Preconfiguration C++test During Installation

For details on how to preconfigure C++test during installation to streamline consistent installation across multiple team machines, see C++test Configuration Overview.

In addition, you can use silent installation to streamline and standardize the Parasoft C++test installation and setup process across all team machines. During a silent installation, C++test is automatically installed in the designated directory, without displaying the wizard pages (for Windows) or requiring you to respond to any installation prompts or questions.

This topic explains how you can preconfigure C++test during installation to streamline consistent installation across multiple team machines. This process applies to all supported platforms and IDEs.

Sections include:

Preconfiguring the Installation

To preconfigure C++test during installation:

1. Create a localsettings file that includes the options you want to preconfigure the installation with. An example is provided below. For more details, see Configuring Localsettings.
2. Pass that localsettings file to the installer using the `/configure` or `--configure` parameter. For example:
    - Windows: `/configure=<path-to-localsettings-file>`
    - Unix: `--configure <path-to-localsettings-file>`

The localsettings file will be copied into the root directory of Parasoft Test as `parasofttest.ini`. Settings from that file are used to initialize a clean workspace during startup in UI mode.

> **Tip - Other Ways to Configure Licenses**
>
> You can also configure licensing upon command line startup using -localsettings. For details, see Licensing.

If you want to apply settings to an existing workspace, you need to add an additional property to the localsettings file: `enforce.configure=true`. This overrides existing settings during the first startup after the installation/reinstallation.

Note that with the Windows installer:

- If the same version (or a newer version) of Parasoft Test is installed on the machine, the localsettings file won't be copied during installation or reinstallation.
- If the location passed in `/ParasoftTestDir` is different than the location of an existing C++test installation, the localsettings file won't be copied during reinstallation.

Windows Options

Option	Description
`/SILENT` `/VERYSILENT`	Run installer in silent mode. The wizard and the background window are not displayed.
`/DIR=<location>`	Destination location for installed product's files.
`/EclipseDir=<location>`	Location of target platform. *Eclipse plug-in installer only.*
`/RootSuffix=<suffix>`	Root suffix. *VS installer only/*
`/Configure=<localsettings file>`	Preconfigure product during installation step.
`/ParasoftTestDir=<location>`	Destination location for Parasoft Test's files (Use this option if Parasoft Test is not already installed). *Supported by all products installers except Parasoft Test.*

See http://www.jrsoftware.org/ishelp/index.php?topic=setupcmdline for additional options and help.

UNIX Options (for Linux or Mac)

Option	Description
`--non-interactive`	Run installer in non-interactive mode.
`--target-location <location>`	Location of target platform (e.g., Eclipse). *Eclipse plug-in installer only.*
`--configure <localsettings file>`	Preconfigure product during installation step.
`--parasofttest-install-dir <location>`	Destination location for Parasoft Test's files. *Supported by all products installers except Parasoft Test.*
`--help`	Prints the available command line options

Installation Examples

Windows

```
parasoft_cpptest_desktop_10.3.2_win32.exe /SILENT /DIR=<cpptest dir> /ParasoftTestDir=<parasofttest dir>
/configure=<localsettings file>
parasoft_cpptest_desktop_10.3.2_win32_eclipse.exe /SILENT /DIR=<cpptest dir> /ParasoftTestDir=<parasofttest dir>
/EclipseDir=<eclipse dir> /configure=<localsettings file>
```

Unix

```
cpptest_10.3.2.169_linux.sh --non-interactive --parasofttest-install-dir <parasofttest dir> --
configure=<localsettings file>
cpptest_10.3.2.169_linux_eclipse_plugin.sh --non-interactive --parasofttest-install-dir <parasofttest dir> --
target-location <eclipse dir> --configure=<localsettings file>
```

Sample localsettings File

Here is a sample localsettings file that configures Team Server and License:

```
cpptest.license.use_network=true

cpptest.license.network.host=main1.parasoft.com.pl

cpptest.license.network.port=2222

cpptest.license.network.edition=server_edition

tcm.server.accountLogin=true

tcm.server.enabled=true

tcm.server.name=main1.parasoft.com.pl

tcm.server.password=test

tcm.server.port=18888

tcm.server.username=test

enforce.configure=true
```

Installing C++test in Silent Mode on Windows

1. Execute the distribution file in the command line with one of the following options as well as any additional options you want to use (see below):

- /SILENT
- /VERYSILENT

This will switch the installer to non-interactive mode, and the setup wizard will install C++test and Parasoft Test using the default configuration.

Additional options that are available include:

2. `/DIR=<location>`: Destination location for C++test's files.
3. `/ParasoftTestDir=<location>`: Destination location for Parasoft Test's files (use this option if Parasoft Test is not already installed).

4. `/EclipseDir=<location>`: Location of target platform.

Installing C++test in Silent Mode on UNIX

1. Run `echo "\n1\n\n\n" | <C++test_installer_for_Linux> <options>`

- If you are installing the standalone version, use the `--non-interactive` option.
- If you are installing the Eclipse plugin, use the `--force --non-interactive --target-location $ECLIPSE_LOCATION` options

C++test can also check any number of custom rules that you design with the RuleWizard module. With RuleWizard, rules are created graphically (by creating a flow-chart-like representation of the rule) or automatically (by providing code that demonstrates a sample rule violation). By creating and checking custom rules, teams can verify unique project and organizational requirements, as well as prevent their most common errors from recurring.

For details on performing static code analysis, see Static Code Analysis.