

Configuring Virtualize Server

Most configuration settings for the server are specified in the config.properties file located in the <SOAVIRT_HOME>/WEB-INF directory. The settings in this chapter are the most commonly-specified options, but you can specify more advanced settings, as well as test configuration settings. See the following sections:

- [Advanced Server Configuration Options](#)
- [Test Configuration Options for SOAtest and Virtualize Server](#)



Use Forward Slashes, Not Backslashes

When editing Windows file paths in config.properties, be sure to use forward slashes, not backslashes.

Correct: c:/my/workspace

Incorrect: c:\my\workspace

In this section:

- [Base Configuration Options](#)
- [Report Settings](#)
- [Adding Custom/External Jars](#)

Base Configuration Options

- [parasoft.eula.accepted](#)
- [working.dir](#)
- [logging.config.file](#)
- [scripting.timeout.minutes](#)
- [server.hostname](#)
- [server.port.http](#)
- [server.port.https](#)
- [env.manager.server](#)
- [env.manager.server.name](#)
- [env.manager.username](#)
- [env.manager.password](#)
- [env.manager.notify](#)
- [env.manager.registry.id](#)
- [soatest.license.enabled](#)
- [virtualize.license.enabled](#)
- [soatest.license.use_network](#)
- [soatest.license.network.edition](#)
- [soatest.license.custom_edition_features](#)
- [virtualize.license.use_network](#)
- [virtualize.license.network.edition](#)
- [virtualize.license.custom_edition_features](#)
- [dtp.server](#)
- [dtp.port](#)
- [dtp.user](#)
- [dtp.password](#)
- [license.network.host](#)
- [license.network.port](#)
- [license.network.user](#)
- [license.network.password](#)
- [license.network.use.specified.server](#)
- [license.network.connection.type](#)
- [license.network.auth.enabled](#)
- [virtualize.license.local.password](#)
- [virtualize.license.local.expiration](#)
- [usage.reporting.enabled](#)

parasoft.eula.accepted

This option sets the property to accept the Parasoft End User License Agreement (EULA). The default is false. You must accept the EULA to use the product. Please read the EULA at <https://www.parasoft.com/license>.

Example Usage

```
parasoft.eula.accepted=true
```

working.dir

This option specifies an absolute path to the SOAVIRT working directory. If not specified, a default working directory called 'workspace' is created in the <SOAVIRT_HOME> folder. We recommend changing the default directory so that the workspace is not under the <SOAVIRT_HOME> folder because upgrading to a newer version of the server could overwrite the workspace.

logging.config.file

This option changes the logging level using a built-in or custom log4j2 configuration. If not specified, the default value is /WEB-INF/default.logging.xml.

You can specify a configuration using an absolute path or a relative path. The root for relative paths is the <SOAVIRT_HOME> directory location. The server ships with two built-in configurations available in the /WEB-INF/ folder:

- default.logging.xml: A standard log4j2 configuration that logs INFO, WARN, and ERROR messages.
- debug.logging.xml: A standard log4j2 configuration that provides verbose logging with extended details.

You can modify these built-in configurations as needed; see the [log4j2 documentation](#) for details.

Example Usage

```
logging.config.file=/WEB-INF/default.logging.xml
```

If you unzipped the WAR into c:/wars/soavirt, the server would resolve the relative location of logging.config.file to the absolute path c:/wars/soavirt/WEB-INF/default.logging.xml.

```
logging.config.file=d:/MyConfigurations/default.logging.xml
```

scripting.timeout.minutes

This option specifies the number of minutes after which Virtualize will attempt to stop an unresponsive script and log an error message. The default is 10 minutes.

Example Usage

```
scripting.timeout.minutes=5
```

server.hostname

This options specifies the hostname that CTP will use to connect to the server. Under most circumstances, this option does not require configuration. If the server container or network infrastructure prevents the connection from being established, however, the option may be necessary if the remote host can not determine a valid way to connect back to the server. This may be the case when working with containerized environments.

server.port.http

This option tells CTP which port Parasoft Virtualize Server is listening to for HTTP. You must specify either server.port.http or server.port.https port. This value must match the HTTP port under which Parasoft Virtualize Server is deployed.

server.port.https

This option tells CTP which port Parasoft Virtualize Server is listening to for HTTPS. You must specify either server.port.https or server.port.http port. This value must match the HTTPS port under which Parasoft Virtualize Server is deployed.

env.manager.server

This option specifies the location of the CTP server. Be sure to specify the port and protocol (http/ https). Colons must be escaped with the backslash \ character.

Example Usage

```
env.manager.server=http\\://em.parasoft.com\:8080/
```

```
env.manager.server=https\\://em.parasoft.com\:9443/
```

Setting the Property with the REST API

You can also set this property after the server has been started using the REST API. Note that settings related to CTP in the .properties file are named "env.manager", which refers to the legacy name Environment Manager. The CTP object in the API, however, is labeled "ctp". These settings refer to the same component.

1. Start the server and go the endpoint for writing to the .properties file:

`http://<host>:<port>/soavirt/api/v5#!/preferences/preferencesPUT_config`

2. Click **Try it out** and enter the Continuous Testing Platform (CTP) server settings as a JSON object into the input field.

PUT /v6/preferences/config Updates Server Configuration

Updates current configuration with the given configuration.

Parameters Cancel

No parameters

Request body application/json

The new configuration

```
{
  "ctp": {
    "server": "http://emdemo.parasoft.com",
    "name": "emdemo",
    "username": "admin",
    "password": "admin",
    "notify": true
  },
  "license": {
    "licenseServer": {
      "host": "snake",
      "port": 443,
      "useSpecifiedServer": true,
      "connectionType": "http",
      "authEnabled": true,
      "username": "atujillo",
      "password": "***"
    },
    "virtualize": {
      "enabled": true,
      "useNetwork": true,
      "license": {

```

Execute Clear

3. Click **Execute** to update the configuration on the server.

You can copy the following example JSON, paste it into your input field, and change the values to your installation:

```
{
  "ctp": {
    "server": "<your-ctp-server>:<port>",
    "name": "<your-ctp-server-name>",
    "username": "<your-username>",
    "password": "<your-password>",
    "notify": false,
  }
}
```

env.manager.server.name

This option specifies the name that will be used to label this server on CTP. You can use any value that helps you identify this server.

Example Usage

```
env.manager.server.name=MyVirtServerLabel
```

Setting the Property with the REST API

You can also set this property after the server has been started using the REST API. See [env.manager.server](#) for details.

env.manager.username

This option specifies the username for logging into CTP.

Example Usage

```
env.manager.username=me
```

Setting the Property with the REST API

You can also set this property after the server has been started using the REST API. See [env.manager.server](#) for details.

env.manager.password

This option specifies the (plain text or encoded) password for logging into CTP.

We strongly recommend running a command line instance of SOAtest or Virtualize with the `-encodepass` flag to ensure successful authentication and a secure connection. Examples:

```
soatestcli.exe -encodepass <your_password>
```

```
virtualizecli.exe -encodepass <your_password>
```

The command line tool will generate an encoded string that you can use in place of the password property in the SOAtest and Virtualize Server configuration file. See [CLI Options](#) for additional information.

Example Usage

```
env.manager.password=12345
```

Setting the Property with the REST API

You can also set this property after the server has been started using the REST API. See [env.manager.server](#) for details.

env.manager.notify

This option enables/disables notifications to Parasoft CTP when virtual assets are deployed.

Example Usage

The following example enables notifications:

```
env.manager.notify=true
```

Setting the Property with the REST API

You can also set this property after the server has been started using the REST API. See [env.manager.server](#) for details.

env.manager.registry.id

This is a read-only setting used by CTP.

soatest.license.enabled

This option enables/disables the license related to SOAtest functionality. The default is `true`. If this option is not present in the `.properties` configuration file, the default is used. If this option is set to `false`, other license-related settings will be ignored.

Example Usage

```
soatest.license.enabled=true
```

virtualize.license.enabled

This option enables/disables the license related to Virtualize functionality. The default is `true`. If this option is not present in the `.properties` configuration file, the default is used. If this option is set to `false`, other license-related settings will be ignored.

Example Usage

```
virtualize.license.enabled=true
```

soatest.license.use_network

This option enables/disables licensing over a network. If you are using a network license, you must also configure either a connection to License Server or to Parasoft DTP.

Configure the following options to connect to License Server:

- `license.network.host`
- `license.network.port`

Configure the following options to connect to DTP:

- `dtp.server`
- `dtp.port`
- `dtp.user`
- `dtp.password`

Example Usage

The following example enables licensing over a network:

```
soatest.license.use_network=true
```

soatest.license.network.edition

This option specifies the type of license to retrieve from License Server or DTP. You can specify either `custom_edition` or `professional_edition`. Contact your Parasoft representative if you are unsure of which edition you should set.

Example Usage

```
soatest.license.network.edition=custom_edition
```

soatest.license.custom_edition_features

This option specifies a comma separated list of features that should be requested for a custom edition license from License Server or DTP. Contact your Parasoft representative if you are unsure of which features you should specify.

Example Usage

```
soatest.license.custom_edition_features=RuleWizard,Command Line,SOA,Web,Server API Enabled,Jtest Connect,Message Packs,Advanced Test Generation Desktop,Advanced Test Generation 100 Users
```

virtualize.license.use_network

This option enables/disables licensing over a network. If you are using a network license, you must also configure either a connection to Parasoft DTP or to License Server (deprecated).

Configure the following options to connect to License Server:

- `license.network.host`
- `license.network.port`

Configure the following options to connect to DTP:

- `dtp.server`
- `dtp.port`
- `dtp.user`
- `dtp.password`

Example Usage

The following example enables licensing over a network:

```
virtualize.license.use_network=true
```

virtualize.license.network.edition

This option specifies the type of license to retrieve from DTP or License Server. You can specify either `custom_edition` or `professional_ edition`. Contact your Parasoft representative if you are unsure of which edition you should set.

Example Usage

```
virtualize.license.network.edition=custom_edition
```

virtualize.license.custom_edition_features

This option specifies a comma separated list of features that should be requested for a custom edition license from DTP or License Server. Contact your Parasoft representative if you are unsure of which features you should specify.

Example Usage

```
virtualize.license.custom_edition_features=Virtualize,Validate,Performance,Extension Pack,Service Enabled,Unlimited Hits/Day
```

dtp.server

This option specifies the host name of the DTP server. You must also set the `virtualize.license.use_network` option to `true` (see [virtualize.license.use_network](#)).

Example Usage

```
dtp.server=main1.parasoft.com
```

dtp.port

This option specifies the port number of the DTP server. You must also set the `virtualize.license.use_network` option to `true` (see [virtualize.license.use_network](#)).

Example Usage

```
dtp.port=8080
```

dtp.user

This option specifies the user name for DTP authentication. You must also set the `virtualize.license.use_network` option to `true` (see [virtualize.license.use_network](#)).

Example Usage

```
dtp.user=admin
```

dtp.password

This option specifies the password for DTP authentication. You must also set the `virtualize.license.use_network` option to `true` (see [virtualize.license.use_network](#)).

Example Usage

```
dtp.user=mypassword
```

license.network.host

This option specifies the host name of the License Server. You must also set the `virtualize.license.use_network` option to `true` (see [virtualize.license.use_network](#)). This option is deprecated in 9.10.3 and should be replaced with the [dtp.server](#) option.

license.network.port

This option specifies the host port of the License Server. You must also set the `virtualize.license.use_network` option to `true` (see [virtualize.license.use_network](#)). This option is deprecated in 9.10.3 and should be replaced with the `ntp.port` option.

license.network.user

Specifies the username for authentication on the secondary DTP server specified with the `license.network.host` and `license.network.port` options. It requires the `license.network.use.specified.server` and `license.network.auth.enabled` options set to `true`.

Example: `license.network.user=JohnDoe`

license.network.password

Specifies the password for authentication on the secondary DTP server specified with the `license.network.host` and `license.network.port` options. It requires the `license.network.use.specified.server` and `license.network.auth.enabled` options set to `true`, and the `license.network.user` to be configured.

We strongly recommend encoding your password to ensure successful authentication and a secure connection. You can encode your password by running a command line instance of SOAtest or Virtualize with the `-encodepass` flag. The command line tool will generate an encoded string that you can use in the configuration file. Examples:

```
soatestcli.exe -encodepass <your_password>
```

```
virtualizecli.exe -encodepass <your_password>
```

The command line tool will generate an encoded string that you can use in place of the password property in the SOAtest and Virtualize Server configuration file. See [CLI Options](#) for additional information.

license.network.use.specified.server

This option enables the server to connect to either DTP and License Server or two instances of DTP simultaneously. If set to `true`, specify a license network connection type with the `license.network.connection.type` setting.

license.network.connection.type

This option specifies whether to connect to DTP/License Server over HTTP or over TCP. HTTP over a secure port (SSL) is recommended.

Default is `http`.

license.network.auth.enabled

Enables or disables authentication on the secondary DTP server specified with the `license.network.host` and `license.network.port` options.

This option requires the `license.network.use.specified.server` option to be set to `true`.

virtualize.license.local.password

This option specifies the password for the local Virtualize license. You must also set the `virtualize.license.use_network` option to `false` (see [virtualize.license.use_network](#)).

virtualize.license.local.expiration

This option specifies an expiration for the local Virtualize license. You must also set the `virtualize.license.use_network` option to `false` (see [virtualize.license.use_network](#)).

usage.reporting.enabled

This setting enables or disables collecting anonymous usage information and sending it to Parasoft. Read our privacy statement to learn more at <https://www.parasoft.com/privacy-policy>.

Example Usage

The following example enables anonymous usage data reporting:

```
usage.reporting.enabled=true
```

Report Settings

- [session.tag](#)
- [build.id](#)
- [report.format](#)
- [report.custom.extension](#)
- [report.custom.xsl.file](#)
- [report.developer_errors](#)
- [report.developer_reports](#)
- [report.authors_details](#)
- [report.testcases_details](#)
- [report.associations](#)
- [report.assoc.url.\[tag\]](#)
- [report.failed_tests_only](#)
- [report.env_details](#)
- [report.output_details](#)
- [report.test_suites_only](#)

session.tag

This setting specifies a tag for signing results from the test session. The tag can be a string, one or more variables, or a combination. Reports for different test sessions should be marked with different session tags so that similar runs can be distinguished from each other. The default is `session`.

```
tag=${config_name}
```

Example Usage

You can use the session tag to represent a specific analysis type made on a specific code branch in a specific test environment. The following configuration could identify functional tests on the master branch for different operating systems:

```
session.tag=ft_master_${os}
```

build.id

This setting specifies a build identifier used to label results. It may be unique for each build, but it may also label more than one test session executed during a specified build. The default is `${dtp_project}-${time}`.

Example Usage

The default build ID includes the name of the project in DTP and the date of the build. For example, for the ATM project, the build ID included in the report may resemble the following: ATM-2017-07-07.

The following configuration specifies the custom build ID that consists of the name of the project and the build number passed via the environmental variable `BUILD`:

```
build.id=${dtp_project}-${env_var:BUILD}
```

For the ABC project and the build number 114 on a CI server, this may resolve to ABC-114.

report.format

This setting specifies the report format. Use a comma separated list of format values to publish reports in more than one format. By default, an XML report containing the data and an HTML file for presenting the data are published. The following values are supported:

- `xml` (default)
- `html` (default)
- `pdf`
- `csv`
- `custom` (see [report.custom.extension](#) and [report.custom.xsl.file](#) for usage)

report.custom.extension

This setting specifies the file extension for a custom report. Custom formats are defined in an XSL file. The [report.format](#) option must be set to `custom` and the [report.custom.xsl.file](#) option must also be configured.

report.custom.xsl.file

This setting specifies the path to an XSL file used to create a custom report format. Use double backslashes when specifying the file path on Windows.

report.developer_errors

This setting determines if details about developer errors should be included in manager reports. Set to `true` to include developer errors in reports or set to `false` to exclude developer errors in reports. The default is `true`.

report.developer_reports

This setting determines if detailed reports for all developers should be generated in addition to the summary report for managers. Set to `true` to enable generating detailed reports for developers. Set to `false` to disable generating detailed reports for developers. The default is `true`.

report.authors_details

This setting determines if an overview of the number and type of tasks assigned to each developer should be included in the report. Set to `true` to include types and numbers of tasks assigned to each developer. Set to `false` to exclude types and numbers of tasks assigned to each developer. The default is `true`.

report.testcases_details

This setting determines if additional test case details should be included in the report. Set to `true` to include additional details about test cases. Set to `false` to exclude additional details about test cases. The default is `false`.

report.associations

This setting enables or disables showing requirements, defects, tasks, and feature requests associated with a test in the report. Set to `true` to include requirements, tasks, and feature requests in the report. Set to `false` to exclude requirements, tasks, and feature requests from the report. The default is `false`.

report.assoc.url.[tag]

This setting generates a link to an association inside the HTML report. The URL is a query string containing an [%ID%] or `#{id}` placeholder for the issue identifier. Supported tags are `pr`, `fr`, `task`, `req`, and `test`.

Example Usage

The following example creates a link to a PR in Bugzilla in the HTML report:

```
report.assoc.url.pr=http://bugzilla.company.com/show_bug.cgi?id=[%ID%]
```

report.failed_tests_only

This setting determines if only tests that failed should be included in the report. This option is only valid for functional testing tools. Set to `true` to only include tests that failed in the report. Set to `false` to include all tests in the report. The default is `false`.

report.env_details

This setting turns on additional details in the XML report about the active environment used when running a test. The default is `false`.

report.output_details

This setting turns on additional details in the XML report about output tools that were executed when running a test. The default is `false`.

report.test_suites_only

This setting determines if summary section of the report lists only top-level suites or if it displays a tree-like view of the individual tests. Set to `true` to only include top-level suites in the summary section of the report. Set to `false` to include a tree-like view of the individual tests. The default is `true`.

Adding Custom/External Jars

You can make external/custom jars (e.g., Parasoft SOAtest/Virtualize custom extensions, third-party jars, JDBC drivers, etc.) available to Parasoft Virtualize Server.

1. Add the jar(s) using one of the following methods:
 - Copy the jar(s) into the <workspace>/VirtualAssets/system_jars folder, which is where Parasoft Virtualize Server searches for custom /external jar files to load. Add this folder if it does not already exist.
 - Upload the jar(s) using the REST API upload operation (/v5/files/upload) as described in the Swagger documentation (see [API Usage](#)). The upload should specify the /VirtualAssets/system_jars as the parent folder. The /VirtualAssets/system_jars folder will be created if it did not already exist.
2. Reload the jars using one of the following methods:
 - If the jars contain Virtualize custom extensions, restart the server.
 - Otherwise, either restart the server or call the /v5/preferences/systemProperties/reload REST API.

Additional Configuration Notes

If you want to modify the value of the "Server" HTTP Header in Parasoft Virtualize Server's HTTP Response to 'Parasoft Server' (this is the default for traditional Eclipse-based Virtualize server deployments), you need to modify the servlet container configuration.

Some servlet containers reject Trace requests (i.e., HTTP requests using the TRACE HTTP method). If your servlet container rejects Trace requests but you want Parasoft Virtualize Server to support Trace requests (like traditional Eclipse-based Virtualize Server deployments do), you will need to update your servlet container configuration accordingly.