opentext™

OpenText™ Embedded Output Transformation Engine

Standalone Installation Guide

This document provides information about how to install or upgrade a standalone instance of Output Transformation Engine.

VDTOTS240200-ITE-EN-1

OpenText™ Embedded Output Transformation Engine
Standalone Installation Guide

VDTOTS240200-ITE-EN-1
Rev.: 2024-Apr-16

This documentation has been created for OpenText™ Embedded Output Transformation Engine CE 24.2.
It is also valid for subsequent software releases unless OpenText has made newer documentation available with the product, on an OpenText website, or by any other means.

Open Text Corporation

275 Frank Tompa Drive, Waterloo, Ontario, Canada, N2L 0A1

Tel: +1-519-888-7111
Toll Free Canada/USA: 1-800-499-6544 International: +800-4996-5440
Fax: +1-519-888-0677
Support: https://support.opentext.com
For more information, visit https://www.opentext.com

Table of Contents

Chapter 1

OpenText Embedded Output Transformation Engine Installations

1.1 Installation Overview

This guide details the installation process for running Output Transformation Engine as a standalone module outside of Output Transformation Server. If you want to install Output Transformation Engine as a module running within Output Transformation Server, please refer to the instructions in *OpenText Output Transformation Server Installation Guide*.

1.2 License Keys

License key files (or licenses) are sent out by OpenText Licensing upon the availability of a new installation, the renewal of a yearly contract, or an update of an OpenText product. You can send your licensing inquiries to:

support@opentext.com

To deploy a new license to Output Transformation Designer during installation, you must manually copy the license to the installation directory. For more information about deploying the license, refer to step 4 under "Installation Procedure" on page 8.

For more information about deploying an updated or renewed license, see *OpenText Output Transformation Designer - User Guide (VDTOTS-H-UTD)*.

1.2.1 Storing License Key Files

The license key files OpenText Licensing sends you when you purchase an OpenText product are used to:

- Install the product.
- Deploy the product's license key files for future validation purposes.

You store license key files anywhere on your computer system when you first receive them. During the installation, you must copy your license file into the `settings` folder in your installation directory.

1.3 Requirements

Output Transformation Engine is written 100% in Java. It has been tested and is supported on various platforms running the specified Java Virtual Machine (JVM).

The system requirements for running Output Transformation Engine are listed in the subsequent topics.

For the most up-to-date listing of supported operating systems, JVM/JDKs, application servers, ECM systems, databases, web browsers and their compatible versions, refer to the Release Notes.

1.3.1 Minimum Server Machine Requirements

The minimum system requirements for Output Transformation Engine is listed below. Keep in mind that all memory and free disk space requirements pertain to OpenText products only; you will need access to additional memory and/or disk space for storage and processing of your projects, documents, and data.

- Processor:
  - Windows: x86 compatible, Pentium 1 GHz or higher
  - AIX: Power
  - Linux: x86 compatible, Pentium 1 GHz
  - Solaris: UltraSPARC T1 or T2
- RAM (memory): 1024 MB
- Free disk space: 1 GB

1.3.2 Linux Kernel Requirements

In order to run Output Transformation Engine on either AIX, Sun, or Linux platforms, you must have a kernel of 2.4.18 or higher.

1.3.3 Output Transformation Engine Installation Files

Output Transformation Engine is distributed as a ZIP file (also known as an installation module) named `OTS_OutputTransformation_<version>.zip` in an installation package via FTP. In addition to this ZIP file, to install the product you must also have the base and the core installation modules. The base installer file is named `OTS_Base.zip` and is stored in the `base` folder of your installation package.

The core installation module is named `OTS_Core_<version>.zip` and contains Output Transformation Designer as well as some core and third party JAR files. The core installation module along with all optional installation modules are stored in the modules directory of your install package.

Also included with your installation package is a `subproducts` folder, which includes some additional integration JAR files, an PDF API toolkit for developers, and the Embedded Output Transformation Engine ZIP file.

1.4 Installing Output Transformation Engine

1.4.1 Prerequisites

Before starting the installation process, make sure you have the following software components and files ready:

- Java JDK 1.8 installed on your target machine. You can confirm whether it is installed by opening a command prompt window and running the following command:

  ```
  java -version
  ```

  A similar response should be returned:

  ```
  java version "1.8.0_25"
  Java(TM) SE Runtime Environment (build 1.8.0_25-b17)
  Java HotSpot(TM) 64-Bit Server VM (build 23.25-b01, mixed mode)
  ```

- The following files downloaded from My Support or otherwise sent to you:
  - `OTS_Base.zip` module
  - License file (`license.txt`)
- The following required software modules:
  - `OTS_Core_<build_label>.zip`
  - `OTS_OutputTransformation_<build_label>.zip`
- If you are using z/OS, before installing the application you must enable automatic file conversions between EBCDIC and ASCII code sets so that data and property files can be translated by the system when necessary. The environment variable to run is:

  ```
  export _BPXK_AUTOCVT=ON
  ```

  You can enable this setting for one time use by simply running the command on your z/OS server, or you can add the environment variable to a user's `.profile` file, which will enable the setting every time the particular user logs on.

1.4.2 Installation Procedure

Before attempting an installation, users should be aware that depending on the location of the installation directory, administrator rights may be required. For example, if you are using Windows and copying and extracting the files to a folder under `Program Files` or `Program Files (x86)`, then you will need administrator rights to copy or write files.

Note: The installation procedure is the same for Windows, z/OS, AIX, Sun, or Linux-based machines, however, all commands and file paths are written using a backslash, which Unix users must substitute with a forward slash.

To install Output Transformation Engine:

1. On your local machine, create a new folder to store your installation and put `OTS_Base.zip` from your installation package in this folder. This directory is your base installation folder and subsequent references to it will be represented by the `<OTS_home>` variable.

2. Open a command prompt window and navigate to your `<OTS_home>` directory, then run the following command to extract the contents of the installation file. At the command prompt, type:

   ```
   jar -xf OTS_Base.zip
   ```

   Note: If the jar command is not recognized, you may have a Java JRE installed instead of the full JDK. If this is the case, you can extract the files to the `<OTS_home>` directory using your preferred ZIP utility.

   Several folders are extracted from the zip and added to your installation directory. This may take up to several minutes.

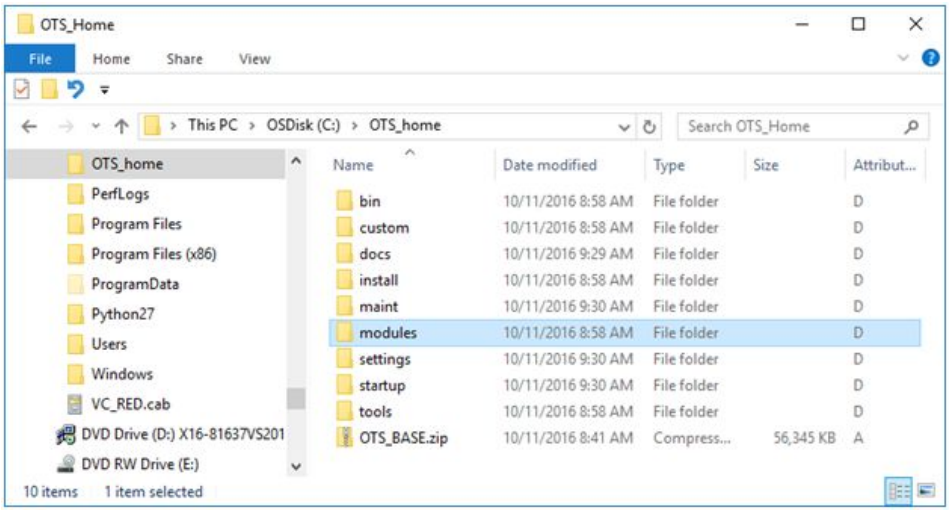

Figure 1-1: Folder structure following extraction of the OTS_Base.zip file

3. From your installation package, copy the zipped installation modules to the `<OTS_home>\modules` folder.

   Note: As each module is kept in a separate file, if you have any other module then you can add them to the `<OTS_home>\modules` folder to install them in conjunction with Output Transformation Engine. At a minimum, you must include `OTS_Core_<version>.zip` and `OTS_OutputTransformation_<version>.zip` in the folder.

4. Before you can run Output Transformation Engine, you must be in possession of a valid license. Copy and paste your license file into the `<OTS_home>\settings` folder. If necessary, you must rename your license file to `license.txt` for it to be recognized by the system.

5. You are now ready to install the product and any other modules you have. Navigate to the `<OTS_home>` directory and at the command prompt, type:

   ```
   java -jar bin\OT-Bootstrap.jar
   ```

6. In the command prompt window, you are asked to open and review the End User License Agreement (EULA) before continuing. The EULA can be located at `<OTS_home>\docs\licensing\OT_Clickthrough_EULA.txt`. After reviewing the EULA, if you accept its terms then type **Y** at the command prompt and press **Enter** to continue. (If you do not accept the terms, the installation process will be cancelled.)

   A script is executed to extract the contents of your installation modules in the `<OTS_home>\modules` directory and a command prompt window is displayed to show the progress. The extracted contents from the installation modules are put into the `<OTS_home>\install\<version>` directory.

7. By default, this installation procedure uses `<OTS_home>` as the working directory for all users, which will require users to have write permissions to your `<OTS_home>` directory. For more information on maintaining separate work and installation directories, see *OpenText Embedded Output Transformation Engine - User Guide (VDTOTS-H-UTE)* and perform those necessary steps before proceeding to the next step.

8. You are now ready to launch the product for the first time. First, navigate to the `<OTS_home>\startup` directory. Then you can launch Output Transformation Engine for the first time by using `startDesigner.bat` for Windows users or `startDesigner.sh` for Unix and Linux users.

   For more information about the locations and details of the files installed, see "Installed Files and Directory Structure" on page 10.

   Note: For Windows users, the installation process does not create any shortcuts on your desktop or Start menu, however, you can manually create a shortcut to the `startDesigner.exe` file.

1.5 Special Considerations

1.5.1 Manual Deployments

A pre-packaged engine file known as the Embedded Output Transformation Engine ZIP file is available to assist with manual deployments to another machine. If any of the following conditions are true, you might want to use the packaged engine file rather than installing Output Transformation Engine directly to your target server:

- You are installing to a production server where there is no display, such as Linux, IBM z/OS USS, or another type of mainframe.
- Your server has limited memory or processing capacity and a smaller install footprint to conserve system resources is required.
- You do not need to set up projects.
- You want to use only the engine to submit jobs through the API so the GUI is not required.

In these cases, you can use the `EOTE-<version>.zip` engine file that is shipped in the `subproducts` directory along with your installation files.

1.6 Installed Files and Directory Structure

It is strongly recommended that you use the directory structure that is created by the installer for Output Transformation Engine and not modify it in any way. This consistent structure, which is reflected in all installations of Output Transformation Engine, ensures that file paths are correctly represented in all components, thereby minimizing file-locating errors. Furthermore, all files and folders are relative to the top level installation folder so changing them will cause issues.

Tip: We recommend that you do not make changes to the default directory structure unless there are pressing operational reasons to do so.

1.6.1 Output Transformation Engine Root Directory Structure

The directory structure for Output Transformation Engine beneath your chosen installation folder is shown in the table below.

Directory	Description
\BaseRepositories	Common storage area for solutions and resources used by both the engine and Output Transformation Designer. You may change the location your base repository folder by modifying the `settings/BaseRepositoryLocation.properties` file.

Directory	Description
\bin	JAR files for maintenance and general running of the product
\client	Sample class files for connecting to an application server and running jobs through the job runner
\custom	Contains subfolders for custom function definitions, Data Transformation Engine pre-parsers, third party JAR files, and Output Transformation Designer plugins
\custom\jars	Holds any custom components, third party JAR files required for the custom components, plus Data Transformation Engine pre-parsers and external functions
\custom\plugins	Holds any custom Output Transformation Designer plugins
\docs	Help system files for OpenText Output Transformation Suite
\install	Contains subfolders with version(s) of the product. All files below this folder should never be modified.
\install\<version>\dev-studio	Files pertaining to Output Transformation Designer
\install\<version>\initialFiles\common\	Folder used to create any base repositories for the first time
\install\<version>\initialFiles\common_classes	Location where all compiled classes from Output Transformation Designer are stored
\install\<version>\initialFiles\common_configs	System configuration files
\install\<version>\initialFiles\common_deployments	Project deployments
\install\<version>\initialFiles\common_logProfiles	Location where logging profiles are stored
\install\<version>\initialFiles\common_logs	Default location for logs
\install\<version>\initialFiles\common_resources	Contains all resources for event listeners and other components
\install\<version>\initialFiles\common_sample	Samples for each of the plugins
\install\<version>\initialFiles\common_stats	Output Transformation Designer statistics
\install\<version>\initialFiles\common\com	Holds user's custom code

Directory	Description
\install\<version>\initialFiles\common\input	Contains input files
\install\<version>\initialFiles\common\output	Contains a variety of outputs: logs, statistics, output files, etc.
\install\<version>\initialFiles\system\	Contains component definitions, sample component for developers, and some default configurations
\install\<version>\lib	System libraries
\maint	Contains scripts for installation and patches
\modules	Installation modules for OpenText Output Transformation Suite plugins
\settings	Contains the license file plus various property files
\startup	Contains startup scripts for launching the product
\tools	Location of built-in tools

1.7 Upgrading to a Newer Version of Output Transformation Engine

If you already have Output Transformation Engine installed on your machine and want to upgrade to a new version, you can apply a patch to upgrade the application while reusing your projects, configurations, and resources from the previous version.

Note: The patch method of upgrading Output Transformation Engine is only applicable to version 16.0 or higher.

The patch is contained within a zip file that must always abide by a particular naming convention for them to be recognized by the installer scripts. Consequently, after downloading the patch file onto your machine the file name should never be modified. The patch file name must retain the `OTS_Patch_16.x.xx_<Build>_(PatchOf)_16.x.xx_<BaseBuild>.zip` pattern, with `<Build>` representing the new version you are upgrading to and `<BaseBuild>` indicating the required base version to upgrade from.

Patches can be deployed through two different approaches, either through "Basic Patch Upgrade Options" on page 13 or "Advanced Patch Upgrade Options" on page 14. **Basic** upgrades are recommended for users who can download the patch files directly onto their system and only need to perform the upgrade on the same machine. The **Advanced** upgrade method allows for some additional patch settings to be configured, such as automatically retrieving the patch from a network location and the ability to distribute a patch to your deployed servers. Refer to the section on your preferred upgrade method for more details.

In order to run the patches, you must ensure that the embedded patch scripts contain the correct <OTS_home> location. If you have moved the patch script, you must update the correct value for the OTS_HOME= parameter within the CheckForPatches.bat/.sh file.

1.7.1 Basic Patch Upgrade Options

To upgrade to a newer version of Output Transformation Engine:

1. Download the patch zip file onto your machine and save it in your existing version's <OTS_home>\modules directory.
2. Navigate to the <OTS_home>\maint directory and run the CheckForPatches.bat\.sh file.

   Note: If you encounter any read/write errors when running the executable file in Windows, try using the **Run as administrator** option to execute the file.

   A script is executed to extract the contents of the installation modules in the patch directory and a command prompt window is displayed to show the script's progress. The files and folders extracted from the patch installation modules are put into the <OTS_home>\install\<NewVersion> directory. When it is finished, you are prompted to press any key to close the window.

   Also, a CheckForPatches.log file is created in the <OTS_home>\maint folder, which contains all information presented on the screen as well as more detailed information not presented on the screen.

Once the patch has been successfully applied, you are now ready to launch the product as normal. For more information on switching to the new version of the software as this may not be automatic, see "Managing Different Versions During Startup" on page 15.

The <OTS_home>\install\ directory contains separate installation folders for the original and upgraded versions. If you do not intend on rolling back to the previous version, then you can safely delete the directory containing the original version's installation files since it will not be used unless you explicitly set the properties to use the older version upon starting the application.

1.7.2 Advanced Patch Upgrade Options

Some custom patching options can be configured by modifying the `CheckForPatches.properties` file. In addition to expanding patches, the patches may be retrieved from a network location and/or distributed to other network locations (other servers with Output Transformation Engine installed).

To set the options, you must open the property file for editing and add your parameter values in the code where indicated. After saving your edits to the property file, you can run the `CheckForPatches.bat/.sh` file as normal and your configured options are used accordingly. (If you encounter any read/write errors when running the executable file in Windows, try using the **Run as administrator** option to execute the file.)

Note: When setting file paths for Windows, you must use the backslash (\) escape character to escape the colon (:) and backslash (\) characters. For example, `C\:\\patches` is used to represent the `C:\patches` file path.

You can set the following options in the `CheckForPatches.properties` file:

Table 1-1: Advanced Patch Upgrade Options

Property	Description	Accepted Values
`retrieve.location=`	Specifies a network location from which to obtain the patch files. The network location must include the patch folder name matching the defined pattern.	The network location is a folder containing one or more patch folders matching the defined pattern. During retrieval, patches are copied to the `<OTS_home>\modules` folder. If no location is provided, the application looks for patches in the `\modules` folder.
`retrieve.mode=`	Determines the patch versions to retrieve from the specified retrieval location.	**LATEST** to retrieve the most recent patch files or **ALL** to retrieve all patch files available.
`expand.mode=`	Indicates which patch versions are extracted for use from the `<OTS_home>\modules` folder to the `<OTS_home>\install` folder.	**LATEST** to extract the most recent patch version or **ALL** to extract all patch versions available. This setting can also be used for Basic-style upgrades.
`distribute.mode=`	Determines which patches are dispersed to your deployed servers.	**LATEST** to distribute the most recent patch version or **ALL** to distribute all patch versions available.

Property	Description	Accepted Values
distribute.location.x=	Indicates the network location for where patches are copied. This must be another <OTS_home> install location, and the patches are copied to the other <OTS_home>\modules directory. Any number of locations can be set by changing the x variable to represent an individual location; for example, distribute.location.1=, distribute.location.2=, etc. **Note:** Even though this distribution is an easy way to push patches to many of your servers, you must run the CheckForPatches.bat/.sh file on each of the servers in order to expand the patches and make them ready for use on the servers.	Set a number for the distribution location and the full network path.

1.7.3 Managing Different Versions During Startup

As multiple versions of the product can be installed at any given time, you can use the <OTS_home>\settings\startup.properties file to control which version is used upon starting the application.

If you are using startDesigner.bat\.sh to launch Output Transformation Designer, open the startup.properties file for editing and locate the designer.version property; this property specifies the version to use when launching the application and you can enter an asterisk (*) as the value if you wish to always use the newest installed version.

If you are using startEngine.bat\.sh to launch a standalone instance of the engine, open the startup.properties file for editing and locate the engine.version property; this property specifies the version to use when launching the engine and you can enter an asterisk (*) as the value if you wish to always use the newest installed version.

For users with an application server, you must repackage your EAR or WAR files using the **Package and Deploy Wizard** in Output Transformation Designer and then redeploy the new EAR/WAR file to your application servers. Tomcat and JBoss Web Server users must make an additional change. In the `setenv.bat` or `.sh` file, locate the line stating set "`OTS_VERSION=16.x.xx_<BuildNumber>`" and as the value, indicate the version you want to run. Tomcat users can locate this file in the `<OTS_home>\TomcatBase\<instancename>\bin` folder while JBoss Web Server users can locate this file in the `<OTS_home>\JBossWebServerBase\<instancename>\bin` folder.

1.7.4 Identifying Installed Modules and Build Versions

Following a successful installation or update of Output Transformation Engine, you can quickly discover which products and respective versions are installed on your machine by inspecting the build property files that are added to the `<OTS_home>\install\<version>` directory. In this folder, you can verify whether a specific product was installed by the presence of an `OTS_<ProductName>_build.properties` file that is added for each installed product, while opening the file also identifies its version on the `OTS_<ProductName>_buildLabel=<Version>` comment line. For instance, having the `OTS_ArchiveNavigator_build.properties` file in the directory and upon opening the file, seeing the `OTS_ArchiveNavigator_buildLabel=16.0.01_1234` comment reveals that build version 16.0.01_1234 of the Archive Navigator plugin was successfully installed on your machine during the installation or update process.

Note: As with all other files stored beneath the `\install` folder, it is important to never modify any of the property files as it may have adverse effects on your installation.

1.8 Uninstalling Output Transformation Engine

During the installation, nothing is written to the registry on Windows or the system configuration files on Linux/Unix-based machines so if you wish to remove Output Transformation Engine, the installation directory can simply be deleted.

1.9 Troubleshooting

1.9.1 Java Virtual Machine Conflicts in Windows

After the successful installation of Output Transformation Engine, if you are having difficulty starting the application in designer mode on your system, it may be caused by a conflict with your Java versions due to having multiple versions installed on your machine. You must ensure that the version of Java Output Transformation Engine is using is consistent with the version Windows is using.

To check which version of Java Windows is using:

1. Open a command prompt window.
2. On the command line, type `java -version` and press **Enter**.

If you are using the 64-bit version Java, this will be explicitly stated. If you are using the 32-bit version of Java, the version is not stated at all.

If Windows is using one version of JVM and you are trying to start Output Transformation Engine with another version of JVM, you must modify your Java settings in Windows so that they are consistent.

Note: The instructions below do not explicitly explain how to access the Environment Variables dialog as it varies between different versions of Windows, but it often resides with other advanced system properties. Consult the Microsoft Windows documentation for your respective version for more information.

The following changes to Windows' Java settings need to be made on the **Environment Variables** dialog:

- In the **User variable for <user>** section, select the **Path** variable. Click **Edit** and ensure that your preferred JVM version appears in the list first, followed by any other versions.
- In the **System variables** section, select the **Path** variable. Click **Edit** and ensure that your preferred JVM version appears in the list first, followed by any other versions.

After making the changes, click **OK** to save your changes. You must log out of the user account and then log back in with the same user to ensure that your changes take affect and then you can start Output Transformation Engine as usual.

1.9.2 Extracting the Installation Files Using WinZip

When using WinZip to extract the files from the base installation file (`OTS_Base.zip`) on Windows machines, you may encounter errors regarding file path names being too long or skipped files, which will cause the extraction process, and subsequently the installation process, to fail due to the missing files. To bypass these errors, you can try using a different extraction utility. Or if the error states your file path is too long, you can try shortening the file path location for your home installation directory before running WinZip again.

Chapter 2

Running in Headless Environments

2.1 Abstract Window Toolkit

Many environments, such as mainframe machines and dedicated servers, do not support a display, keyboard, or mouse; these are so-called headless environments. In UNIX environments, this usually means that X or an equivalent windowing system does not have to be present.

The **Abstract Window Toolkit** (AWT) is the standard API for providing graphical user interfaces for Java programs. Parts of the AWT rely on graphical libraries provided by the operating system for display—such as: dialogs, toolbars, and buttons—font manipulation, and other features.

The Output Transformation Engine engine relies on the graphics processing functionality provided by the AWT for a number of uses during the transformation process, so this presents a problem when there is a requirement to run in a headless server environment where no graphics display capabilities are available by default.

There is a solution to this problem:

- The AWT library in Java 1.4.x and later features headless support.

For more information, see:

- http://java.sun.com/j2se/1.4.2/docs/guide/awt/AWTChanges.html#headless
- http://www.oracle.com/technetwork/articles/javase/headless-136834.html

To run the Java environment with a headless implementation, the following property is specified at the Java command line:

```
-Djava.awt.headless=true
```

This feature, however, still requires native graphical libraries, such as X11, to be installed.

Chapter 3

Installing the Private Help Server

3.1 Providing the online help on a local help server (Private Help Server)

The online help for this module is delivered using the OpenText Global Help Server (GHS) system, which provides your users with live access to the latest version of the help. If you cannot use the GHS system, for example, if your site does not have Internet access, you can install the OpenText Private Help Server (PHS), a local version of the help system that can host your OpenText online help on your organization's network. After the PHS is installed, you can then configure your OpenText module(s) to forward all online help requests to your PHS. For detailed information about installing the PHS, see *OpenText Help System - Private Help Server Administration Guide (OTHS-AGD)*.

Notes

- The Private Help Server can support multiple OpenText modules. If the Private Help Server has already been installed within your organization to support another OpenText module, you can add additional OpenText module online helps to that installation.
- If you are replacing a previous PHS installation, see *OpenText Help System - Private Help Server Administration Guide (OTHS-AGD)*.
- If the server you want to use for the PHS installation cannot connect to the Internet, see *OpenText Help System - Private Help Server Administration Guide (OTHS-AGD)*.

Once the PHS is installed or upgraded, you can use its Online Help Deployer to download online helps from the GHS system by entering the help deployment codes listed in "Help deployment codes" on page 21. For more information about using the codes, see *OpenText Help System - Private Help Server Administration Guide (OTHS-AGD)*.

Table 3-1: Help deployment codes

Code	Product
VDTOTS240200-ITE	OpenText™ Embedded Output Transformation Engine CE 24.2

3.1.1 Configuring Output Transformation Engine to use the Private Help Server

Once you have the Private Help Server fully configured, you can redirect help requests from the Global Help Server to the Private Help Server in Output Transformation Designer.

To redirect the help requests:

1. On the **File Systems** tab in Output Transformation Designer, open your system configuration file for editing. (By default, `default.xSystemConfig` is the active system configuration file unless you have changed it.)
2. In your system configuration file, locate the **HelpSystem** > **UrlRoot** parameter.
3. The **UrlRoot** parameter indicates the URL root for the help system. If no value is specified, then help calls are directed to the Global Help Server. To redirect calls to the Private Help Server, enter the URL root to your instance of the Private Help Server using the following convention:

   http://<host_name>:<port_name>/OTHelpServer/mapperpi
4. **Save** the configuration settings.