for any of the supported source control systems:

1. Make sure that the command line client for the given source control is on the system%PATH%/\$PATH and is available when C++test is launched.
  - For example, if you have Subversion, it is not sufficient (or even required) to install the Subclipse plugin to Eclipse (SVN Eclipse plugin). Instead, you should have the plain command line svn.exe Subversion client.
2. Choose **Parasoft> Preferences**. The Preferences dialog will open.
3. Select **Parasoft> Scope and Authorship** in the Preferences dialog.
4. Check **Use source control (modification author) to compute scope**.
5. Select **Parasoft> Source Controls** in the Preferences dialog.
6. If the appropriate repositories ARE already set (from the auto-configuration process described in [C++test Configuration Overview](#)), enter your user and password, and specify the path to your source control client executable (if it is not already on your system path).



7. If the appropriate repositories ARE NOT already set (from the auto-configuration process described in [C++test Configuration Overview](#)), specify them as follows:
  - a. Enable the check box for the source control system you want to use.
  - b. If the source control client executable is not already on your system path, specify the path to it in the text field to the right of the source control system's name.
  - c. Enter the source control properties required for the selected type of source control system by clicking **New** in the Defined Source Controls table, completing the Create Source Control Description dialog's fields as appropriate for your system, then clicking **OK**.
    - Note that the fields in the Create Source Control Description dialog are described in the following sections.
  - d. Click **OK** to close the Source Control Description dialog.
8. Click **Apply**, then **OK**.

To test the integration:

1. In the Parasoft environment, open a project that is checked out from the repository.
2. Open a file in the editor.
3. Right-click the source code, and choose **Parasoft> Show Author at Line**. If the correct author is shown, the integration was successful.

## Debugging Tip

To troubleshooting problems with source control integration, run `-consolelog -J-Dcom.parasoft.xtest.logging.config.jar.file=/com/parasoft/xtest/logging/log4j/config/logging.on.xml`. This should result in detailed log information being printed to the console.

To include messages from the source control system that may contain fragments of user source code, use an additional flag: `-Dscontrol.log=true`

# AccuRev Configuration

When you are enabling source control support, specify the following repository properties in the Create Source Control Description dialog:

- **Server:** Enter the hostname of server where AccuRev is running (required).
- **Port:** Enter the port of the server where AccuRev is running (required).
- **Username:** Enter the AccuRev username/login (required).
- **Password:** Enter the AccuRev password (if needed).

# ClearCase Configuration

To use ClearCase with C++test:

- Check whether a source file is controlled by ClearCase by calling the `cleartool describe -fmt %N <file_path>` command. No output means that the file is not controlled by ClearCase.
- Ensure that the VOB root directory contains a `lost+found` directory.

When you are enabling source control support, specify the following repository properties in the Create Source Control Description dialog:

- **VOB location:** Enter the dynamic or snapshot VOB access path. Use the full VOB path (e.g., /vobs/myvob (Linux dynamic view) or M:\my\_dynamic\_view\myvob (Windows VOB path). Note that when you enter a Vob Location, the Vob tag field will automatically display the vob tag. If the location is not a proper vob path, a warning message is displayed

## CVS Configuration

To use CVS with C++test, ensure that the .cvspass file is in one of the following locations:

- user.home system property
- HOME env variable
- (Windows) combination of HOMEDRIVE and HOMEPATH (example: "C:" + "\home")
- current working directory

When you are enabling source control support, specify the following repository properties in the Create Source Control Description dialog:

### General tab

- **Connection type:** Enter the authentication protocol of the CVS server.
- **User:** Enter the user name under which you want to connect to the repository.
- **Password:** Enter the password for the above user name.
- **Repository path:** Enter the path to the repository on the server.
- **Server:** Enter the CVS server's machine name or IP address.
- **Port:** Enter the CVS server's port.

### Custom SSH/CVS\_RSH tab

- **CVS\_SERVER value:** If connecting to a CVS server in EXT mode, this specifies which CVS application to start on the server side.
- **Use custom authentication properties for ext/server method:** Enable this option if you want to use custom authentication for ext/server method.
- **Remote shell login:** Enter your SSH login.
- **Remote shell password:** Enter the password for the above SSH login.
- **Private key file:** Enter the private key file.
- **Passphrase for private key file:** Enter the passphrase for the above private key file.
- **Use command-line program to establish connection:** Enables you to run an external program to establish an EXT connection. Use this option only for non-standard and legacy protocol connections (telnet, rsh). Linux/Unix/Cygwin ssh prompts for passwords/passphrases/ security word sequences are not currently supported.
- **CVS\_RSH path:** Specifies the full path to the executable used to establish EXT connections.
- **CVS\_RSH parameters:** Specifies the parameters for the executable. The following macro-definitions (case sensitive) can be used to expand values into command line parameters:
  - {host} - host parameter
  - {port} - port parameter
  - {user} - user parameter from primary page
  - {password} - user password from primary page
  - {extuser} - user parameter from EXT/CVS\_RSH page
  - {extpassword} - password parameter from EXT/CVS\_RSH page
  - {keyfile} - path to key file
  - {passphrase} - password to key file

## Git Configuration

To use Git with C++test:

1. git clone <repository>. Note that the repository must allow anonymous pulls.
2. If cloning from a git:// url, then the git push url must be configured if you want to be able to have git automatically push generated test cases that get checked in.
3. Set git config on the newly created repository user.name (this step can be skipped if a global git user.name is already set). Note that this user.name must match the user's DTP user name.

When you are enabling source control support, specify the following repository properties in the Create Source Control Description dialog:

- **URL:** Specify the remote repository URL to pull/push to. If pulling and pushing is disabled, this field can be left blank. You can use all git urls (e.g., git://host/repositoryssh://user@host user@host). Note that since the Git command line doesn't support setting credentials with flags, protocols which normally require authentication (such as ssh) must have identity files set to enable login without credentials or they must accept anonymous connections.
- **Branch:** Enter the name of the branch in the local workspace that the source control module will use. If this is left blank, the currently checked out branch will be used.
- **Working folder:** Enter the root of the local git repository.

## Perforce Configuration

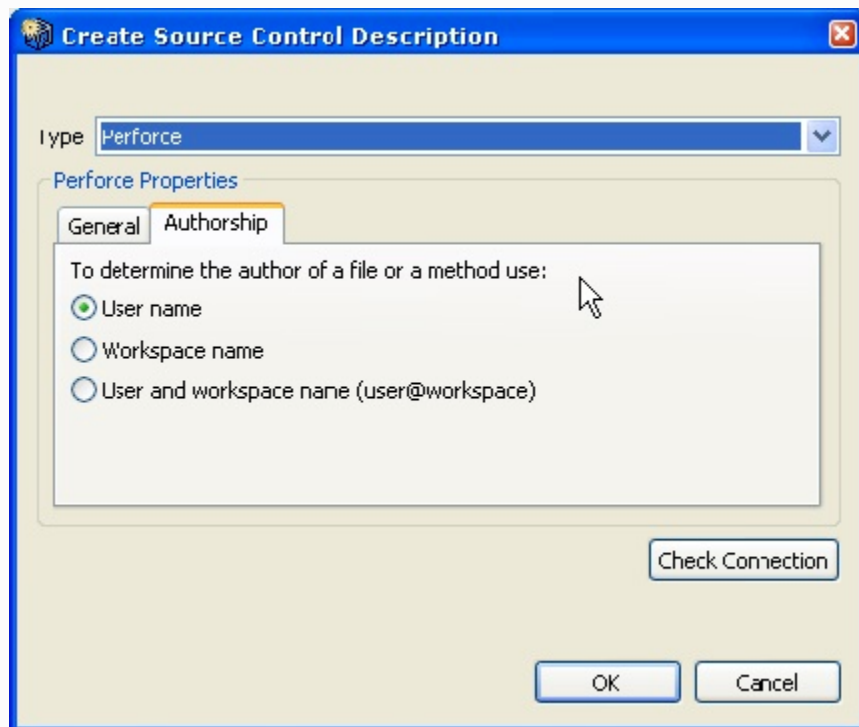
When you are enabling source control support, specify the following repository properties in the Create Source Control Description dialog:

- **Server:** Enter the Perforce server's machine name or IP address.
- **Port:** Enter the Perforce server's port.
- **User:** Enter the user name under which you want to connect to the repository.
- **Password:** Enter the password for the above user name.
- **Client:** Enter the client workspace name, as specified in the P4CLIENT environment variable or its equivalent.

### Using Workspaces to Determine Authorship

By default, username is used to determine file/method authorship. However, some teams access Perforce with a shared user name and a unique workspace for each developer.

If you want to use workspace name (or user name and workspace name) to determine authorship, open the **Authorship** tab and modify the setting as needed.



## Serena Dimensions Configuration

### Linux Configuration Note

To use Serena Dimensions with C++test, Linux users should run C++test in an environment prepared for using Serena programs, such as 'dmcli'

- LD\_LIBRARY\_PATH should contain the path to <SERENA Install Dir>/libs.
- DM\_HOME should be specified.

When you are enabling source control support, specify the repository properties in the following tabs.

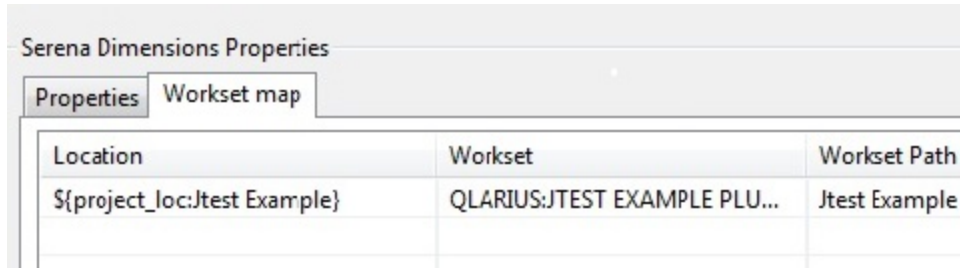
### Properties tab

- **User:** Enter the Serena Dimensions login name.
- **Password:** Enter the Serena Dimensions password.
- **Server:** Enter the Serena Dimensions server host name.
- **Database name:** Enter the name of the database for the product you are working with.
- **Database connection:** Enter the connection string for that database.
- **Server locale:** Select the language.

### Workset Map tab

If the project has been downloaded/moved to a location other than default work area, use this tab to specify a mapping between the project (or stream) with the Serena repository and the local project.

If you are working in the default work area, you do not need to define mappings.



Location	Workset	Workset Path
<code>\${project_loc}\Jtest Example</code>	QLARIUS:JTEST EXAMPLE PLU...	Jtest Example

Enter an expression that maps workspace resources to Serena Dimension repository paths.

- Example 1: If you use `scontrol.rep.serena.mapping_1=${project_loc}\MyProject`;PRODUCT1\ WORKSET1;src\MyProject, then Project 'MyProject' will be mapped to the Serena workset PRODUCT1:WORKSET1 and workset relative path: src\MyProject
- Example 2: If you use `scontrol.rep.serena.mapping_2=${workspace_loc}`;PRODUCT1\WORKSET1 then the complete workspace will be mapped to the Serena workset PRODUCT1:WORKSET1.

## StarTeam Configuration

To use StarTeam with C++test ensure that:

- Ensure that you have the Borland StarTeam SDK installed. This can be downloaded for free from the Borland web site.

For Visual Studio plugins, Parasoft StarTeam support is available for VS 2005 and VS 2008. When you are enabling source control support, specify the following repository properties:

- **Server:** Enter the StarTeam server's machine name or IP address.
- **Port:** Enter the StarTeam server's port.
- **User:** Enter the user name under which you want to connect to the repository.
- **Password:** Enter the password for the above user name.
- **Paths lookup> StarTeam Path:** When working with large multi-project repositories, you can improve performance by specifying the project, view, or folder that you are currently working with. You can indicate either a simple Project name (all views will be scanned when searching for the repository path), a Project/View (only the given view will be scanned) or Project/View/Folder (only the specified StarTeam folder will be scanned).
- **Paths lookup> Working Folder:** If the **StarTeam Path** setting specifies a StarTeam view or folder, you can use this field to indicate a new working directory for the selected view's root folder (if the path represents a view) or a new working directory for the selected folder (if the path represents a folder).

## Subversion Configuration

Parasoft's Subversion support is based on the command line client 'svn'. To use Subversion with C++test, ensure that:

- The Subversion 1.2.x, 1.3.x, 1.4.x, 1.5.x, 1.6.x, 1.7.x, or 1.8.x client is installed.
- The client certificate is stored in the Subversion configuration area. The Subversion client has a built-in system for caching authentication credentials on disk. By default, whenever the command-line client successfully authenticates itself to a server, it saves the credentials in the user's private runtime configuration area—in `~/.subversion/auth/` on Unix-like systems or `%APPDATA%\Subversion/auth/` on Windows.

When you are enabling source control support, specify the following repository properties in the Create Source Control Description dialog:

- **URL:** Enter the URL for the SVN server. The URL should specify the protocol, server name, port and starting repository path (for example, `svn://buildmachine.foobar.com/home/svn`).
- **User:** Enter the user name under which you want to connect to the repository.
- **Password:** Enter the password (not encoded) for the above user name.

## Synergy Configuration

Synergy 7.0 introduces a new, faster way of working called Web mode. C++test supports both of the following Synergy/CM communication modes:

- Web mode (since Synergy 7.0)
- Traditional mode

When you are enabling source control support, specify the following repository properties in the Create Source Control Description dialog:

- **Database path:** Enter the absolute Synergy database path.

- **Engine host:** Enter the Synergy server's machine name or IP address. *For Web mode, the host must be a valid Synergy Web URL with protocol and port (e.g., `http://synergy.server:8400`).*
- **User:** Enter the user name under which you want to connect to the repository.
- **Password:** Enter the password for the above user name.
- **Use remote client (UNIX systems only):** Enable this option if you want to start CCM as a remote client session. *This is not used for Web mode.*
- **Local database (remote client):** Enter the path to the location where the database information is copied when you are running a remote client session. *This is not used for Web mode.*

## Team Foundation Server Configuration

C++test works with TFS 2008, 2010, 2012, 2013, 2015C++test for Eclipse doesn't require any additional software to be installed for TFS integration (the required libraries are included).

When you are enabling source control support, specify the following repository properties in the Create Source Control Description dialog:

- **URL:** Enter the URL for the Team Foundation Server repository (for example, `http://localhost:8080/tfs`).
- **Use custom credentials:** If you want to provide custom TFS credentials, select this option then provide those credentials. When this option is disabled, system credentials will be used.
- **User:** Enter the user name.
- **Password:** Enter the password.

By default, C++test uses the cached credentials for accessing TFS (this could be your user login or some previously logged in information). You can provide custom credentials if you want to use them instead of the cached ones.

## Visual SourceSafe Configuration

When you are enabling source control support, specify the following repository properties in the Create Source Control Description dialog:

- **VSS Database Path:** Enter the database path (the location of SRCSAFE.INI).
- **User:** Enter the user name under which you want to connect to the repository.
- **Password:** Enter the password for the above user name.
- **Project root in repository:** Enter the project root. This begins with `$/`; for example, `$/nightly_test`.

## Specifying Source Control Definitions via localsettings

Source control definitions can be specified in localsettings (e.g. for sharing team-wide settings via DTP or specifying options at the command line). See [Configuring Localsettings](#) for details.