opentext™

OpenText Professional Performance Engineering

Software version: 25.1

Installation Guide

Go to Help Center online
https://admhelp.microfocus.com/lr/

Document release date: January 2025

Send Us Feedback

Let us know how we can improve your experience with the Installation Guide.
Send your email to: admdocteam@opentext.com

Legal Notices

The only warranties for products and services of Open Text and its affiliates and licensors (“Open Text”) are as may be set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Open Text shall not be liable for technical or editorial errors or omissions contained herein. The information contained herein is subject to change without notice.

Disclaimer
Certain versions of software accessible here may contain branding from Hewlett-Packard Company (now HP Inc.) and Hewlett Packard Enterprise Company. This software was acquired on September 1, 2017 by Micro Focus and is now offered by OpenText, a separately owned and operated company. Any reference to the HP and Hewlett Packard Enterprise/HPE marks is historical in nature, and the HP and Hewlett Packard Enterprise/HPE marks are the property of their respective owners.

Contents

Welcome

Welcome to the OpenText™ Professional Performance Engineering Installation Guide.

OpenText Professional Performance Engineering, a tool for performance testing, stresses your application to isolate and identify potential client, network, and server bottlenecks.

This guide describes installation and set up.

Help Center

Access context sensitive help for product components by pressing F1 within a dialog box, or use the Help menu.

You can access the Help Centers on the web.

- https://admhelp.microfocus.com/lr/
- https://admhelp.microfocus.com/vugen/

When you open a link to a Help Center, make sure that the relevant product version is selected.

You can also download and install local versions of the help. For details, see Download Help Centers.

To switch between online and local modes, select **Help > Help Center Options > Open Online** or **Open Locally**.

> **Note:**
>
> - To view the Help Center in a browser or within VuGen, make sure JavaScript (Active scripting) is enabled in your browser settings.
> - If the online Help Center is unavailable at the start of an application session, the **Open Online** option might remain disabled during the whole session with that application.

Product components

This installation guide describes the installation of the following OpenText Professional Performance Engineering components. For more information about each of these components, refer to the Help Center.

Virtual User Generator (VuGen)	A tool for creating virtual user (Vuser) scripts, primarily through recording. Vuser scripts emulate users without a graphical user interface by using direct function calls.
Controller	Controls the execution of load and performance tests based on Vuser scripts. Includes the online monitors which monitor and display information about the test execution.
Analysis	Provides graphs and reports for analyzing the load test.
Load generator	Component for running Vusers (including Windows-based GUI Vusers) to generate load.
Network Virtualization	Enables you to emulate network conditions in your testing environment, such as latency, packet loss, and bandwidth, to accurately test how your application responds under a multitude of network conditions.
Monitors over Firewall	Component on the agent machine for monitoring over a firewall.
MI Listener	Component for the MI Listener machine, used in running Vusers and monitoring over a firewall.
Data Hub	Communication channel that supports more robust communication between the components.
Data Hub Web Connector	Enables the Data Hub to communicate over the web and over firewalls.

Standalone applications

Some of the components above are available for standalone installation. These can be found in the installation package's **Standalone Applications** folder.

Additional components

Additional components (such as the Citrix agent) can be found in the installation package's **Additional Components** folder. For details and installation instructions, see Additional components and applications in the Help Center.

Download installation packages

You can access the installation packages from the free trial page or by selecting your account on the Software Licenses and Downloads page.

You can also download VuGen, OpenText Performance Engineering for Developers, VTS, Data Hub, and more, for free from AppDelivery Marketplace.

System requirements and prerequisites

This chapter includes:

System requirements

The following information can be found in the Support matrix:

- System requirements for running OpenText Professional Performance Engineering components on a Windows system
- System requirements for running a load generator on a Windows or Linux system
- A full list of components that can be installed on the same physical machine at the same time

Using TLS (SSL) certificates

During installation of some OpenText Professional Performance Engineering components, you have the option to install CA and TLS (SSL) certificates.

These certificates are used for authentication and secure communication. Both certificates must be in *.cer (X.509) format.

- **CA certificate**: Enter the path to an existing certificate, or leave it blank to skip the step. The CA certificate should be generated in advance.
- **TLS certificate**: Select from an existing certificate file. Alternatively, the certificate can be generated automatically if you provide the CA certificate containing the private key.

For details on generating a CA certificate and working with certificates, see Secure communication with TLS (SSL) in the Help Center.

Before you install

Before you install OpenText Professional Performance Engineering components, review the information below.

For additional notes and limitations, see Known issues in the Help Center.

Security review	We strongly recommend that you install the components on dedicated machines that do not contain, or provide access to, sensitive information; and that you do a thorough security review of the network topology and access levels in your testing environment.
Virtual environments	The architectures provided by virtualization vendors are rapidly evolving. OpenText Professional Performance Engineering is expected to function as designed in these changing environments, as long as the third-party vendor guarantees full compatibility of the virtualized environment with the OpenText Professional Performance Engineering-approved hardware requirements. If you follow the system requirements and support matrix to create the virtual machine, OpenText Professional Performance Engineering will work correctly. Working on top of a virtual machine may require access to the virtualization server hardware/monitoring environment, to ensure the virtualization server is not saturated; otherwise, this might obscure the virtual machine's measurements and lead to false results.
Permission requirements	The installation process for all components requires a full administrator account (root account for Linux systems) for all operating systems. You can install components with UAC and DEP enabled, but we recommend to keep them disabled during installation (or enable DEP for essential Windows programs and services only). After installation, all of the OpenText Professional Performance Engineering applications and components can run under a standard user account (an administrator account is not required), with UAC and DEP enabled.

Antivirus applications	It is recommended that you close all antivirus applications before installation.
Running Web Services Vusers on Windows	The Axis toolkit and WSE are no longer automatically installed with OpenText Professional Performance Engineering . When running Web Services Vusers on a standalone load generator, some legacy configurations may require activation of .NET 3.5 (see MSDN), the inclusion of the Axis java files, or an installation of WSE (2.0 SP3 and WSE 3.0). To avoid using these components, recreate the scripts and import the WSDL using the default WCF toolkit. To use these components: • For WSE, install the WSE components from the OneLG installation package's folders, **DVD\lrunner\Common\wse20sp3** and **DVD\lrunner\Common\wse30**, or download them from the Microsoft website. • For Axis, copy the Axis java files to **<installdir>\bin\java\Axis**. For OneLG, the files are located in the **prerequisites\Axis** subfolder of the extracted installation package, by default **<OneLG <version> root folder>\prerequisites\Axis**.This allows you to run legacy scripts on an Axis toolkit. For more details, see Known issues for Web Services.
Running Java Vusers on Windows	OpenJDK 32-bit is no longer automatically installed. To work with Java protocols, you need to manually install a supported version of the JDK. VuGen uses internal logic to automatically detect your JRE version. For details, see Set up the environment for Java protocols.
Coexistence with OpenText Functional Testing	If you install OpenText Professional Performance Engineering and OpenText Functional Testing on the same machine, and then uninstall one of them, the remaining software program may not function correctly. **Resolution:** Run a Repair of the software program you want to use on the machine.

Licensing	Note the following regarding licensing: • To run Vusers with Controller, there must be a valid license installed on the Controller machine. The Community license bundle is automatically installed (free of charge) during OpenText Professional Performance Engineering setup, providing 50 Vusers. • If upgrading from a version earlier than 2020, you need to replace your current license to work with the upgraded installation. If you are using the Community license, it is automatically replaced during installation. For any other license, contact licensing support to obtain a valid, compatible license. For details on managing licenses and extending your Vuser capacity, see About licenses in the Help Center.
Custom certificates	When upgrading from version 12.55 or earlier, if you used custom certificates in your previous installation, install them again when prompted for certificates during setup. Otherwise, the setup program will overwrite them using the defaults.
Network Virtualization	• When upgrading from a version prior to 12.50, you must manually uninstall all Network Virtualization components before running the installer. • If a firewall is active, open the NV default port 8182 (or another port you select during installation) for TCP. You can set up an exception to Windows Firewall by selecting custom installation in the NV Setup wizard. • If you choose to install NV automatically (as part of a full installation), you must deactivate Windows SmartScreen before proceeding with the NV installation. Do the following: a. Open HKEY_LOCAL_ MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\ Explorer in the Registry Editor b. Change the Value data for **SmartScreenEnabled** to **Off**. **Note:** You do not need to deactivate SmartScreen when installing NV individually.

Installing components on Windows

This chapter describes how to install either the full version of OpenText Professional Performance Engineering or one of its components on a Windows platform.

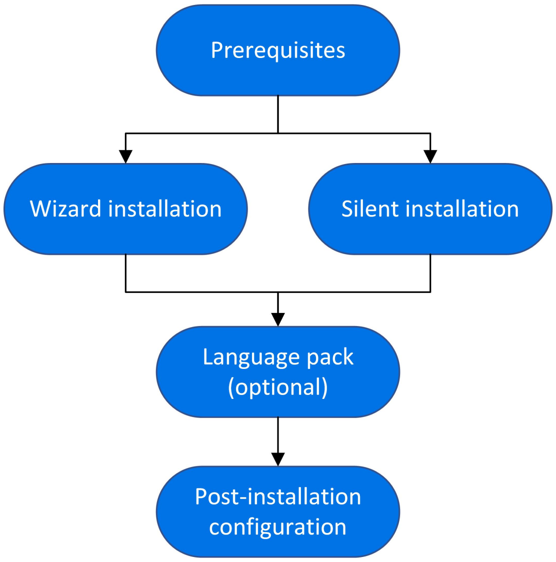
This chapter includes:

Installation workflow

The installation package includes the Setup wizard that guides you through the process of installing the components.

The final stage of the installation includes the optional installation of Network Virtualization.

Installing on a Windows system involves the steps shown below:



1. **Prerequisites**

   - Make sure that your system meets the hardware and software requirements. For details, see "System requirements" on page 8.
   - Review the information here: "Before you install" on page 9.
   - Installation must be performed at the destination machine. Installation using terminal service is not supported.

2. **Run the setup**

   - **Wizard installation:** To run the installation using the Setup wizard, see "Install components using the Setup wizard" on the next page.
   - **Silent installation:** To perform a silent installation, see "Install components silently" on page 18.

   After you run the setup, run any patch installations if necessary.

3. **Install a language pack (optional)**

   After you complete the installation, you can install a localized version to view the user interface in your local language. For details, see "Install a language pack" on page 25.

4. **Complete post-installation configuration**

Configure Vusers to run on a load generator machine without the need to log on manually. For details, see "Configure user login settings" on page 26.

Upgrade your installation

OpenText Professional Performance Engineering is a full installation that can be installed over any previous version, 12.5x and later.

To upgrade all components in your Windows installation using the Setup wizard, follow the installation process as described in "Install components using the Setup wizard" below. The installation process detects the older version and gives you the option to upgrade.

For silent upgrade, see "Install components silently" on page 18.

Install components using the Setup wizard

This section describes how to install the full version of OpenText Professional Performance Engineering , standalone components, or additional components using the Setup wizard.

Setup wizard information

One or more of the following options may be displayed when the starting the Setup wizard, depending on the components that you are installing.

Review the following information to help you successfully install the component. For more information on each component, refer to the Help Center.

OpenText Professional Performance Engineering Agent	Select the **Start OpenText Professional Performance Engineering Agent after installation** option to start the agent on a load generator immediately after installation. The agent enables communication between the load generator and Controller.
Agent mode	During the installation of the OneLG load generator, MI Listener, or Monitors over Firewall components, you are prompted to select the mode for the installed agent : • **OpenText Professional Performance Engineering** mode. The agent runs as a regular process, requiring log in before the process can start. **Note:** Select this mode if you are installing a load generator for use with OpenText Business Process Monitoring (BPM). • **OpenText Enterprise Performance Engineering** mode. The agent runs as a service, and is launched automatically when the operating system starts. • **OpenText Core Performance Engineering** mode. This is relevant for OneLG installation only. Select if you want to use the load generator with OpenText Core Performance Engineering. Agent configuration can be changed after installation, if required.
Gatling	Select the **Install Gatling after installation** option to install Gatling on a load generator that will be used to run Gatling scripts. This installs the version of Gatling included with OpenText Professional Performance Engineering . **Note:** For manual installation, see Gatling tests.
JMeter	Select the **Install JMeter after installation** option to install JMeter on a load generator that will be used to run JMeter scripts. This installs the version of JMeter version included with OpenText Professional Performance Engineering . **Note:** For manual installation, see JMeter tests.

Running the installation

The following procedure guides you through the Windows installation process.

To install components:

1. Make sure the Windows Update process is not running on your machine.
2. Run the **setup.exe** file in the root folder of the installation package.

   The installation program begins and displays the installation options.
3. Select the required installation option. For details on each of the available components, see" Product components" on page 6.

   For the machine that runs the load testing scenarios, select **OpenText Professional Performance Engineering Full Setup**. The Full Setup includes the following components:

   - VuGen
   - Controller
   - Analysis
   - OneLG load generator
   - Network Virtualization (optional)

   > **Note:** The Setup wizard can install either OpenText Professional Performance Engineering or OpenText Enterprise Performance Engineering. Make sure to select **OpenText Professional Performance Engineering**.
4. Specific software needs to be installed on Windows before you can perform the installation (see the Support matrix). If the prerequisite software is not already installed on your computer, a dialog box opens displaying the list of prerequisite programs that are required.

   Click **OK** to install the listed software before continuing with the installation. If you click **Cancel**, the Setup wizard exits because OpenText Professional Performance Engineering cannot be installed without the prerequisite software.

   After installing prerequisite software, the machine might restart. If that occurs, rerun **setup.exe**.
5. Perform the installation.

The Setup wizard opens, displaying the Welcome page. Follow the instructions in the wizard to complete the installation.

During installation of certain components, you have the option to install CA and TLS (SSL) certificates that will be used by the OpenText Professional Performance Engineering Agent. For details and requirements for these certificates, see "Using TLS (SSL) certificates" on page 8.

> **Note:**
>
> - The default installation path is: **C:\Program Files (x86)\OpenText\**
> - The installation path cannot contain non-English characters.

6. After the installation is complete, you can choose to install Network Virtualization (NV). For details on the NV installation modes (**Typical** or **Custom**), see "Install NV using the Setup wizard" on page 60.

7. If there is a patch available for the installed version:

   a. Run the patch installation file provided with the installation package.

   b. Follow the on-screen instructions.

8. If you installed NV, perform the post-installation tasks. For details, see "NV post-installation tasks" on page 62.

> **Note:**
>
> - To change, repair, or remove (uninstall) your installation, run the **setup.exe** file located in the root directory of the installation package. Select **OpenText Professional Performance Engineering Full Setup**, and then select the relevant operation in the Setup dialog.
> - For a list of the components that were registered during setup, such as **DLL** and **OCX** files, see the **RegisteredComponents_OpenText <product>.txt** file in the **build_info** folder of the installation.

Install components silently

A silent installation is an installation that is performed without the need for user interaction. You use the command line to run the setup files. For details, see "Installation command line options" on page 21.

> **Note:**
>
> - Use standard MSI command line options to define installation properties. For example, use INSTALLDIR to specify an alternate installation folder.
> - To prevent the OpenText Professional Performance Engineering Agent on the load generator from starting immediately after installation, add the following to the command line command: `START_LGA=0`. The agent enables communication between the load generator and Controller. For more details on the agent , see the Help Center.

To perform a silent installation:

1. Run one of the following commands from the command line:

   - To install all of the OpenText Professional Performance Engineering components including the prerequisite software, using a single command:

     ```
     <Installation_package>\lrunner\<your_language_
     folder>\setup.exe /s
     ```

   - To first install all of the prerequisites, and then OpenText Professional Performance Engineering , run the following commands:

     ```
     <Installation_package>\lrunner\<language_folder>\setup.exe
     /InstallOnlyPrerequisite /s
     ```

     ```
     msiexec.exe /qn /i "<Installation_
     package>\lrunner\MSI\LoadRunner_x64.msi"
     ```

   - By default, Network Virtualization is installed during the OpenText Professional Performance Engineering installation. To facilitate this installation, add the following to the installation command:

```
REBOOT_IF_NEED=1
```

- To prevent the installation program from installing Network Virtualization, add the following to the installation command:

  ```
  NVINSTALL=N
  ```

- To install Gatling or JMeter as part of the OneLG installation, add the following to the installation command:

  Gatling:

  ```
  INSTALL_GATLING=1
  ```

  JMeter:

  ```
  INSTALL_JMETER=1
  ```

- To secure your environment with TLS certificates and/or two-way TLS for OpenText Professional Performance Engineering products, OneLG, MiListener, or MoFW, add the following to the installation command as relevant:

  ```
  CACERT = <path to the root certificate in *.cer format>
  CERT = <path to the server certificate in *.cer format>
  CACERT_PK = <the CA certificate private key>
  CACERT_PK_PWD = <password for the CA certificate encrypted
  private key>
  CERT_PK = <the TLS certificate private key>
  CERT_PK_PWD = <password for the TLS certificate encrypted
  private key>
  TWO_WAY_SSL = High | Medium
  ```

  > **Example:** The following command performs a silent installation of MI Listener.

```
 SetupMIListener.exe /s /a /s INSTALLDIR="C:\Program Files
(x86)\OpenText\MI Listener" IS_RUNAS_SERVICE=0 START_LGA=1
NVINSTALL=N CACERT=c:\temp\root.cer CERT=server.cer TWO_WAY_
SSL=Medium
```

For certificate requirements, see "Using TLS (SSL) certificates" on page 8.

- To install an OpenText Professional Performance Engineering standalone application (not VuGen or OneLG):

  ```
  <Installation_package>\Standalone
  Applications\Setup<component_name>.exe /s /a /s
  ```

- To install VuGen and OneLG:

  ```
  <Installation_package>\Standalone Applications\SetupVuGen.exe
  -s -sp"/s"
  ```

  ```
  <Installation_package>\Standalone Applications\SetupOneLG.exe
  -s -sp"/s"
  ```

- To install additional components:

  ```
  <Installation_package>\Additional Components\<setup_file_path>
  /s /a /s
  ```

> **Tip:** See additional examples here: "Examples " on page 24

2. If you are installing an upgrade, run the following command:

   ```
   msiexec.exe /update <full path to msp file> [/qn] [/l*vx <full
   path to log file>]
   ```

   The msp files are located in the installation package.

   For details, see "Upgrade your installation" on page 14.

3. If you installed NV as part of the installation, perform the post-installation tasks. For details, see "NV post-installation tasks" on page 62.

Installation command line options

You can use the Windows command line to install the products. The CLI uses Wrapper file (setup.exe) or Package for the Web files (<PFTW>.exe).

- The full installation can be launched using the setup.exe file. For details, see "Installing the full package from the wrapper file" below.
- The installations of standalone applications and additional components can be launched using PFTW files. For details, see "Installing from a PFTW file" on page 23.

Installing the full package from the wrapper file

You can install full OpenText Professional Performance Engineering from the installation wrapper file, **setup.exe**, located in **\lrunner\<your_language>** folder of your installation package. The following command line options are available:

Option	Description
/s	Runs the installation in the background (silently), with no user interaction.
/qb	Runs the installation in unattended mode, with limited user interaction.
/InstallOnlyPrerequisite	Installs only the prerequisites – does not install any OpenText Professional Performance Engineering components. By default, the setup program checks that your machine has the required prerequisites, and installs them if necessary, before installing the components.

You can set public properties for the command line installation, using the following syntax:

```
setup.exe PROPERTY_NAME="value"
```

The following properties are available:

Property name	Description
INSTALLDIR="*your_path*"	Specifies the location where the application will be installed.
REBOOT_IF_NEED	1: Reboots the machine after installation, if required. This is recommended if you include the NV installation. 0: Does not reboot the machine after installation. Default: 0
NVINSTALL	Empty string: Excludes the installation of the NV component. Y: Includes the installation of the NV component. Default: Y
INSTALL_NV_MODE	1: Sets the NV installation mode to **Typical**. 2: Sets the NV installation mode to **Custom**. Default: Typical
INSTALL_DIR="*your_path*"	Specifies a path for the installation.
START_LGA	Empty string: Instructs the machine not to start the load generator after installation. 1: Instructs the machine to start the load generator after installation. Default: 1
IS_RUNAS_SERVICE	0: OpenText Professional Performance Engineering mode, runs the load generator's agent as a process. 1: OpenText Enterprise Performance Engineering mode, runs the load generator's agent as a service. 2: (Relevant for OneLG only) OpenText Core Performance Engineering mode, runs the load generator's agent as a service, and triggers launch of agent configuration at the end of the installation. Default: 1

Property name	Description
INSTALL_ GATLING	0: Does not install Gatling on the load generator. 1: Installs Gatling on the load generator. Default: 0
INSTALL_ JMETER	0: Does not install JMeter on the load generator. 1: Installs JMeter on the load generator. Default: 0
SKIP_CHECK_ PRIVILEGE	0: The installation process checks the install path for access privileges. 1: The installation process skips the access check for the install path. Default: 0

Example:

The following command performs the installation silently and then reboots the machine if necessary:

```
DVD\lrunner\en\setup.exe /s REBOOT_IF_NEED="1"
```

Installing from a PFTW file

You can install an OpenText Professional Performance Engineering standalone application or one of the additional components using the Package for the web (PFTW) files that are included with the installation media. The installation files are located under the **Standalone Applications** or **Additional Components** folders on the installation media.

Double-click on the packaged file, or run it from the command line using the following options:

Option	Description
/s	Runs the installation in the background (silently), with no user interaction.
/e	Only extracts the installation files; does not run them.
/f	Specifies the path of a temporary folder for file extraction. For example: `/f "c:\my_temp_folder"` If you leave out this option, the default temporary folder is used.
/a	Allows you to pass parameters or properties to the autorun file, such as setup.exe. Use the properties defined above (in the wrapper file section) for the setup.exe file.

> **Note:** Silent installation for VuGen and OneLG: If you need to insert a quotation mark character into the command line parameter, use two consecutive quotation marks.

Examples

The following commands silently run setup, and install the application in the specified folder. Where relevant, they also start the agent after installation:

- Examples for **SetupVuGen.exe** and **SetupOneLG.exe**:

```
SetupOneLG.exe -s -sp"/s" INSTALLDIR=""C:\Program Files
(x86)\OpenText\LoadRunner OneLG"" IS_RUNAS_SERVICE=0 START_LGA=1
NVINSTALL=N"
```

```
SetupVuGen.exe -s -sp"/s" INSTALLDIR=C:\VuGen"
```

- Examples for **other standalone applications**:

```
SetupAnalysis.exe /s /a /s INSTALLDIR="C:\Analysis"
```

```
SetupMIListener.exe /s /a /s INSTALLDIR="C:\MiListener"
```

```
SetupMoFW.exe /s /a /s INSTALLDIR="C:\MoFW"
```

Install a language pack

The language packs enable you to view the OpenText Professional Performance Engineering user interface in your local language. You install the language pack for the relevant component: OpenText Professional Performance Engineering full, VuGen standalone, or Analysis standalone from the installation package.

For a list of supported languages, see the Support matrix.

The language packs are supported for native non-English operating systems only.

> **Important:** The native language of the operating system must be the same as the language pack you are installing. For example, the Spanish language pack must be installed on a machine with a native Spanish operating system.

To install a language pack:

1. Make sure that OpenText Professional Performance Engineering in English is already installed.
2. In the root folder of the installation package, run **setup.exe**. The installation program begins and displays the installation options.
3. Click **Language Packs**. The Language Packs folder in the installation package opens.
4. Navigate to the folder for the language and component you want to install and run the installation file. For example, to install the French language pack on a VuGen standalone machine, the path is **..\Language Packs\French\VugenSA\Vugen_FRA.msp**.
   Follow the online instructions.

Configure user login settings

By default, you need to manually log on to a computer before running Vusers on that computer. However, you can configure a load generator machine to run Vusers, without the need to manually log on to the machine.

To configure user login settings:

1. In the Windows Start menu, select **OpenText Professional Performance Engineering AgentRuntime Settings Configuration**.

   The agent Runtime Settings dialog box opens.

2. Select one of the following options:

   - **Allow virtual users to run on this machine without user login.** This provides automatic login to the network from the load generator machine, so the Vusers can run without any manual intervention. Enter the network domain where the user machine resides, a user name, and password.

     > **Note:** When created, the agent service starts with the **LocalSystem** account (not as a specified user). The specified credentials are used by the agent service to start the **mdrv.exe** process when you run the script.
     >
     > You must reboot and log on to the system at least once after the OpenText Professional Performance Engineering installation before the automatic login can work.

   - **Manual log in to this machine.** The user must manually log on to the network from the load generator machine for each session of running Vusers.

3. Click **OK**.

Installing a load generator

Load generators are the machines that run Vuser scripts to generate load. To create a load generator, you install the load generator software on a host computer. Load generators can be either Windows-based or Linux-based. You use the Windows-based Controller to control both the Windows-based and the Linux-based Vusers.

You can install a load generator using the UI-based setup wizard, perform a silent installation, or deploy a load generator using a Docker container. Alternatively, you can provision a load generator in a cloud account. For details, see Load generators on the cloud in the Help Center.

This chapter includes:

Install a load generator on Windows

To install a load generator on Windows, you use the same installation process as other OpenText Professional Performance Engineering components. For details, see "Installing components on Windows" on page 12.

To deploy a load generator on Windows using a Docker container, see "Deploy Dockerized load generators on Windows" on the next page.

Deploy Dockerized load generators on Windows

This section describes how to run a Dockerized load generator on a Windows platform. Docker is a platform that allows you to develop, ship, and run applications using a container.

For supported protocols on Dockerized load generators, see the Supported Protocols guide.

> **Note:** FTP is not fully supported with load generator for Docker on Windows. Passive mode FTP is supported, but Active mode FTP is not.

Prerequisites

- Install Docker on the target machine, along with its dependencies, and set up the target machine environment as required. Currently, only the 64-bit version is supported. For installation details, see the Docker online documentation.
- Pull the Windows load generator Docker image from the from the relevant page, accessible from the performance testing page (https://hub.docker.com/u/performancetesting) in the Docker hub. Use the following command and appropriate **<tag version number>**, for example, 23.1:

  ```
  docker pull performancetesting/opentext_onelg_windows:<tag version
  number>
  ```

  > **Note:** The Docker image for the OneLG load generator replaces the previous Windows standalone load generator docker image.

Run a Dockerized load generator using the predefined image

Use the ready-to-use image to run a load generator (OneLG) on Docker for Windows. If you need customization for your container, for example, for Java or to run under a specific user, see "Run a Dockerized load generator using a custom image" below.

To run a Dockerized load generator:

Run the load generator container using the following command:

```
docker run -id -p <host_port>:54345 performancetesting/opentext_
onelg_windows:<tag version number>
```

> **Note:** Check that the <host_port> on the machine is available and allows incoming requests. You specify this port on the Controller side when connecting to this load generator.

Run a Dockerized load generator using a custom image

If your environment requires customized settings for running the container, you can create a Dockerfile to build a custom image for Docker on Windows.

Examples for custom images:

- To use a specific user account for the processes under which the Vusers are running, to provide support for accessing network resources like script parameter files. After running, the container should be able to verify the user.
- To run Java protocols on Windows load generator containers.
- To define environment variables for proxy server host and port.

To run a custom Dockerized load generator:

1. Create a new folder, and within it create a file named **dockerfile**. Paste the following **FROM** line into the file, using the appropriate OpenText Professional Performance Engineering version for the **<tag version number>**, and add the relevant customization lines:

   ```
   FROM performancetesting/opentext_onelg_windows:<tag version
   number>
   ```

```
<Customization lines>
```

For customization examples, see "Examples of customized content for Dockerfiles" below

> **Tip:** For information on commands that can be used in Docker files, see the Docker online documentation.

2. Save the Dockerfile.
3. Open a command line at the **dockerfile** folder path and run the following command, using the name you want for your custom image:

```
docker build -t <custom image name> .
```

4. Create a container for each load generator you want to use, by running the following command (or use any Docker orchestrator tool for running containers):

```
docker run -id -p <host_port>:54345 <custom image name>
```

If the custom image in step 3 was built with a tag then include it in the command:

```
docker run -id -p <host_port>:54345 <custom image name>:<tag
version number>
```

> **Note:** Check that the <host_port> on the machine is available and allows incoming requests. You specify this port on the Controller side when connecting to this load generator.

Examples of customized content for Dockerfiles

Example for Vusers under a specified user account

The following gives an example of dockerfile content for running the Vusers under a specified user account with network access to shared locations. Replace the values between **<>** with credentials for a valid user account in your environment, with network access to the shared resources.

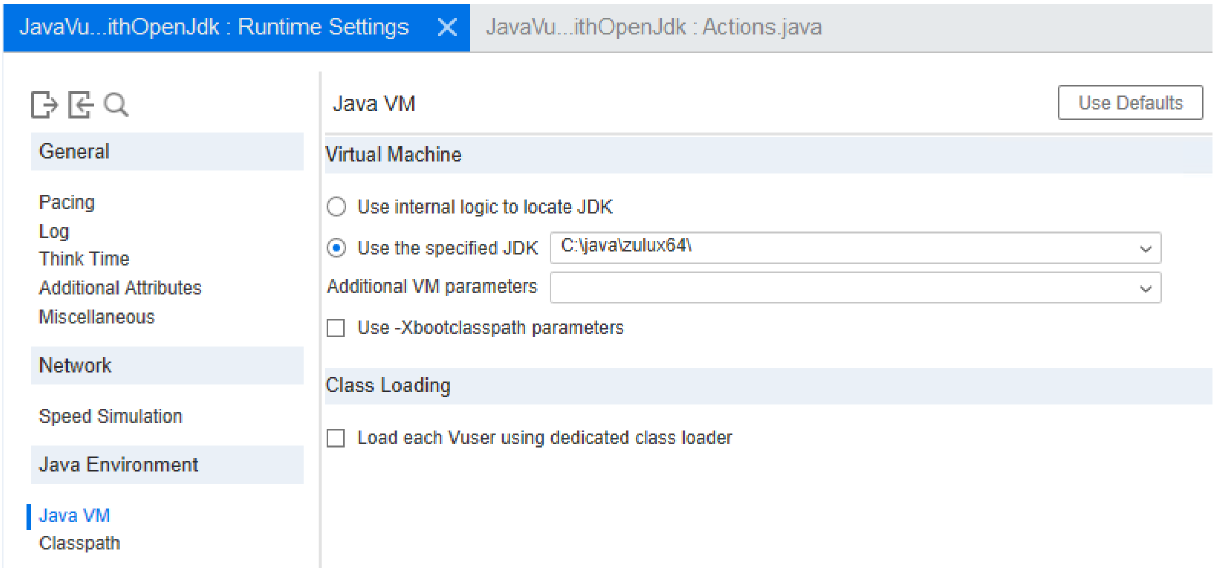
```
Example:
#escape=`
FROM performancetesting/opentext_onelg_windows:<tag version
number>
RUN c:\LG\launch_service\bin\magentservice.exe -remove
RUN c:\LG\launch_service\bin\magentservice -install
<domain>\<user name> <password>
```

Example for running Java/JMeter/Gatling protocols

The following gives an example of dockerfile content to run Java, JMeter, or Gatling protocols:

```
Example:
#escape=`
FROM performancetesting/opentext_onelg_windows:<tag version
number>
COPY .\<folder contains JDK> <target path in the container>
```

- The path to the target JDK directory defined in the **COPY** line for the **<target path in the container>** must also be added to the **Java VM** runtime settings page:



Java VM

Virtual Machine

○ Use internal logic to locate JDK

● Use the specified JDK: C:\java\zulux64\

Additional VM parameters:

□ Use -Xbootclasspath parameters

Class Loading

□ Load each Vuser using dedicated class loader

- For **Java 64-bit protocol** testing, include the following command line in the dockerfile, in order to add the path to the JDK 64-bit **bin** folder to the machine PATH environment variable:

  ```
  RUN powershell [Environment]::SetEnvironmentVariable(\"Path\",
  $env:Path + \";<target JDK path in the container>\bin\",
  [EnvironmentVariableTarget]::Machine)
  ```

- For **JMeter/Gatling protocol** testing, include the following command line in the dockerfile, in order to add the path for the JDK/JRE to the machine JAVA_HOME environment variable:

  ```
  RUN powershell [Environment]::SetEnvironmentVariable(\"JAVA_
  HOME\", \"<target JDK/JRE path in the container>\",
  [EnvironmentVariableTarget]::Machine)
  ```

After running the load generator containers

Add the load generators containers to scenarios. For details, see Configure Dockerized load generators in the Help Center.

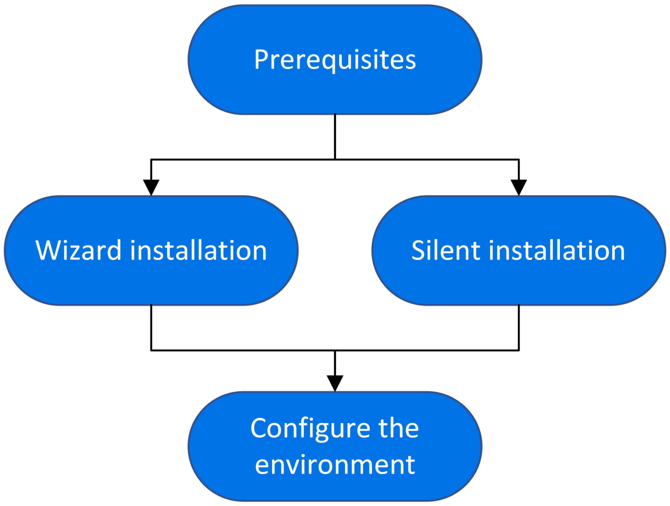
Tips and guidelines

- Dockerized load generators, run from the predefined image, are not supported when running over a firewall.
- Use `docker ps` to list the containers that are running.
- To stop the load generator service:
  - Use `docker stop <load generator container name or ID>` if you want to reuse the same load generator.
  - Use `docker rm -f <load generator container name or ID>` in order to remove the load generator container.
- To access the host network directly, use `--net=host` in place of `-p <host_port>:54345`. We recommend you use this flag if the AUT generates a lot of network activity.

Installation workflow on Linux

The following diagram shows the primary steps in installing the load generator on a Linux platform using the standard installation program.

For details on deploying a load generator on Linux using a Docker container, see "Deploy Dockerized load generators on Linux" on page 37.



1. **Prerequisites**
   - Make sure that your system meets the requirements, as described in "System requirements and prerequisites" on page 8.
   - If you are using a **RHEL 8** operating system: By default, libnsl.so.1 is not installed with RHEL 8.0, so you need to manually install it before installing the load generator, or installation will fail. Install using the package manager: `yum install libnsl.i686`
   - If you plan to replay **ODBC Protocol** scripts on the Linux machines: Install unixODBC v.2.3.1 or higher.
   - The load generator for this version can be installed over any load generator installation from version 12.56 and later. If load generator version 12.55 or earlier is installed, you must first uninstall it. For details, see "Uninstall load generator 12.55 or earlier" on page 52.

2. **Run the setup**

   - **Wizard installation:** To run the installation using the Setup wizard, see "Install a load generator on Linux with the Setup wizard" below.
   - **Silent installation:** To perform a silent installation, see "Silent load generator installation on Linux" on page 36.

3. **Configure your environment**

   After the installation, before you can begin working with the load generator you need to configure your environment. This involves setting the appropriate environment variables, checking access to the load generator, and verifying the installation. See "Configure the Linux environment" on page 42.

Install a load generator on Linux with the Setup wizard

This section describes how to install the load generator on a Linux platform using the Setup wizard.

To install the load generator using the Setup wizard:

1. Switch to super user.
2. If load generator version **12.55 or earlier** is installed on the machine, you must uninstall it manually. If you do not uninstall the load generator, you will get an error during the installation. For details, see "Uninstall load generator 12.55 or earlier" on page 52.

   If load generator **12.56 or later** is installed, the Setup wizard uninstalls it during the installation process, then reinstalls it.
3. Unpack the load generator installation package and open the location of the **.bin** file.

4. Launch the Setup wizard:

For:	Type:
[sh and bash shells]	`source ./installer.sh`
[csh and tcsh shells]	`source ./installer.csh`
If the source command is not supported by the current shell, use the "dot" command.	For example: `. ./installer.csh`

> **Note:** It is recommended that you use the **source** command to run the Setup wizard, as shown above. If you run the Setup wizard without using this command, you must manually set the environment variables for the current shell session. For details, see "Set the environment variables" on page 43.

The Load Generator Setup wizard checks that the required prerequisite software is installed on the computer. If any prerequisite software is missing, a message is displayed, and the Setup wizard aborts. Install the required package. Then rerun the Setup wizard.

5. Follow the online instructions to install the load generator.

During installation, you have the option to install CA and TLS (SSL) certificates for the load generator. For details and requirements for these certificates, see "Using TLS (SSL) certificates" on page 8.

For more help during installation, see "Troubleshoot the load generator Linux installation" on page 53.

> **Note:** The default installation path is: **/opt/OT/OT_LoadGenerator/_OT_LoadGenerator_Installation**

6. Exit super user or switch to another user.

7. Configure your environment as described in "Configure the Linux environment" on page 42.

Silent load generator installation on Linux

This section describes how to perform a silent installation of the load generator on a Linux machine.

To perform a silent installation of the load generator:

1. Switch to super user.
2. If a load generator is installed on the machine, you must first uninstall it, otherwise the silent installation will fail. To uninstall the load generator, see "Uninstall the load generator from a Linux machine" on page 50.
3. Change directory to `/<path_to_installer_dvd>/load_generator_linux_x64/`.
4. Run the following command to silently install the load generator. You can add installation options as shown in the table below.

```
source ./installer.sh -i silent
```

Installation option	Possible values	Details
–DSTART_PRODUCT_AFTER_INSTALL	Yes, No	Launch the load generator at the end of the installation. **Default value:** Yes
-DINSTALL_GATLING	Yes, No	Install Gatling on the load generator. **Default value:** No
-DINSTALL_JMETER	Yes, No	Install JMeter on the load generator. **Default value:** No

> **Example:** The following command runs a silent installation of a load generator with JMeter. The load generator is not launched at the end of the installation.
>
> ```
> source ./installer.sh -i silent -DINSTALL_JMETER=Yes -DSTART_PRODUCT_AFTER_INSTALL=No
> ```

> **Note:** It is recommended that you use a **source** command to perform a silent installation, as shown above. If you perform the installation without using the **source** command, you will need to set up the environmental variables after installing the load generator. For details, see "Set the environment variables" on page 43.
>
> If the **source** command is not supported by the current shell, use the "dot" command. For example: `. ./installer.sh - i silent`.
>
> If you encounter an error during the installation, see "Troubleshoot the load generator Linux installation" on page 53.

Deploy Dockerized load generators on Linux

This section describes how to run a Dockerized load generator on a Linux distribution.

For supported protocols on Dockerized load generators, see the Supported Protocols guide.

Prerequisites

> **Note:** The Ubuntu image for the OneLG load generator replaces the previous Ubuntu load generator docker image.
>
> For information on OneLG, see Load generators in the Help Center.

- Install Docker on the target machine, along with its dependencies, and set up the target machine environment as required. Currently, only the 64-bit version is supported. For installation details, see the Docker online documentation.
- Obtain the predefined load generator Docker image. Two images are available for Linux: Ubuntu and RHEL.

  Pull the image from the from the relevant page, accessible from the performance testing page (https://hub.docker.com/u/performancetesting) in the Docker hub.

Use one of the following commands and appropriate **<tag version number>**, for example, 24.1:

Ubuntu

```
docker pull performancetesting/opentext_onelg_ubuntu:<tag version
number>
```

RHEL

```
podman pull performancetesting/opentext_onelg_rhel:<tag version
number>
```

Run a Dockerized load generator using the predefined image

Use the ready-to-use image to run a load generator on Docker for Linux. If you need customization for your container, for example, for proxy servers, see "Run a Dockerized load generator using a custom image" on page 40.

To run the Ubuntu OneLG image, the appropriate environment variable must be included in the run command: `ONELG_FLAVOR=1`

In addition, you can use the following environment variables to enable JMeter and Gatling on the load generator, if required:

- `ENABLE_JMETER`
- `ENABLE_GATLING`

To run a Dockerized load generator:

Run the load generator container using the appropriate command for Ubuntu or RHEL (to enable Gatling or JMeter, include the bold variables).

Check that the <host_port> on the Linux machine is available and allows incoming requests. You specify this port on the Controller side when connecting to this load generator.

- **Ubuntu**:

  ```
  docker run -id -p <host_port>:54345 -e "ONELG_FLAVOR=1" -e
  "ENABLE_GATLING=1" -e "ENABLE_JMETER=1" --net=host
  performancetesting/opentext_onelg_ubuntu:<tag version number>
  ```

- **RHEL:**

  ```
  podman run -id -p <host_port>:54345 -e "ENABLE_GATLING=1" -e
  "ENABLE_JMETER=1" performancetesting/opentext_onelg_rhel:<tag
  version number>
  ```

Example using SSH

The following gives a simple C# code example for running multiple load generator containers using SSH. There are container orchestrator tools which do the same, for example, Kubernetes, OpenShift, Docker Swarm.

```
using (var client = new SshClient(dockerHost, dockerHostUserName,
dockerHostPasswd))
{
  client.Connect();
  for (int i =0; i > numOfContainers; i++)
  {
       string command = "docker run -id -p " + lgInitialPort + i) +
":54345 -e "ONELG_FLAVOR=1" --net=host performancetesting/opentext_
onelg_ubuntu:<tag version number>";
       var terminal = client.RunCommand(command);
       if (terminal.ExistStatus != 0)
       {
       throw new Exception("Failed to create new Docker container");
       }
       Console.WriteLine("Docker LG with external port" + lgInitialPort +
i + "created.");
  }
   client.Disconnect();
}
```

Run a Dockerized load generator using a custom image

If your environment requires customized settings for running the container, for example for proxy servers, you can follow the procedure below to create a Dockerfile to build a custom image.

Alternatively, you can do the following: Start the container; once it is running, set up the load generator environment variables, then start the load generator manually inside the container.

To create a Dockerfile to run a custom Dockerized load generator:

1. Create a new folder, and within it create a file named **dockerfile**. Paste the **FROM** line, plus the required customization lines, into the file, using the appropriate OpenText Professional Performance Engineering version for the <tag version number>.

   For example, for Ubuntu:

   ```
   FROM performancetesting/opentext_onelg_ubuntu:<tag version
   number>
   ENV http_proxy http://my_proxy_name:port
   ```

   > **Note:** The above customization example is for a proxy. It defines an environment variable for the proxy server host and port in the target image.

2. Save dockerfile.
3. Open a command line at the dockerfile folder path and run the following command, using the name you want for your custom image:

   Linux-Ubuntu:

   ```
   docker build -t <custom image name> .
   ```

RHEL:

```
podman build -t <custom image name> .
```

4. Create a container for each load generator you want to use, by running the following command:

   Ubuntu:

   ```
   docker run -id -p <host_port>:54345 -e "ONELG_FLAVOR=1" --
   net=host <custom image name>
   ```

   RHEL:

   ```
   podman run -id -p <host_port>:54345 <custom image name>
   ```

   If the custom image in step 3 was built with a tag then include it in the command, for example:

   ```
   docker run -id -p <host_port>:54345 -e "ONELG_FLAVOR=1" --
   net=host <custom image name>:<tag version number>
   ```

   > **Note:** Check that the <host_port> on the Linux machine is available and allows incoming requests. You specify this port on the Controller side when connecting to this load generator.

After running the load generator containers

Add the load generators containers to scenarios. For details, see Configure Dockerized load generators in the Help Center.

Tips and guidelines

- Dockerized load generators that are run from the predefined image are not supported when running over a firewall. (Workaround for advanced users: You can develop your own Docker image with MI Listener support.)
- Use `docker ps` to list the containers that are running.
- To stop the load generator service:

  - Use `docker stop <load generator container name or ID>` if you want to reuse the same load generator.
  - Use `docker rm -f <load generator container name or ID>` in order to remove the load generator container.
- The Dockerfile container has an ENTRYPOINT section. The container first runs the commands in ENTRYPOINT. It sets up the environment and then starts the load generator. The command uses a While loop to wait for input, in order to keep the container from exiting. This behavior prevents you from accessing the container while it is running. Make sure to add **-i** while starting the container; otherwise the While loop will consume an excessive amount of CPU.
- If you need entry into the container, add an argument such as `--entrypoint=/bin/bash` when starting the container. After entering the container, set the load generator environments and start the load generator . You can then switch to the host using CTRL+p and CTRL+q while keeping the container running in the background. To access the container again, use the `docker attach container_id` command.
- To access the host network directly, use `--net=host` in place of `-p <host_port>:54345`. We recommend you use this flag if the AUT generates a lot of network activity.

Configure the Linux environment

This section describes the configuration steps you need to complete after installing the load generator, before you can begin working with the load generator.

To complete the setup process after installing the load generator:

1. Set the appropriate environment variables.

   See "Set the environment variables" on the next page.

> **Note:** If you used a **source** command to install the load generator, the Setup wizard automatically sets the appropriate environment variables, and there is no need to perform this step.

2. Verify the load generator installation.

   See "Verify the Linux installation" on the next page.

3. Start the load generator.

   ```
   .cd /opt/OT/OT_LoadGenerator/;source env.csh;cd bin;./m_daemon_
   setup -install
   ```

4. Check that Controller is able to access the load generator.

   For details, see "Check Controller connection" on page 46.

Set the environment variables

> **Note:** This topic is applicable only if you ran the Load Generator Setup wizard without using a **source** (or "dot") command. If you used these commands, there is no need to perform any of the procedures described below.

To enable the load generator to run, the following environment variables must be defined:

- **M_LROOT.** The location of the Linux load generator root folder.
- **PATH.** The location of the Linux load generator **bin** directory.
- **PRODUCT_DIR.** The location of the Linux load generator root folder.

The Load Generator Setup wizard performs the following tasks relating to the environment variables:

- Adds the environment variable definitions to the system-wide startup scripts.

  If the variable definitions were not correctly set during the setup, see "Troubleshoot the load generator Linux installation" on page 53 for possible solutions.

- Sets environment variables for the current shell session if the **source** command was used to run the Setup wizard.

  This topic describes how to set the environment variables for the current shell session if a **source** command was not used to run the Setup wizard.

To determine if environment variables are set, run **verify_generator** (see "Verify the Linux installation" below) or use the following command:

```
echo $M_LROOT
```

If the name of the load generator root folder is returned, then the environment variables are correctly set for current shell. If the name of the load generator root folder is not returned, then manually set the variables as described below.

To manually set the environment variables for the current shell session (if the **source** command was not used to run the Setup wizard), execute one of the following commands:

- Bash users:

  ```
  source <load generator root>/env.sh
  ```

- C Shell users:

  ```
  source <load generator root>/env.csh
  ```

Verify the Linux installation

The load generator installation includes a setup verification utility, **verify_generator**, that checks the load generator setup on your Linux machine. The verification utility checks environment variables and your startup scripts (**/etc/csh.cshrc**, **${HOME}/.cshrc** or **/etc/profile**, **${HOME}/.profile**) to verify that they are set up correctly.

Run verify_generator

It is recommended that you run the **verify_generator** utility after installing the load generator, before attempting to invoke the load generator.

> **Note:**
>
> - To run this command, you must be a "normal" user and not root user.
> - Before you run the **verify_generator** utility, make sure that you have set the DISPLAY environment variable on your machine.

To run verify_generator:

1. From the **<load generator root>/bin** folder, run the following command:

   ```
   ./verify_generator
   ```

   For example:

   ```
   /opt/OT/OT_LoadGenerator/bin/verify_generator
   ```

   If you want to receive detailed information about the checks, you can use the -v option, as follows:

   ```
   ./verify_generator -v
   ```

2. View the results.

   - If the settings are correct, **verify_generator** returns **OK**.
   - If any of the settings are incorrect, **verify_generator** returns **Failed**, and suggestions on how to correct the setup.

Verification checks

The **verify_generator** utility checks the following:

- All the prerequisite software is installed. (This check is performed for 64-bit installations only.)
- There are at least 128 file descriptors
- The `.rhosts` permissions have been defined properly: `-rw-r--r--`

- The host can be contacted by using rsh to the host. If not, it checks for the host name in .rhosts
- **M_LROOT** is defined
- **.cshrc** or **.profile** defines the correct **M_LROOT**
- `/etc/csh.cshrc`, `${HOME}/.cshrc` or `/etc/profile`, `${HOME}/.profile` defines the correct **M_LROOT**
- **.cshrc** or **.profile** exists in the home directory
- The current user is the owner of the **.cshrc** or **.profile**
- A Linux load generator installation exists in **$M_LROOT**
- The executables have executable permissions
- PATH contains **$M_LROOT/bin** and **/usr/bin**

Check Controller connection

If Controller is to connect remotely to the load generator using **rsh** (remote shell), you need to make sure that the load generator can be remotely accessed by Controller.

To check remote access:

1. On the load generator machine, locate the **.rhosts** file which is located in the user home directory.
2. In the **.rhosts** file, verify that Controller is included in the list of machines. If it is not listed, add it to the list.

If Controller still cannot connect to the load generator, contact your system administrator.

Connecting to a Linux load generator without using rsh

You can configure Controller to connect to the load generator without using **rsh**. In this case, you need to activate the agent daemon on the load generator, as described below.

How to connect to a Linux load generator without using **rsh**:

1. On the Linux load generator, run the agent daemon by entering the following command from **<load generator root>/bin**:

   ```
   ./m_daemon_setup -install
   ```

   This runs a daemon called **m_agent_daemon**, and if successful, you receive a message: **m_agent_daemon <process ID>**.

   The agent now keeps running, even if the user is logged off. You can stop the agent by using the command explained in step 9 below, or by rebooting the machine.

   > **Note:** If you look at the **m_agent_daemon[xxx].log** log file in the temp directory, you may see communication errors, even if the installation succeeded.

2. In Controller, select **Scenario > Load Generators**. The Load Generators dialog box opens.
3. Click **Add**. The Add New Load Generator dialog box opens.
4. In the **Name** box, enter the name of the computer on which the load generator is running.
5. From the **Platform** list, select **Linux**.
6. Click **More**.
7. Click the **Linux Environment** tab, and make sure that the **Don’t use RSH** check box is selected.
8. Connect as usual.
9. To stop the agent daemon, run the following command from the **<installdir>/bin** directory:

   ```
   ./m_daemon_setup -remove
   ```

   This stops the **m_agent_daemon** daemon, and if successful, you receive the message: **m_agent_daemon is down**.

Improve load generator performance

This section includes recommendations for improving load generator performance. You can increase the number of file descriptors, process entries, and amount of swap space by configuring the kernel.

> **Note:** Most operating systems using the Linux load generator have sufficient default file descriptors, process entries, and swap space, and rarely require reconfiguration.

Increase file descriptors

A load generator uses the following file descriptor resources:

- 14 file descriptors for the launch service
- 20 file descriptors for the agent
- 30 file descriptors for each Vuser driver. By default, there is a driver for every 50 Vusers.
- File descriptors for the running Vusers. Each Vuser requires two descriptors.

For example, to compute the number of file descriptors used in running 100 threaded Vusers, the load generator requires:

Descriptors	Purpose of the descriptors
14	For the launcher
20	For the agent
60	For 2 drivers (30 x 2, each one drives 50 Vusers)
200	For 100 Vusers (each Vuser requires 2)

Total: 294 file descriptors

If Vusers are run as processes instead of threads, one driver is run per Vuser. Therefore, each Vuser requires 30 file descriptors.

The procedure to increase the number of file descriptors differs between shells.

In the examples below, the number of descriptors is increased to the maximum of 1024.

- For sh and ksh users, type:

  ```
  ulimit -n 1024
  ```

- For csh users, type:

  ```
  limit descriptors 1024
  ```

Below is an alternate procedure to increase file descriptors. In this example, the number of descriptors is increased to the maximum of 8192.

1. Add the following line to the **/etc/security/limits.conf** file:

   ```
   hard nfile 8192
   ```

2. Add the following line to the **/etc/sysctl.conf** file:

   ```
   fs.file-max = 8192
   ```

3. Reboot the machine.

Increase process entries

Each Vuser requires several free process entries. To increase the number of process entries on your system, you must reconfigure the kernel.

This section describes how to reconfigure the kernel for Linux platforms.

1. Locate the **/etc/security/limits.conf** file.
2. Set the maximum number of processes in the limits file. Type:

   ```
   hard nproc 8192
   ```

3. Reboot the machine.

Increase swap space

Each Vuser requires swap space ranging in size from 200 KB to 4 MB. Before adding space to your system configuration, you should determine your paging requirements. For environments running programs with very large memory requirements, it is recommended to have paging space of four times the physical memory. If you do not have enough paging space, certain processes may be killed, and others will be unable to start.

Uninstall the load generator from a Linux machine

This section describes how to uninstall the load generator from a Linux machine.

This section includes:

- "Uninstall load generator 24.1 or later" below
- "Uninstall load generator 12.56 to 2023 R1" on the next page
- "Uninstall load generator 12.55 or earlier" on page 52

Uninstall load generator 24.1 or later

For installations of load generator (OneLG) 24.1 or later, the Setup wizard uninstalls it during the installation process, then reinstalls it.

You can also uninstall a load generator using a manual command, or silently.

To uninstall load generator version 24.1 or later:

1. Make sure that you are logged in as the same user who installed the load generator.
2. Make sure that the **m_agent_daemon** process is not running on the machine. If it is running, kill its process:

```
cd /opt/OT/OT_LoadGenerator/bin;./m_daemon_setup -kill;su -;
```

3. Change the current directory to the installation directory:

```
cd <path_to_installation_folder>/_OT_LoadGenerator_Installation
```

4. Switch to super user.
5. Uninstall the load generator:
   - **Manual uninstall:** Run the following command:

   ```
   sh ./Change_OT_LoadGenerator_Installation
   ```

   - **Silent uninstall:** Run the following command:

   ```
   sh ./Change_OT_LoadGenerator_Installation -i silent
   ```

Uninstall load generator 12.56 to 2023 R1

For installations of load generator versions 12.56 through 2023 R1 (including OneLG installations), the Setup wizard uninstalls it during the installation process, then reinstalls it.

You can also uninstall a load generator using a manual command, or silently.

To uninstall load generator version 12.56 or later:

1. Make sure that you are logged in as the same user who installed the load generator.
2. Make sure that the **m_agent_daemon** process is not running on the machine. If it is running, kill its process:

```
cd /opt/MF/MF_LoadGenerator/bin;./m_daemon_setup -kill;su -;
```

3. Change the current directory to the installation directory:

```
cd <path_to_installation_folder>/_MF_LoadGenerator_Installation
```

4. Switch to super user.
5. Uninstall the load generator:
   - **Manual uninstall:** Run the following command:

     ```
     sh ./Change_MF_LoadGenerator_Installation
     ```

   - **Silent uninstall:** Run the following command:

     ```
     sh ./Change_MF_LoadGenerator_Installation -i silent
     ```

Uninstall load generator 12.55 or earlier

You can uninstall load generator 12.55 or earlier versions using a manual command or silently.

To uninstall load generator version 12.55 or earlier:

1. Make sure that you are logged in as the same user who installed the load generator.
2. Make sure that the **m_agent_daemon** process is not running on the machine. If it is running, kill its process:

   ```
   cd /opt/HP/HP_LoadGenerator/bin;./m_daemon_setup -kill;su -;
   ```

3. Change the current directory to the installation directory:

   ```
   cd <path_to_installation_folder>/_HP_LoadGenerator_Installation
   ```

4. Switch to super user.
5. Uninstall the load generator:
   - **Manual uninstall:** run the following command:

     ```
     sh ./Change_HP_LoadGenerator_Installation
     ```

- **Silent uninstall:** Run the following command:

  ```
  sh ./Change_HP_LoadGenerator_Installation -i silent
  ```

Troubleshoot the load generator Linux installation

This section describes troubleshooting tasks relating to the setup of the Linux load generator.

This section includes:

- "Environment variables were not set correctly in the system-wide startup scripts" below
- "Error when installing the load generator on a Linux platform" on page 57
- "Environment variables are not unset after uninstalling the load generator" on page 57
- "Unable to run Vusers on the load generator" on page 58

Environment variables were not set correctly in the system-wide startup scripts

To enable the load generator to run, the system-wide startup scripts must be modified to set specific environment variables. The required modifications to the startup scripts are made by the Load Generator Setup wizard. If the startup scripts were not correctly modified during the setup of the load generator, you can manually make the required changes to the startup scripts as described below. The required changes differ slightly between C shell users, and Bourne and Korn shell users.

- **Manually modifying the startup scripts for C shell users**

During the load generator installation process, the wizard creates the **env.csh** script. This script includes the commands to set the required environment variables for C shell users. A sample **env.csh** script is shown below.

```
setenv PRODUCT_DIR <Load Generator installation directory>

setenv M_LROOT ${PRODUCT_DIR}

   if ( ! $?PATH ) then

      setenv PATH ""

   endif

setenv PATH ${M_LROOT}/bin:${PATH}"
```

Add the following line to the **/etc/csh.cshrc** or **~/.cshrc** startup script to execute the **env.csh** script during the shell startup:

```
source <Load Generator installation directory>/env.csh
```

For example:

```
source /opt/OT/OT_LoadGenerator/env.csh
```

The effect of making the above modification to the startup script is similar to the modifications that are made by the Setup wizard. A sample of the modifications that the Setup wizard makes to the **/etc/csh.cshrc** startup script is shown below:

```
# New environment setting added by OT_LoadGenerator on Wed Jan 30
16:20:10 IST 2020 2.

# The unmodified version of this file is saved in
/etc/.login1557000131.

# Do NOT modify these lines; they are used to uninstall.

setenv PRODUCT_DIR "/opt/OT/OT_LoadGenerator"

# End comments by InstallAnywhere on Wed Jan 30 16:20:10 IST 2020
2.
```

```
# New environment setting added by OT_LoadGenerator on Wed Jan 30 16:20:10 IST 2020 5.

# The unmodified version of this file is saved in /etc/.login1557000131.

# Do NOT modify these lines; they are used to uninstall.

setenv M_LROOT "/opt/OT/OT_LoadGenerator"

# End comments by InstallAnywhere on Wed Jan 30 16:20:10 IST 2020 5.

# New environment setting added by OT_LoadGenerator on Wed Jan 30 16:20:10 IST 2020 8.

# The unmodified version of this file is saved in /etc/.login1557000131.

# Do NOT modify these lines; they are used to uninstall.

if ( ! $?PATH ) then

setenv PATH ""

endif

setenv PATH "/opt/OT/OT_LoadGenerator/bin:${PATH}"

# End comments by InstallAnywhere on Wed Jan 30 16:20:10 IST 2020 8.
```

- **Manually modifying the startup scripts for Bourne and Korn shell users**

  During the load generator installation, the Setup wizard creates the **env.sh** script. This script includes commands to set the required environment variables for Bourne shell and Korn shell users.

  Add the following line to the **/etc/profile** or **~/.profile** startup script to execute the **env.sh** script during the shell startup:

  ```
  source <Load Generator installation directory>/env.sh
  ```

  For example:

```
source /opt/OT/OT_LoadGenerator/env.sh
```

The effect of making the above modification to the startup script is similar to the modifications that are made by the Setup wizard. A sample of the modifications that the Setup wizard makes to the **/etc/profile** startup script is shown below:

```
# New environment setting added by OT_LoadGenerator on Fri Jan 16 11:14:24 IST 2020 1.

# The unmodified version of this file is saved in /etc/profile1806316421.

# Do NOT modify these lines; they are used to uninstall.

PRODUCT_DIR=/opt/OT/OT_LoadGenerator

export PRODUCT_DIR

# End comments by InstallAnywhere on Fri Jan 16 11:14:24 IST 2020 1.

# New environment setting added by OT_LoadGenerator on Fri Jan 16 11:14:24 IST 2020 4.

# The unmodified version of this file is saved in /etc/profile1806316421.

# Do NOT modify these lines; they are used to uninstall.

M_LROOT=/opt/OT/OT_LoadGenerator

export M_LROOT

# End comments by InstallAnywhere on Fri Jan 16 11:14:24 IST 2020 4.

# New environment setting added by OT_LoadGenerator on Fri Jan 16 11:14:24 IST 2020 7.

# The unmodified version of this file is saved in /etc/profile1806316421.

# Do NOT modify these lines; they are used to uninstall.

PATH="/opt/OT/OT_LoadGenerator/bin:${PATH}"

export PATH
```

```
# End comments by InstallAnywhere on Fri Jan 16 11:14:24 IST 2020
7. Settings #PATH=${M_LROOT}/bin:$PATH; export PATH
```

Error when installing the load generator on a Linux platform

When you use the **source installer.sh** command to install the latest load generator version on a Linux machine on which a load generator was previously installed, you may receive the following error message:

"An error occurred while trying to manage the selected instance."

Solution:

1. Open the registry file **/var/.com.zerog.registry.xml** and locate the element "**product**" with attribute "**name**"="**OT_LoadGenerator**".

   For example:

   <product name="OT_LoadGenerator" id="77f695c1-1f0c-11b2-883d-c486a85f6555" version="20.00.0.0" copyright="2019" info_url="http://www.anyco.com" support_url="http://www.anyco.com" **location="/opt/OT/OT_LoadGenerator"** last_modified="2019-11-21 13:12:14">
2. Record the value of the "**location**" attribute.
3. Remove the entire directory that is referred to by the "**location**" attribute.
4. Delete the registry file **/var/.com.zerog.registry.xml**.
5. Rerun the **source installer.sh** command.

Environment variables are not unset after uninstalling the load generator

When you uninstalled the Linux load generator, the Setup wizard might not have unset the load generator environment variables (M_LROOT, PRODUCT_DIR, and PATH) for the current shell. To unset the environment variables, close the current shell session and invoke a new one, or manually unset the variables as described below:

- To unset the M_LROOT and PRODUCT_DIR variables:
  - [bash shells] Use the **unset** command.
  - [csh shells] Use the **unsetenv** command.
- To update the PATH variable to exclude the load generator binary directory, type:
  - [bash shells] `PATH=<required list of paths>; export PATH`
  - [csh shells] `setenv PATH <required list of paths>`

Unable to run Vusers on the load generator

If you are unable to run Vusers on the load generator, no specific error is reported, and the Vuser protocol requires a third-party application or client to be present on load generator side, check the dynamic libraries used by the application. This enables you to establish if any shared objects cannot be found. A shared object that cannot be found may indicate either a missing prerequisite package or an environment variable issue.

To check the dynamic libraries used by an application, type:

```
ldd application_name
```

For example, type `ldd mdrv` to determine if all the dependencies of the **mdrv** executable can be found. If any dependencies are not found, run **verify_generator** as described in "Verify the Linux installation" on page 44.

> **Note:** If you are running Vusers for a protocol that requires a client installation (for example, Oracle), make sure that the path for the client libraries is included in the dynamic library path environment variable (LD_LIBRARY_PATH or SHLIB_PATH).

Install Network Virtualization

Network Virtualization (NV) can be installed as part of the main installation process, or later as a separate process. You can install NV using the Setup wizard, or silently.

This chapter includes:

NV installation files

The NV installation files can be found in the **Additional Components/Network Virtualization** folder of the installation package. Install the required NV components on the machine, as follows.

Component	Required NV components	NV installation files
Full OpenText Professional Performance Engineering installation OpenText Enterprise Performance Engineering Host	• NV for Controller • NV for Load Generator and VuGen	• `NV4ControllerSetup.exe` • `NV4LGSetup.exe`
OpenText Enterprise Performance Engineering Server	NV for Enterprise Performance Engineering	`NV4ServerSetup`
VuGen	NV for Load Generator and VuGen	`NV4LGSetup.exe`
Load generator (standalone)	NV for Load Generator and VuGen	`NV4LGSetup.exe`

NV installation logs

NV installation logs are located in the **C:\Temp\NV_Logs** folder.

Install NV using the Setup wizard

Install NV using the Setup wizard. You can run the Setup wizard as part of component installation, or as a separate process.

To install NV using the Setup wizard:

1. Run the relevant installation file to open the wizard. For details, see "NV installation files" on the previous page.
2. Select an installation mode.
   - **Typical:** Installs with default options.
   - **Custom:** Enables you to configure the following options before installation. Some options are enabled only when running the first NV installation on a machine.

Option	Details
Destination Folder	Location of the NV program files. Default: C:\Program Files\OpenText\NV
Data Folder	Storage location for temporary internal application data and user data, such as test run results. Default: C:\ProgramData\OpenText\NV
Server Port	◦ **NV for Load Generator and VuGen:** The port that is used to retrieve NV statistics. ◦ **NV for Controller:** The port that is used for internal NV communication on the machine. Default: 8182
Enable Remote Connections	Adds an exception to Windows Firewall. Required for NV for Load Generator.

3. Follow the rest of the on-screen instructions to complete the installation.

Install NV silently

NV is installed by default when you install OpenText Professional Performance Engineering components silently. If you opted not to install NV at that time, you can run a silent NV installation.

To run a silent NV installation:

1. Prerequisites:
   - Make sure the Microsoft SmartScreen Filter has been turned off.
   - You must run the installation with administrator privileges.
2. Run the following command with the relevant command line options.

```
<installation_file> /s /v"/qn <command_line_options>"

Example: NV4LGSetup.exe /s /v"/qn PORT=8182"
```

Command line options:

Option	Details
PORT=<port number>	(Required) The port used to connect. Default value: `8182`
INSTALLDIR=\"<path to installation folder>\"	(Optional) The location where the application files will be installed. Default value: `C:\Program Files\OpenText\NV`
ENABLE_ REMOTE=<TRUE | FALSE>	(Optional) Opens the port in the firewall. Required for enabling remote access to web-based NV components. Default value: `TRUE`

Option	Details
REBOOT_IF_NEED=<TRUE | FALSE>	(Optional) If a reboot is required, automatically reboots the computer after installation completes. Even if set to FALSE, you need to reboot the machine before NV for Controller is operational. Default value: `FALSE`
DATA_FOLDER="<path to data dir>"	(Optional) The location where temporary internal application data and user data is saved. Default value: `C:\ProgramData\OpenText\NV`

Uninstall NV silently

To uninstall NV silently, run the following command:

```
<installation_file> /s /removeonly /v"/qn PORT=<port number> REBOOT_IF_NEED=<TRUE | FALSE>"

Example: NV4LGSetup.exe /s /removeonly /v"/qn PORT=8182 REBOOT_IF_NEED=<TRUE | FALSE>"
```

NV post-installation tasks

After installation, if you installed NV on load generators that have more than one network interface card, you must set the active adapter for each load generator. For details, see Customize NV settings in the Help Center.

Install the Data Hub and Web Connector

The Data Hub is a communication channel based on Kafka technology that enables more robust communication between OpenText Professional Performance Engineering components.

If your components communicate over the web (for example, if a load generator is behind a firewall and communicates using a proxy), you need to install the Data Hub Web Connector, in addition to the Data Hub.

The Data Hub and Web Connector must be installed on Linux. They can communicate with other components running on Linux and Windows machines.

You can download the Data Hub installation package from AppDelivery Marketplace. The Web Connector installation package is included in the Data Hub installation package.

After you install the Data Hub and Web Connector, you configure the components to work with them. For details, see Data Hub and Web Connector setup in the Help Center.

Installation process

The following procedure describes how to install using the Setup wizard. The Setup wizard is used for both Data Hub and Web Connector installations.

To perform a silent installation, see "Silent install for the Data Hub and Web Connector" on the next page.

To install the Data Hub/Web Connector:

1. **Prerequisites**
   - Make sure that your system meets the requirements, as detailed in the Support Matrix.
   - The Data Hub requires the latest version of Java JRE to be installed.

2. Switch to super user.
3. Unpack the relevant installation package and open the location of the **.bin** file.
4. Launch the Setup wizard: For [sh and bash shells], type `./inst64.bin`
5. Follow the online instructions to install the product.
    - For the Data Hub, the default installation path is: **/opt/OT/OT_LoadRunner_Data_Hub/_OT_LoadRunner_Data_Hub_Installation**
    - For the Web Connector, the default installation path is: **/opt/OT/OT_LoadRunner_Data_Hub_Web_Connector/_OT_LoadRunner_Data_Hub_Web_Connector_Installation**

    During installation, you have the option to install CA and TLS (SSL) certificates. For details and requirements for these certificates, see "Using TLS (SSL) certificates" on page 8.
6. Exit super user or switch to another user.
7. To verify successful installation, run the following commands to check the status of the relevant services. If these services are running, the installation was successful.
    - For Data Hub:

      ```
      service zookeeper status
      ```

      ```
      service kafka status
      ```

    - For Web Connector:

      ```
      service webconnector status
      ```

Silent install for the Data Hub and Web Connector

This section describes how to perform a silent installation of the Data Hub and Web Connector.

For general installation details, see "Install the Data Hub and Web Connector" on page 63.

Silent install for Data Hub

You can install the Data Hub silently by following the steps below.

To perform a silent installation of the Data Hub:

1. Switch to super user.
2. Unpack the Data Hub installation package.
3. Create a file named **installer.properties** using the following template, changing the parameters as needed.

> **Tip:** An **installer.properties** file is generated when you run the Setup wizard, and can be used again for later silent installations.

```
#Choose Install Folder
#---------------------
USER_INSTALL_DIR=/opt/OT/OT_LoadRunner_Data_Hub

#Choose Link Location
#--------------------
USER_SHORTCUTS=/usr/local/bin/OT/OT_LoadRunner_Data_Hub

#Install
#-------
-fileOverwrite_/opt/OT/OT_LoadRunner_Data_Hub/_OT_LoadRunner_
Data_Hub_Installation/Change_OT_LoadRunner_Data_Hub_
Installation.lax=Yes
-fileOverwrite_/opt/OT/OT_LoadRunner_Data_
Hub/zookeeper.service=Yes
-fileOverwrite_/opt/OT/OT_LoadRunner_Data_Hub/kafka.service=Yes
-fileOverwrite_/opt/OT/OT_LoadRunner_Data_Hub/run_after_
install.sh=Yes
```

```

#Install certificate
#-------------------
INSTALL_CERTIFICATE=\"\",\"No\"
```

4. Save the **installer.properties** file in the same location as the installer file.
5. Run the installation command using the following syntax:

   ```
   ./inst64.bin [-f<path to installer properties file> | -options]
   ```

   You can modify the installation by adding options to this command. For details, see "Silent installation options" on page 68.

> **Note:**
>
> - When the **installer.properties** file is saved in the same directory as the installer, it overrides all other command line options unless the `-f` option is used to specify another valid properties file.
> - The path to the installer properties file may be either absolute, or relative to the directory in which the installer resides.

For example, the following command installs the Data Hub silently using an **installer.properties** file that is saved in the same location as the installer:

> **Example:** `./inst64.bin -i silent`

Silent install for Web Connector

You can install the Web Connector silently by following the steps below.

To perform a silent installation of the Web Connector:

1. Switch to super user.
2. Unpack the Web Connector installation package.
3. Create a file named **installer.properties** using the following template, changing the parameters as needed. The template here installs the Web Connector with the default certificates.

> **Tip:** An **installer.properties** file is generated when you run the Setup wizard, and can be used again for later silent installations.

```
#Choose Install Folder
#---------------------
USER_INSTALL_DIR=/opt/OT/OT_LoadRunner_Data_Hub_Web_Connector
#Choose Link Location
#--------------------
USER_SHORTCUTS=/usr/local/bin/OT/OT_LoadRunner_Data_Hub_Web_Connector
#Install
#-------
-fileOverwrite_/opt/OT/OT_LoadRunner_Data_Hub_Web_Connector/_OT_LoadRunner_Data_Hub_Web_Connector_Installation/Change_OT_LoadRunner_Data_Hub_Web_Connector_Installation.lax=Yes
-fileOverwrite_/opt/OT/OT_LoadRunner_Data_Hub_Web_Connector/webconnector.service=Yes
-fileOverwrite_/opt/OT/OT_LoadRunner_Data_Hub_Web_Connector/run_after_install.sh=Yes
-fileOverwrite_/opt/OT/OT_LoadRunner_Data_Hub_Web_Connector/webconnector.sh=Yes
#Install certificate
#-------------------
INSTALL_CERTIFICATE=\"\",\"No\"
```

4. Save the **installer.properties** file in the same location as the installer file.
5. Run the installation command using the following syntax:

   ```
   ./inst64.bin [-f<path to installer properties file> | -options]
   ```

   You can modify the installation by adding options to this command. For details, see "Silent installation options" on the next page.

> **Note:**
> - When the **installer.properties** file is saved in the same directory as the installer, it overrides all other command line options unless the `-f` option is

> used to specify another valid properties file.
>
> - The path to the installer properties file may be either absolute, or relative to the directory in which the installer resides.

For example, the following command installs the Web Connector silently using an **installer.properties** file that is saved in the same location as the installer:

> **Example:** `./inst64.bin -i silent`

Silent installation options

You can add the following options to the silent installation command:

Option	Description
-i [swing | console | silent]	Specifies the user interface mode for the installer.
-D<name>=<value>	Specifies the installer properties.
-r <path to generate response file>	Generates a response file.

The following options can be used to repair or uninstall:

Option	Description
-repair	Repairs the installation.
-uninstall	Uninstalls the Data Hub or Web Connector.