

# SOAtest CLI Overview

In this section:

- [About the Command Line Interface](#)
- [Prerequisites](#)
- [Setting Up the CLI](#)

## About the Command Line Interface

SOAtest's command line interface (`soatestcli`) allows you to perform static analysis and execute tests as part of an automated nightly process. Command line mode is available for the Automation Edition of SOAtest.

`soatestcli` can send results to the Parasoft Report Center, send comprehensive reports to the team manager and to the Parasoft Team Server, and send focused reports to each team developer and tester working on the SOA project. Reports can be generated in a number of formats. Details such as reporting preferences (who should reports be sent to, how should those reports be labelled, what mail server and domain should be used, etc.) Team Server settings, Report Center settings, and email settings, license settings, etc. can be controlled by local settings files.

The optimal team configuration is to have one SOAtest (Automation Edition) on the team build machine, SOAtest (Desktop Edition) on every developer /tester workstation, one SOAtest (Architect Edition) on the architect's machine, and one installation of Team Server on the team build machine or another team machine.

Engineers use their local installations of SOAtest to develop and run tests, then check their work in to source control. Every night, `soatestcli` runs on a team machine. Depending on the configuration, it may execute the available tests, monitor policy adherence, and/or perform the specified static analysis tests. After the test completes, engineers can import test result into the SOAtest GUI to facilitate task review and resolution.

Additionally, SOAtest sends results to the Parasoft Report Center, emails each team member a report that contains only the tasks assigned to him or her, emails group managers a report that lists all team/project quality tasks and identifies which team member is responsible for each task, and uploads reports and results to Team Server.

Throughout the process, Team Server manages the sharing and updating of test settings and test files; this standardizes tests across the team and helps team members leverage one another's work. The standardized test settings and custom team rules are configured and maintained by the team architect, who is using SOAtest Architect Edition.

## Prerequisites

- The command line mode requires a command line interface license (available with SOAtest Automation Edition).
- Before you can run a test from the command line, you need to setup a project, .tst file, and test suites. See [Adding Projects, .tst files, and Test Suites](#) for details.
- We strongly recommend that you configure DTP, Team Server, reporting, and other Parasoft preferences as described in [Configuration](#) before you start testing.

## Setting Up the CLI

Parasoft SOAtest has two user modes: interactive desktop usage in the GUI and command line mode via the command line interface (CLI). The CLI interface is a standard feature of the Automation Edition.

CLI mode is typically used to perform regular or continuous code analysis and test in conjunction with regular/continuous builds or as a part of an automatic regression test infrastructure. As part of the CLI execution, SOAtest can perform one or more of the following:

- Execution of functional, security, and regression tests
- Policy enforcement
- Static analysis
- Analysis of SCM code repository to identify code changes since the last run and initiate code review sessions on updated code.
- Generation of reports and their distribution to a central report server and/or to individual developers and managers, according to specified reporting configurations.

As part of the execution, SOAtest can use your SCM client (if supported) to automatically retrieve file modification information from the SCM system and generate tasks for specific individuals based on results of code analysis and executed tests.

Specific execution options for SOAtest are controlled via Test Configurations and Preferences.

**Test Configurations** can be sourced from the built in set, or created using SOAtest interactive mode in the GUI. We suggest using the built in configurations as starting templates for customer-specific configurations, which are then stored on disk or on Parasoft Team Server (part of the Parasoft Concerto deployment infrastructure).

**Preferences** can be configured from the SOAtest GUI. Most of the preference settings can also be supplied with a configuration file that is provided as a parameter to a CLI call. A table of the configuration file preference settings is available in [Local Settings Files - Options](#). SOAtest preferences set from the GUI are applied by default. These can be overridden—on an individual basis—by preference values contained in the configuration file used with a given run. This enables you to have a basic set of preferences configured for all CLI runs, and then vary individual settings as necessary by providing an additional configuration file for a specific run with a given Test Configuration. This can be useful, for example, to include different information in reports for different runs, or to change options for email distribution of reports, including report names, email headings, etc.

## Step 1: Configure Preferences

Preferences are accessed through the **Parasoft> Preferences** menu. Start by configuring the following preferences within the **Parasoft** group:

- **License:** Specify the license or License Sever settings.
- **Team:** Check **Enable Team Server**. If Team Server is not autodetected, enter the Team Server's IP address in **Server Information> Host Name**. If you are running Team Server on the same machine as your Automation Edition product, enter localhost. Unless you changed the Team Server default port (18888) when it was installed, do not change the port here. Click **Test Connection** to verify the correct settings.
- **Source Controls:** These settings enable automatic mapping of the tool results to the individuals who last changed the affected code or test artifact. Check your source control system, and use the instructions in [Connecting to Your Source Control Repository](#) to set the options appropriate for your SCM.
- **Scope and Authorship:** Check the appropriate options for your environment as described in [Configuring Task Assignment and Code Authorship Settings](#).
- **Reports:** The following options are enabled by default and are a good starting point:
  - **Detailed report for developers** (includes task breakdown with details).
  - **Overview of tasks by authors** (summary table).
  - **Generate formatted reports in command line mode.**
  - **Suppressions Details** (applies to static analysis only).
- **E-mails:** Enter settings that will be used to send emails with reports. This needs to be an existing email account on an email server accessible from the SOAtest test machine.
- **Reports> Email Notifications:**
  - If desired, enable **Send Reports by Email**. Regardless of this setting, reports will always be uploaded to Parasoft Team Server for later viewing (controlled by the CLI option). Email distribution will use the settings for E-mails above.
  - Manager reports contain a rollup of all test results generated by SOAtest. Developer/tester reports contain only results for individual developers and testers. Enable options and specify email addresses accordingly.

## Step 2: Customize Test Configurations

Create a custom Test Configuration as described in [Configuring Test Configurations and Rules for Policies](#). See [SOAtest Test Configuration Settings](#) for details on SOAtest-specific options.

## Step 3: Create a localsettings File

Create a localsettings file as described in [Local Settings Files - Options](#).

## Step 4: Activate CLI in the Currently-Running Build System (e.g., batch script)

For example, a sample command line to be added might be:

- `soatestcli -resource <project name> -config "team://Team Configuration" -publish -localsettings acme_policy.settings`

The reports will be sent after each batch run, and trend reports will be populated with data. The reports will also be available for viewing via **Parasoft> Explore> Team Server Reports**.