

Using T3D

T3D is a command line tool which can be called by simply typing "t3d" in Windows or on Unix systems. The Windows command-line can be accessed by starting the program "cmd". For Unix systems, we only support bash environments.

Configuration

T3D uses an XML-based configuration format. Starting with version v1.0.0, the location of the configuration file must be supplied as a command line option during execution (the location was previously based in the default user's data location - depending on the system, a file `t3d.xml` was stored in `%APPDATA%\T3D` (Windows) or in `~/.t3d/` (Unix)). The location of the `%APPDATA%` directory depended on the username, the Windows version and possibly the localization. For example, for a user 'foobar' on a German Windows XP machine, the resulting configuration path would be `C:\Documents and Settings\foobar\Application Data\T3D`. In version v1.0.0 and onwards, the location of the configuration file has to be supplied every time T3D is called. Additionally, starting with version v1.0.1, if there is no existing configuration file in the selected location, the tool will prompt the user to use the appropriate option to generate a new configuration with the default settings. Thus, the tool has to be started with the appropriate parameter to generate a new configuration prior to actual usage. This way, it will be possible to modify the configuration file to accommodate the particular needs of the user prior to generating any output. By location here the path, filename, and extension of the configuration file are meant, which implies that any filename and any extension can be used. It is probably best to retain certain norms in naming the configuration files, at least in preserving the file extensions (`.xml`) to avoid confusion. If the fact that a new XML configuration is automatically generated if existing one is not found is too confusing, there may be a separate option available in the command line to generate new default configuration files in future releases.

Please note that after an update, during the development stage, the configuration files will often be extended. Thus, it may be necessary to generate a new configuration file and transfer the custom settings from the old configuration file. The configuration file has two sections:

- A configuration profiles section
- A main section

The **ConfigurationProfiles** section contains a list of **DocumentationProfile** elements. Its elements contain the tags **profilename** as well as elements with settings for the documentation generation, such as output path and paths to files needed for the generation among others. In the default `t3d.xml` that is created, there is a profile called *defaultProfile* that initializes all profile configuration values to the default values. This is also the configuration of an implicitly given profile called *all*. The *all* profile is not part of the XML configuration, but it exists and is known to the tool.

Since v0.4, configuration profiles also have a version (**profileVersion** element), that regulates the profile compatibility. If a profile from an older version is used, T3D will throw an error and recommend profile upgrade or the selection of another profile. Also, further robustness checks have been introduced to provide hints if the configuration profile is otherwise incompatible or corrupted. In case of a problem that can be localized, a corresponding error message is provided suggesting the location of the problem.

Also, there are several generic options, that affect the overall behavior of the T3D tool. The known extensions for files that are to be processed can be specified by means of a regular expression in the **resourceExtensionsRegExp** option. By default, *ttcn*, *ttcn3* and *3mp* file extensions are recognized. Since v1.0.1, there is also support for supplying input by means of what will be referred to as *project* files. These files specify a list of input files and directories (including wildcards). The **projectExtension** option regulates what is to be considered a project file. Note that this is not a regular expression option, therefore only a single value is currently supported. Since v1.0.3, the **ignoredResourceRegExp** option enables the specification of regular expressions for resources within the set of input files that will be ignored during the processing steps. Note that all resources in the input set will be parsed and pre-processed so that definitions within ignored resources will still be correctly resolved during the documentation generation phase. The ignored resources will be skipped during the documentation generation steps (annotated with **(Skipped)** in the output). Note also that currently this may lead to broken links in the HTML documentation output. The default setting for ignored resources is **.*IGNORED.***, meaning that resources which contain **IGNORED** in their full path will be skipped. There is an option to switch recursive processing on and off (**settingRecursiveProcessing**). There is an option to switch aborting on (parsing) errors on and off (**settingAbortOnError**). Both of these options are turned on by default. Turning them off may result in unreliable output. Files that contain syntax errors will be documented only up to the point of the first syntax error occurrence.

Then there are the options regulating the documentation generation process and the logging output (further details are featured below), and the options defining the paths to dynamically linked files necessary for the documentation generation process (which do not concern the user in the usual usage scenarios). These files are described in more detail below.

Starting with v1.0.1, it is possible to define custom names for the supported documentation tags in the **commentTagsConfiguration** section of the configuration.

Finally, there are options regulating the output. Currently there is only one setting - the output directory (**outputDirectory**), where the output files will be placed. Other options regulating the output have been relegated to the command line interface. Note that this option can also be overridden by a corresponding command line option.

In the *main section*, there is currently one single configuration element **defaultConfigurationProfile**. It points to the implicit *all* profile by default. The **all** profile has all settings set to their default values. The **defaultConfigurationProfile** is the profile that will be used if no specific profile is provided as command-line parameter. If the specified **defaultDocumentationProfile** does not exist in the configuration, T3D will fall back to the implicit *all* profile.

Some of the profile options may be moved to the *main section* in the future, as they are rather generic.

A default newly generated configuration file currently looks like this:

```
<T3DConfig>
  <ConfigurationProfiles>
    <DocumentationProfile>
      <profileName>defaultProfile</profileName>
      <profileVersion>v1.0.2</profileVersion>
      <resourceExtensionsRegExp>ttn|ttn3|3mp</resourceExtensionsRegExp>
      <ignoredResourceRegExp>.*IGNORED.*</ignoredResourceRegExp>
      <projectExtension>t3p</projectExtension>
      <settingRecursiveProcessing>true</settingRecursiveProcessing>
      <settingAbortOnError>true</settingAbortOnError>
      <loggingConfiguration>
        <showFullPath>>false</showFullPath>
        <showFilename>true</showFilename>
        <showMessageClass>true</showMessageClass>
        <showDetails>true</showDetails>
        <logOutputPrefix>    </logOutputPrefix>
      </loggingConfiguration>
      <statShowSummary>true</statShowSummary>
      <statShowLOC>true</statShowLOC>
      <commentTagsConfiguration>
        <descTag>desc</descTag>
        <authorTag>author</authorTag>
        <configTag>config</configTag>
        <exceptionTag>exception</exceptionTag>
        <memberTag>member</memberTag>
        <paramTag>param</paramTag>
        <purposeTag>purpose</purposeTag>
        <remarkTag>remark</remarkTag>
        <returnTag>return</returnTag>
        <seeTag>see</seeTag>
        <sinceTag>since</sinceTag>
        <statusTag>status</statusTag>
        <urlTag>url</urlTag>
        <verdictTag>verdict</verdictTag>
        <versionTag>version</versionTag>
      </commentTagsConfiguration>
      <includeConstructBody>true</includeConstructBody>
      <hideConstructBody>>false</hideConstructBody>
      <showOriginalT3DocTags>>false</showOriginalT3DocTags>
      <checkUndocumentedParameters>true</checkUndocumentedParameters>
      <checkFunctionDescTagsRequired>>false</checkFunctionDescTagsRequired>
      <checkConsistentTagUsage>true</checkConsistentTagUsage>
      <checkIdenticalDescriptionTags>true</checkIdenticalDescriptionTags>
```

```

    <checkCyclicImports>true</checkCyclicImports>
    <cssFile>$T3D_HOME/css/doc.css</cssFile>
    <jsFile>$T3D_HOME/js/doc.js</jsFile>
    <xsltFileHTML>$T3D_HOME/xslt/html.xsl</xsltFileHTML>
    <xsltFileImport>$T3D_HOME/t3d/xslt/html_import.xsl</xsltFileImport>
    <outputDirectory>DOCUMENTATION</outputDirectory>
  </DocumentationProfile>
</ConfigurationProfiles>
<defaultConfigurationProfile>defaultProfile</defaultConfigurationProfile>
</T3DConfig>

```

where `{T3D_HOME}` is substituted by the value of the `T3D_HOME` environment variable as specified in the setup when the configuration file is generated. This means that the same profile may not be directly usable on a different configuration, where the `T3D_HOME` environment variable has a different value. This will be subject to change in the future in that at least substitution with the environment variable is done at runtime in the default configuration.

The options regulating the documentation generation process and logging output have the following effects:

- **includeConstructBody** (by default *true*) - The documentation will contain the bodies of module definitions. Leaving out the bodies contributes to a more abstract and concise documentation. However the details may often be necessary.
- **hideConstructBody** (by default *true*) - The module definition bodies are hidden by default when an HTML documentation page is loaded. This makes the initial presentation more compact.
- **showOriginalT3DocTags** (by default *true*) - The HTML documentation will use the original T3Doc tags (e.g. "`@author John Doe`" instead of "`Author(s): John Doe`").
- **checkUndocumentedParameters** (by default *true*) - The T3D generation process log output will include warnings on undocumented formal parameters
- **checkFunctionDescTagsRequired** (by default *false*) - The T3D generation process log output will include warnings on missing `@desc` tags for function definitions
- **checkConsistentTagUsage** (by default *true*) - The T3D generation process log output will include warnings on incorrect tag usage in consistence with the TTCN-3 Documentation Specification Standard (for example: multiple version or other tags that may only be used once, etc.)
- **checkIdenticalDescriptionTags** (by default *true*) - The T3D generation process log output will include warnings identical description tags (defined by string equivalence currently, more sophisticated identity relation may be employed in future releases)
- **checkCyclicImports** (by default *true*) - T3D will check the project for cyclic imports (which may take up to several minutes on large and complex test suites) and report any occurrences in the documentation generation log output.

Command-Line Usage

T3D is used as follows:

```
t3d [options] ((--html | --xml-only) | --local-dependencies)+  
      (filename | path)+
```

This seemingly complicated syntax indicates several peculiarities of the command line usage of the tool:

Apart from any required options (see below), there always have to be at very least two other parameters that needs to be specified. One of these is the input parameter, which can be either a path that contains the TTCN-3 files that should be analyzed, the name of an individual file, or any combination of these (a mixed list), including wildcards.

- If a path provided as input (and recursive processing is enabled in the configuration, which is the default setting), T3D will recursively parse and analyze all files in this directory that match the provided file extensions (which are specified in the configuration file) and generate documentation for all of them. For the current path a simple '.' is sufficient. If no files match these extensions, T3D will output a corresponding message and quit.
- If an individual file is provided as input, then only that file will be processed and documented.
- If the path or filename contains spaces, you need to put the path into quotation marks or use the auto-completion feature of the environment (provided there is one), which should take care of escaping spaces or enclosing the complete path in quotation marks.
- If a list of individual files and/or paths are provided as input, these will be combined and processed together.
- If wildcards are used, these will be expanded by the command-line environment into a list and subsequently analyzed as such.

The other mandatory parameter is the output format, which may (currently) be one of three possible options:

- `--html` is probably the most likely option. It will cause the generation of the complete HTML documentation, including intermediate XML files.
- `--xml-only` will result in the generation of the intermediate XML files only, without the HTML files.
- `--local-dependencies` will result in the generation of an intermediate XML file containing the local dependencies as described below. The other intermediate XML files and the HTML documentation will not be generated if only this output option is selected.

The local dependencies intermediate XML file may be generated in addition to either the other intermediate XML files or the complete HTML documentation, including the intermediate XML files by using two output format parameters together (e.g. `--local-dependencies --html`, the order is irrelevant; `--html` supersedes `--xml-only`, meaning it makes no sense to use both).

A summary of the available options is provided below:

<code>--generate-config <FILE NAME></code>	Generate a new default configuration file at the specified location
<code>--config <FILE NAME></code>	Configuration file location
<code>--profile <PROFILE NAME></code>	Configuration profile
<code>--html</code>	Generate HTML documentation, including intermediate XML files
<code>--xml-only</code>	Generate intermediate XML files only
<code>--local-dependencies</code>	Generate local dependencies XML file
<code>--verbosity <LOG LEVEL></code>	Verbosity level (currently supports ERROR, WARNING and INFORMATION values)
<code>--output-path <PATH></code>	Destination path for the output (if applicable, otherwise ignored). Overrides the profile setting.
<code>--help</code>	Show this usage information screen

The options not discussed so far are:

- `--help` will provide brief usage information. T3D will stop and document no files when the help screen is called.
- The `--generate-config` option (new as of v1.0.1) allows the generation of new default configuration files at the location specified. T3D will then quit.
- The `--config` option (new as of v1.0.0) is mandatory and has to be specified every time T3D is run (except when `--help` or `--generate-config` are used). It specifies the location of the configuration file. Starting with v1.0.1, if no configuration file is found at the specified location, the user will be prompted to use the appropriate option (`--generate-config`) to produce a new default configuration. To use the default location from previous versions one will have to specify it manually, e.g.

```
t3d --config ~/.t3d/t3d.xml
```

on a UNIX based system or

```
t3d --config %APPDATA%\T3D\t3d.xml
```

on a Windows based system. The `--generate-config` and the `--config` options are mutually exclusive, with `--generate-config` having precedence, meaning that if both are specified, T3Q will still only generate the new default configuration and quit. Please not that if the location of the configuration file contains spaces, it has to be either enclosed in quotation marks or the auto-completion feature of the environment has to be used to take care of escaping the spaces.

- The `--profile` option overrides the **defaultConfigurationProfile** in the XML configuration. This means that you can specify multiple profiles in the XML configuration and run T3D using another existing profile without the need to change the XML configuration. If the profile specified on the command-line does not exist, T3D will automatically fall back to the default profile provided in the main section of XML configuration. In turn, if this default profile does not exist as well, T3D will fall back to the implicit *all* profile. If the configuration profile name contains any empty spaces, they need to be escaped or the profile name needs to be enclosed in quotation marks, depending on the environment.
- The `--verbosity` option (newly introduced in v1.0.0) regulates the verbosity level of the output messages based on their type. The possible values are *ERROR*, *WARNING* and *INFORMATION* (in an ascending inclusive order, meaning that when *INFORMATION* is selected, the output will include both *ERROR* and *WARNING* verbosity level messages as well). *INFORMATION* is the default setting. More information about the message types is available in the next section.
- The `--output-path` option (newly introduced in v1.0.1) makes it possible to supply the output path at the command line interface (overriding the profile setting). It is added purely for convenience in case the same profile has to be used on different projects with different destination paths.

The native binary executable for Windows supplied with v1.0.1 is discarded again as of v1.0.2, due to the fact that it does not grant the desired advantages. Thus, the basic usage is reverted to the batch scripts. Starting with v1.0.2, the batch scripts include a "hidden" `--echo` option, which simply outputs the deployment specific call to the Java virtual machine, **without** any command line arguments, meaning that it makes no sense to provide any further command line parameters besides the `--echo` option. The sole purpose of this option is to output the full command line necessary for starting the tool, which may be necessary for embedding into third party tools. This is why this option is considered hidden (it also does not show in the standard help screen). It should not affect the general usage of the tool. If, for whatever reason, other command line arguments are supplied together with the `--echo` option, then the `--echo` option needs to be the first command line argument.

Linked Files

A summary of all linked files as specified in the configuration file, their default location and their function is presented below:

- `cssFile $T3D_HOME/css/doc.css` - The CSS file defining the layout in the HTML output
- `jsFile $T3D_HOME/css/doc.js` - The JavaScript file defining basic functionalities for the navigation and other interactive parts
- `xsltFileHTML $T3D_HOME/xslt/html.xsl` - The HTML transformation file used when generating HTML output for the main and the module parameters views
- `xsltFileImport $T3D_HOME/xslt/html_import.xsl` - The HTML transformation file used when generating HTML output of the import view

These linked files can be used to customize and manipulate the output of the T3D tool as described in more detail in the subsequent sections. Please note that changing these files requires adequate knowledge of the technologies involved, the lack of which may cause broken functionalities in the generated content or failure in the generation process. Upon request, some customizations and adjustments may be provided with subsequent or customized versions of the tool.

Output Formats

The output of T3D can be split into several categories. There is the *main output*, which is the product of the documentation process, and, in the general perception, comes in the form of multiple HTML files and several XML files, which store extracted data in an intermediate format and serve as a basis for the documentation generation. Then there is also the output of the documentation generation process itself, which will be referred to as the *documentation generation log*. The details follow below.

Performance and Memory Usage

Large TTCN-3 test suites tend to take quite a while to process (both for parsing and for documentation generation). Therefore, it is generally a good idea to set larger memory limits (as far as the system allows) in order to improve processing time and avoid possible memory problems. The default setting is to set the **upper** memory limit to 512MB, which by today's standards is rather conservative, however, it should be sufficient for smaller to medium-sized TTCN-3 test suites. The optimal memory limits are not easy to determine, and ways to automatically calculate and set these depending on the available system resources, the size of the input TTCN-3 test suite, and the configuration in use are currently being investigated. In the meantime, should processing take too long or memory errors occur, it is advisable to set a higher upper memory limit, depending on the available system resources. This can be done by manually editing the parameters in the start scripts (`t3d.bat` and `t3d` for Windows and Unix respectively). These files should be edited very carefully, as mistakes may prevent T3D from starting. Under Windows, in `t3d.bat`, set the `-Xmx` parameter in following line:

```
set JAVA_CMD=%JAVA% -Xmx512m -Xss128m -cp "%CLASSPATH%" org.etsi.t3d.T3D
```

to the desired upper limit (e.g. to `-Xmx1024m` for a 1GB upper memory limit).

Under Unix, the `t3d` file can be changed in a similar fashion by setting the `-Xmx` parameter in the

```
JAVA_CMD="$JAVA -Xmx512m -Xss128m -cp $CLASSPATH org.etsi.t3d.T3D"
```

line to the desired upper limit.

In general, the optimal setting depends on the size of the TTCN-3 test suite and to a degree on the selected output format.

As of v1.0.3, a tool to guesstimate the optimal memory settings is included with T3D. This tool is launched prior to the actual tool execution and attempts to detect the maximum memory settings with which T3D can be started at that particular moment, aiming to both reduce processing time and avoid potential out of memory errors. It should be noted that this tool can be considered in beta status, as in some edge cases it may cause T3D to crash or to fail at start. Such cases should be reported so that the memory detection tool can be further adjusted to avoid such issues in the future. It is still possible to select preferred memory settings manually by adjusting the start-up scripts as described above, where the particular line has been changed to:

```
set JAVA_CMD=%JAVA% -Xmx%HEAP%m -Xss128m -cp "%CLASSPATH%" org.etsi.t3d.T3D
```

under Windows and to

```
JAVA_CMD="$JAVA -Xmx`$MT_CMD`m -Xss128m -cp $CLASSPATH org.etsi.t3d.T3D"
```

under Unix, and the %HEAP% and ` \$MT_CMD` parts of the line should be substituted with the desired settings.