Legal Notices

Isight, SIMULIA Execution Engine, the 3DS logo, and SIMULIA are commercial trademarks or registered trademarks of Dassault Systèmes or its subsidiaries in the United States, and/or other countries. Use of any Dassault Systèmes or its subsidiaries trademarks is subject to their express written approval. Other company, product, and service names may be trademarks or service marks of their respective owners.

Isight, SIMULIA Execution Engine, and this documentation may be used or reproduced only in accordance with the terms of the software license agreement signed by the customer, or, absent such an agreement, the then current software license agreement to which the documentation relates.

This documentation and the software described in this documentation are subject to change without prior notice.

Dassault Systèmes and its subsidiaries shall not be responsible for the consequences of any errors or omissions that may appear in this documentation.

© Dassault Systèmes, 2015.

Other company, product, and service names may be trademarks or service marks of their respective owners. For additional information concerning trademarks, copyrights, and licenses, see the Legal Notices in the SIMULIA User Assistance 2016 HTML documentation.
# Contents

- About Station Affinities .................................................................................................................. 57
- Station Status Reporting in the Dashboard ....................................................................................... 57
- About Running Multiple Stations on a Single Host Computer ............................................................ 58
- Starting a SIMULIA Execution Engine Station .................................................................................. 58
- Shutting Down a SIMULIA Execution Engine Station ....................................................................... 60
- Restarting a SIMULIA Execution Engine Station Remotely ............................................................... 60
- Configuring SIMULIA Execution Engine Station Properties ............................................................ 60
- Using the Dashboard .......................................................................................................................... 74
- About the Dashboard Interface ........................................................................................................... 74
- Starting the Dashboard ....................................................................................................................... 75
- Viewing Connection Information ......................................................................................................... 79
- Viewing Station Information .............................................................................................................. 79
- Controlling Station Workitems ......................................................................................................... 79
- Shutting Down, Restarting, or Deleting a Station ................................................................................ 80
- Managing Access Control for SIMULIA Execution Engine Users .................................................... 80
- Managing Access Control – System Administration ........................................................................ 81
- Viewing License Usage Information .................................................................................................. 81
- Using the Command Line Client ....................................................................................................... 82
- Starting the Command Line Client .................................................................................................... 82

## Generating Reports of SIMULIA Execution Engine License Usage .................................................. 84
- About License Usage Reports .............................................................................................................. 84
- Running the licusage Utility ............................................................................................................... 84
- License Report Utility Options .......................................................................................................... 84
  - General Options ............................................................................................................................ 85
  - Export Options ............................................................................................................................. 87
  - Query Options ............................................................................................................................. 87
  - Output Options ............................................................................................................................ 87
  - Advanced Options ....................................................................................................................... 88
- Examples ............................................................................................................................................ 88

## Basic Troubleshooting ..................................................................................................................... 90
- User Login Names Containing Punctuation ....................................................................................... 90
- Log Files for the SIMULIA Execution Engine .................................................................................... 90
- Configuring the Windows Firewall .................................................................................................... 91
  - Configuring the Windows Firewall for TomEE ............................................................................. 91
- Resolving Publishing Errors on Windows .......................................................................................... 92
- Linux-based SIMULIA Execution Engine Stops Functioning Correctly ........................................... 92
- Changing Your SIMULIA Execution Engine Passwords ................................................................... 92
What You Need to Know Before Installing the SIMULIA Execution Engine

This section describes information that you should review prior to installing the SIMULIA Execution Engine.

About the SIMULIA Execution Engine Environment

The SIMULIA Execution Engine environment contains several components and interfaces, including a database, J2EE application server, and web browser.

An overview of the SIMULIA Execution Engine environment is shown in the following figure.

![SIMULIA Execution Engine Architecture](image)

Figure 1: SIMULIA Execution Engine Architecture

The items in the environment are described below.

SIMULIA Execution Engine

The SIMULIA Execution Engine is the nerve center that manages simulation process flow, job dispatching, distributed and parallel computing, results processing and archiving, library activity, and collaboration activities. The SIMULIA Execution Engine uses a commercial middleware layer consisting of a standard J2EE application server and relational database, and it exploits EJB, JMS, JTA, JDBC, Servlet, JSP, and other J2EE technologies. The database is tightly coupled to the application server and provides underlying storage of all data.

Clients

Clients include any applications that work with the SIMULIA Execution Engine in a client-server model. The client-server model of computing is a distributed application structure that partitions tasks or workloads between the providers of a resource or service, called servers, and service requesters, called clients.

SIMULIA Execution Engine Stations

SIMULIA Execution Engine stations are computers on the network that have been registered with the SIMULIA Execution Engine to provide services to the system and to handle the execution of work items. They consist of a framework for receiving work items, communicating with the library, executing components, and returning results.
WebTop

The SIMULIA Execution Engine WebTop is a Web browser-based interface that provides access to the SIMULIA Execution Engine system from across a network without any additional software installation on the client computer. It is ideal for users who want to perform only basic operations, to run models, and to see the results with minimal configuration on their part. For more information, see the *SIMULIA Execution Engine WebTop Guide*.

Dashboard

The SIMULIA Execution Engine Dashboard displays the current status of the SIMULIA Execution Engine. The dashboard shows a list of running stations, the number of running jobs, the work items on each station, and the licenses being used by the SIMULIA Execution Engine. For more information, see *Using the Dashboard*.

Command Line Client

The Command Line Client is a console (character mode) program that provides simple text-based access to most functions of the SIMULIA Execution Engine. For more information, see *Using the Command Line Client*.

Basic Installation Steps for the SIMULIA Execution Engine

This section highlights the basic steps necessary for installing the SIMULIA Execution Engine.

1. Install the SIMULIA Execution Engine software.
   For details, see *Installing the SIMULIA Execution Engine Server*.
2. Configure the SIMULIA Execution Engine.
   For details, see *Configuration, TomEE Configuration, and Configuring Security*.
3. Start the SIMULIA Execution Engine in the application server.

Installing as an Administrator (Windows) or Non-Root User (Linux)

You must be an Administrator user on Windows to install the SIMULIA Execution Engine. On Linux, however, root privileges are not required.

Windows

You must use an Administrator user account on Windows to install the SIMULIA Execution Engine.

Installing the license server for the SIMULIA Execution Engine also requires Administrator privileges.

Linux

Root (superuser) privileges are not required to install the SIMULIA Execution Engine on Linux, and it is recommended that you install the SIMULIA Execution Engine as a non-root user. However, root permissions are usually needed beforehand to grant the installing user write access to the installation directory; for example, in `/opt/SIMULIA/SEE/`

As a non-root user, you can install the SIMULIA Execution Engine in any directory to which you have write access. If an administrator creates a directory called `/opt/SIMULIA/SEE/` and makes it writable, a non-root user can also install in `/opt/`. 
What's New?

This section describes the new and enhanced functionality in the SIMULIA Execution Engine.

**New Functionality**

**DS Standard Installer**  
The DS Standard Installer now supports following modes of installation:

- Installing SIMULIA Execution Engine in Text User Interface (TUI) Mode
- Installing SIMULIA Execution Engine in Silent Mode

**TomEE with Oracle Database**  
The SIMULIA Execution Engine can now be run on TomEE with an Oracle database. See the following topics:

- Configuring an Oracle Database
- Configuring TomEE with Derby or Oracle

**Enhanced Functionality**

**Removed Functionality**

**Customizing Preferences**

**User Assistance**
Prerequisites

This section describes the prerequisites for installing the SIMULIA Execution Engine.

Software Requirements

The SIMULIA Execution Engine can only be run on certain platforms, databases, and Java application servers.

Operating System

The SIMULIA Execution Engine must be run on one of the supported operating systems.

The latest system configuration information, including supported platforms for SIMULIA Execution Engine stations, can be found under the SIMULIA Platforms & Configuration Support section at http://www.3ds.com/support/certified-hardware/simulia-system-information/.

Web Browser and PDF Viewer Software

The following software is needed for installing the software and viewing the SIMULIA Execution Engine documentation.

- Web browser. You need a web browser installed on your system to access and configure and Oracle. For a list of supported web browsers and versions, see the SIMULIA Platforms & Configuration Support section at http://www.3ds.com/support/certified-hardware/simulia-system-information/.
- PDF Viewer. You must have Adobe Acrobat Reader or some other PDF viewer installed to access the PDF format documentation.

The documentation is provided in both PDF and HTML formats.

The PDF files are located in the following directory:

<Isight_install_directory>/docs/

You can also access the documentation directly from the installation disk (prior to an installation).

Hardware Requirements

A DVD ROM drive is needed to install the SIMULIA Execution Engine software. If the installation system does not have a DVD drive, the entire contents of the install DVD can be copied to a network file system from a system that does have a DVD drive, and the installation can then be run from the network file system.

Another option is to mount the DVD on a Linux computer, export the /mnt directory as an NFS file system, and then mount that file system on the installation computer. This option is complicated. For further assistance, contact your local system administrator.

Updating the Windows Firewall

To ensure that your SIMULIA Execution Engine will function correctly and be able to communicate with other computers on your network, you need to update the settings of the Windows firewall.
Disabling Real-Time Virus Scanning

You should disable virus scanning software on the computer that is running the SIMULIA Execution Engine.

To ensure that the SIMULIA Execution Engine is running at an optimum level, it is recommended that you disable any real-time virus scanning that is executing on the system running the SIMULIA Execution Engine. For more information, contact your local system administrator.

Migrating to SIMULIA Execution Engine 2016

You do not have to uninstall SIMULIA Execution Engine 5.9 or a 5.9 station before you install SIMULIA Execution Engine 2016 or a 2016 station. However, you must follow a specific procedure to ensure that your upgrade is successful.

For the latest support information and tips on upgrading to the new release of the SIMULIA Execution Engine, search for “SIMULIA Execution Engine Migration Procedure” in the Dassault Systèmes DSX.ClientCare Knowledge Base at http://www.3ds.com/support/knowledge-base.
Installing the SIMULIA Execution Engine Server

This section describes how to install and uninstall the SIMULIA Execution Engine using Isight 2016 installer. SIMULIA Execution Engine can be installed on either Windows or Linux in the following modes:

• Interactive Installation Mode:

In an interactive installation mode, the setup procedure prompts you to answer questions until the installation is completed. You can use any of the two interactive modes of installation to install SIMULIA Execution Engine:

1. Graphical User Interface (GUI) mode
2. Text User Interface (TUI) mode

Even though the two methods are different, the list and the sequence of the questions are exactly same in both the cases.

• Silent Installation Mode:

You need to provide a previously generated response file to the setup, and the installation completes without any interaction.

Text User Interface mode and Silent mode are not supported for TomEE based SIMULIA Execution Engine installations.

You must belong to the Administrators group to install SIMULIA Execution Engine on Windows, or have the privileges assigned to the Administrators group; see Installing as an Administrator (Windows) or Non-Root User (Linux).

Root (superuser) privileges are not required to install SIMULIA Execution Engine on Linux. However, root permissions may be needed beforehand to grant the installing user write access to the installation directory; for example, in `/opt`.

Note: Installation will fail on Windows if the media unload directory names and/or the installation path exceeds certain limits, because of the overall limit of 260 characters imposed by Windows. During installation or unzip operations, subdirectories and files will be created in the specified directories. The longest full pathname length must be limited to well within the 260 characters limit. Therefore, both unload and installation paths should be kept reasonably short to guarantee that this limit is not reached.

Installing the SIMULIA Execution Engine in Graphical User Interface (GUI) Mode

In the GUI mode, the Isight 2016 installer prompts you to answer questions in dialog boxes. On Linux, the dialog boxes are based on Motif (or OpenMotif) and require a graphics card on the Linux machine.

Installing SIMULIA Execution Engine in GUI mode on Linux, also requires an X-Windows display, either local or remote over a network. If the DISPLAY environment variable is not set, the installer will not execute.

Before you begin: If necessary, start the license server software before starting the SIMULIA Execution Engine. Typically, this process is performed automatically by the license server installer. However, there are some cases when it must be done manually, including if you decide to provide a license at a later time. Be sure that you have acquired a license file before starting the license server.
1. Insert the SIMULIA Execution Engine installation DVD, or extract the archive file of the installation media and go to the directory:

On Windows:

<\dvd_media_dir>\SIMULIA_Isight\Windows64\1

On Linux:

<\dvd_media_dir>/SIMULIA_Isight/Linux64/1

2. Launch the GUI mode installation by running the appropriate command for your operating system.

On Windows:

setup.exe

On Linux:

./StartGUI.sh

3. (Windows only) If necessary, click Allow on the User Account Control dialog to confirm the execution of the installation program.

4. Click Next to advance through each panel of the installer, responding to the prompts as needed.

   • In the first panel, choose the destination directory into which the software will be installed.
   • In the second panel, select SIMULIA Execution Engine as the product you want to install:

        Isight Desktop
        SIMULIA Execution Engine
        SIMULIA Execution Engine Station

        You can choose any combination of these components to install on different computers and servers.

5. If you choose to install the SIMULIA Execution Engine with a TomEE application server, the TomEE Configuration Utility automatically starts as the last step of installation. This utility will fully configure TomEE and its associated database connection (either Oracle or Derby).

        You can also launch the TomEE Configuration Utility separately from (after) the installer—see Configuring TomEE with Derby or Oracle.

        The TomEE Configuration Utility will automatically create a .cpr connection profile file to connect stations to the SIMULIA Execution Engine server.

        After the installation is finished, you can use the Edit Logon Profile tool later to create additional .cpr connection profiles. See "Creating a SIMULIA Execution Engine Connection Profile" in the Isight User's Guide.

6. Select the type of license server software you will use, or skip the licensing selection for now:

        Dassault Systemes License Server
        FLEXnet License Server
        Skip this for now

        If you (or your system administrator) already have the FLEXnet or Dassault Systèmes license server installed and running, specify the server computer's host name and port number. This information is used to contact the license server and create a licensing client file that references the server. If you have installed a redundant license server triad, enter the host name and port for all three machines.

7. Verify the information on summary panel that corresponds to the entries made earlier (and make modifications if necessary). Click Close to exit the installer.

8. If you have chosen to use the TomEE Configuration Utility, see one of the following topics for details:

        • Configure TomEE with Derby
Installing SIMULIA Execution Engine in Text User Interface (TUI) Mode

In Text User Interface (TUI) mode, the Isight 2016 installer does not need a graphics card to be present in your machine. The installer prompts you to answer questions in text mode in your machine’s command-line interpreter (terminal).

Note the following while proceeding for the installation in TUI mode:

- If you want to continue with the default value, simply press Enter. If you want to modify the value, type the value and then press Enter.
- If you want to go back to the previous question, clear the default entry, or quit the installation, special commands are available. Entries beginning with the "!" character are considered as special commands:
  - type !b and hit Enter to get the previous page.
  - type !c and hit Enter to clear the default value.
  - type !q and hit Enter to quit.
- If you want to have an entry beginning with a "!" which is not a command, you must precede it by an additional "!". For example, if the question is to enter a password, and if your password is !xoxo, you must write !!xoxo. And if you want to enter !!xoxo, you must write !!!xoxo (and so on).
- Check boxes are used to facilitate selection of multiple options. Check boxes are identified by [ ]:

  1 [ ]: unselected check box
  2 [*]: selected check box

To check or uncheck a check box, enter its corresponding number and press Enter. Press -1 to select all the check boxes and 0 to unselect all.
- For radio buttons, only one choice is possible. Radio buttons are identified by { }:

  1 ( ): unselected radio button
  2 (*): selected radio button

To select a radio button, enter its corresponding number and press Enter (This will automatically unselect the previous choice).

Note: You should not enable the launch of TomEE Configuration Utility during your TUI mode installation; as this will launch GUI based utility and the installation won’t be completely text based.

1. Open a Command Prompt window on Windows or a shell window on UNIX. The command must be launched from an elevated command prompt from Windows 7 onwards.
2. Insert the Isight installation DVD, or extract the archive file of the installation media and go to the directory:

   On Windows :
   
   `<dvd_media_dir>\SIMULIA_Isight\Windows64\1`

   On Linux :
   
   `<dvd_media_dir>/SIMULIA_Isight/Linux64/1`

3. Launch the TUI mode installation by running the appropriate command for your operating system.

   On Windows :
   
   `StartTUI.exe`
On Linux:

`.StartTUI.sh`

4. Press `Enter` to continue.

5. Provide the path to the destination directory into which the software will be installed and press `Enter` to continue.

You can directly press `Enter` if you want to install the software on the mentioned default path.

6. Press `0` and `Enter` then press `2` and `Enter` to select SIMULIA Execution Engine as the product to install:

   1 [ ] Isight Desktop
   2 [*] SIMULIA Execution Engine
   3 [ ] SIMULIA Execution Engine Station

The state of check box preceding a product name gives the selection state of the product that will be installed. You can choose any combination of these components to install on different computers and servers.

Press `Enter` to confirm the selection and proceed to next step.

7. If you want to create a connection profile (.cpr file) to connect Isight Clients to a SIMULIA Execution Engine server during installation time, select the radio button preceding the appropriate application server and press `Enter`:

Press `Enter` to enter the server host name.

Press `Enter` to enter the port number and press `Enter`.

If a Isight Client application will use a SIMULIA Execution Engine, it must have a connection profile specific to the application server getting used. The .cpr file must be located in the following directory:

   `<isight_install_dir>/config/`

An initial configuration profile named `standalone.cpr` is provided for standalone Isight operation. Additional connection profiles can be created later by using the `Edit Logon Profile` tool. See `Creating a SIMULIA Execution Engine Connection Profile` for more details.

8. Select the type of license server software you will use and press `Enter`, or skip the licensing selection for now:

   1 (*) Dassault Systemes License Server
   2 ( ) FlexNet License Server
   3 ( ) Skip this for now

If you (or your system administrator) already have the FlexNet or Dassault Systèmes license server installed and running, press `Enter` and specify the server computer’s host name and press `Enter` to specify port number. This information is used to contact the license server and create a licensing client file that references the server. If you have installed a redundant license server triad, enter the host name and port for all three machines or you may continue pressing `Enter` keeping blank values for the remaining license server host name and port fields.

9. Verify the information on Summary text that corresponds to the entries made earlier (type `!b` and press `Enter` to go back and do some modifications if necessary).

10. Press `Enter` to allow the installer to complete installation.
Installing SIMULIA Execution Engine in Silent Mode

In Silent Mode Installation of Isight, you provide a response file recorded during a previous installation to the installation program. With proper inputs, Isight 2016 installer, once launched, will install Isight products without any user interaction.

In Silent Mode, Isight 2016 installer does not need a graphics card to be present in your machine.

Note: You should not enable the launch of TomEE Configuration Utility during your previous installation as this will invalidate the response file being recorded for using in Silent Mode.

1. Retrieve the response file from the previous installation of Isight Desktop.

   The response file is generated automatically by a previous Isight installation of same version. It contains the responses to the questions you were asked during the previous installation. The file name and path is:
   
   `<Install_Path>/InstallData/UserIntentions.xml`

   For example, after installing Isight on Windows with default path, the path would be:
   
   `C:\SIMULIA\Isight\Isight2016\InstallData\UserIntentions.xml`

   After retrieving the response file, you can put it in any location you like.

   A response file is specific to:
   
   • a particular operating system: response files are different between Windows and UNIX
   • a particular version of Isight

2. Edit the response file.

   Open the response file in any text editor of your choice for editing if required. The variables you can edit are as follows:
   
   • path values
   • text values
   • port number values

   Boolean variables must never be changed from TRUE to FALSE or vice versa, because this changes the sequence and the list of the installation questions and the answers may not be found in the response file. The response file must be:
   
   • XML "well-formed"
   • UTF8 encoded

3. Launch the Isight silent installation.

   You launch the Isight Silent Mode installation in the same manner as in console mode using the StartTUI program, except that you must specify the `--silent` option followed by the full path of the response file.

   For doing this, in the installation media, go to the directory:

   On Windows:
   `<dvd_media_dir>\SIMULIA_Isight\Windows64\1`

   On Linux:
   `<dvd_media_dir>/SIMULIA_Isight/Linux64/1`
Run the following command on Windows:

```
StartTUI.exe --silent C:\UserIntentions.xml
```

and on UNIX:

```
./StartTUI.sh --silent /UserIntentions.xml
```

The installation will run in silent mode with some progress messages printed to the terminal. The messages will include the location of any log files, which would be helpful in case something goes wrong.

---

### Uninstalling the SIMULIA Execution Engine

You can permanently remove the SIMULIA Execution Engine and the license service at any time, as conditions warrant. This process involves stopping any running SIMULIA Execution Engine interfaces and the license server, removing the SIMULIA Execution Engine software, and deleting any temporary directories and files that are left behind.

If you have multiple releases of the SIMULIA Execution Engine installed on a single computer using local license files, uninstalling one release of the SIMULIA Execution Engine may delete the license server used by the other installations.

The procedure for uninstalling the SIMULIA Execution Engine consists of the following steps:

1. Verifying that no jobs are running and that all attached interfaces are closed.
2. If you are running a license server on the same computer as the SIMULIA Execution Engine, stop the license server program.
3. Removing the SIMULIA Execution Engine software.
4. Removing the temporary directories and files created by the SIMULIA Execution Engine.

---

### Closing the SIMULIA Execution Engine Client Applications

Before you can uninstall the SIMULIA Execution Engine, you need to verify that no jobs are running and that all attached interfaces (from both Isight and the SIMULIA Execution Engine) are closed.

1. Login as the user that installed the SIMULIA Execution Engine. This user should be an Administrator or have administrative privileges on Windows.
2. Close all Isight Design Gateways, Isight Runtime Gateways, and SIMULIA Execution Engine WebTops attached to the SIMULIA Execution Engine, which should stop all running jobs.
3. Close all SIMULIA Execution Engine Stations, including any stations that are running as a service.

**Important:** If you installed a SIMULIA Execution Engine Station as a service, you must uninstall it manually before removing the SIMULIA Execution Engine.

4. Open the SIMULIA Execution Engine Dashboard and log in to the SIMULIA Execution Engine.
5. Verify that all SIMULIA Execution Engine Stations are closed and that no jobs are running.
   
   If necessary, you can use the Dashboard to close SIMULIA Execution Engine Stations and stop running jobs. For more information on using the SIMULIA Execution Engine Dashboard, see *Using the Dashboard*.
6. Close the SIMULIA Execution Engine Dashboard.
Removing the SIMULIA Execution Engine Software

You remove the SIMULIA Execution Engine by using the Windows control panel or by running a script.

1. If necessary, stop and remove the license server software as described in the Isight Installation Guide.
2. Uninstall/remove Dassault Systemes Simulia Isight 5.x using the Windows control panel Uninstall a program.
3. Close the control panel.
   
   **Note:** If a message appears informing you that Isight has been removed and that you must restart your system to complete the uninstallation process, be sure to perform the restart before continuing to the next section.

4. Remove any temporary files or directories created by the SIMULIA Execution Engine station. See Deleting Station Temporary Directories and Files on Windows.
5. Optional: Instead of the Windows control panel, you can use the following batch file to uninstall the SIMULIA Execution Engine:

   `<see_install_dir>\Uninstall.bat`

6. On Linux, navigate to the `<see_install_dir>` directory and delete all of the installed files with the following command:

   `./rm -rf *`

Deleting Temporary Directories and Files on Windows

Once you have removed the SIMULIA Execution Engine software, you need to remove the temporary directories and files created by the SIMULIA Execution Engine.

1. Navigate to the directory that contains the top level of your SIMULIA Execution Engine installation directory. For example, if you installed SIMULIA Execution Engine in `C:\SIMULIA\Execution Engine\2016`, navigate to the `C:\SIMULIA\Execution Engine` directory.
2. Delete the `2016` directory and all of its contents.
3. Navigate to the following directory, where `<user_name>` is the name of the user who installed and uninstalled the SIMULIA Execution Engine:

   `C:\Users\<user_name>\`

4. Delete the following items:
   - the `fiper` directory
   - `fiper.preferences` file
5. Navigate to the following directory, where `<user_name>` is the name of the user who installed and uninstalled the SIMULIA Execution Engine:

   `C:\Users\<user_name>\AppData\Local\Temp\`

6. Delete the `fiper` directory.
7. Navigate to the location of the SIMULIA Execution Engine file manager directory. This directory location is specified by the property `fiper.system.filemgr.rootFilePath` in the `acs.properties` file.
8. Delete the entire SIMULIA Execution Engine file manager directory.
9. Navigate to the SIMULIA Execution Engine temporary directory. This directory location is specified by the property fiper.system.temp in the acs.properties file.

10. Delete the following items:
   - the fiper*.mmjarcache directory (there may be more than one directory that matches this format)
   - the fiper directory (if present)

11. If you ran a SIMULIA Execution Engine Station on the SIMULIA Execution Engine system, navigate to the location of the station directory.

12. Delete the entire SIMULIA Execution Engine Station directory (typically the same as the computer name running that station).
   
   Be sure to uninstall any SIMULIA Execution Engine Stations on different systems that were using the deleted SIMULIA Execution Engine and that you do not plan to use with a different SIMULIA Execution Engine.
   
   The SIMULIA Execution Engine removal process is complete.

**Deleting Temporary Directories and Files on Linux**

Once you have removed the SIMULIA Execution Engine software, you need to remove the temporary directories and files created by the SIMULIA Execution Engine.

1. Log in as root (obtain root permissions). Although it is not necessary to be root to delete all the temporary files, it is necessary for some of the files.

2. Navigate to the directory that contains the top level of your SIMULIA Execution Engine installation. For example, if you installed the SIMULIA Execution Engine in /opt/SIMULIA/Execution Engine/2016, navigate to the /opt/SIMULIA/Execution Engine directory.

3. Delete the 2016 directory and all of its contents.

4. Navigate to the location of the SIMULIA Execution Engine file manager directory. This directory location is specified by the property fiper.system.filemgr.rootFilePath in the acs.properties file.

5. Delete the entire SIMULIA Execution Engine file manager directory.

6. Navigate to the SIMULIA Execution Engine temporary directory. This directory location is specified by the property fiper.system.temp in the acs.properties file. By default, this directory is /tmp.

7. Delete any directories that use the following naming convention:
   
   fiper*.mmjarcache

8. Navigate to the $HOME directory for the user that installed the SIMULIA Execution Engine.

9. Delete the following items (if they are present):
   - .fiper.preferences file (notice the leading “.” in the file name)
   - dashboard.log file
   - Fiperinstall.log file
   - Fiperuninstall.log file
   - gateway.log file

10. Navigate to the following directory:
    
    /var/tmp

11. Delete the flexlm.log file (if it is present).
12. If you ran a SIMULIA Execution Engine station on the SIMULIA Execution Engine system, navigate to the location of the station directory. This directory location was specified during the SIMULIA Execution Engine installation.

13. Delete the entire SIMULIA Execution Engine station directory (typically the same as the computer name running that SIMULIA Execution Engine station).
   Be sure to uninstall any SIMULIA Execution Engine stations on different systems that were using the deleted SIMULIA Execution Engine and that you do not plan to use with a different SIMULIA Execution Engine.
   The SIMULIA Execution Engine removal process is complete.
Installing a SIMULIA Execution Engine Station on Windows

Several steps are required to successfully install the SIMULIA Execution Engine Station software.

1. Start the SIMULIA Execution Engine Installation Wizard on the system that will be running the SIMULIA Execution Engine. This wizard is used to install both the SIMULIA Execution Engine and the station software.

2. Manually install the station as a service. If you install the station as an application and later want to run it as a service, you can make this switch manually.

Installing the Station Software on Windows in Graphical User Interface (GUI) Mode on Windows

To install the SIMULIA Execution Engine Station on Windows, you run the installation wizard.

In the GUI mode installation of Station, the Isight 2016 installer prompts you to answer questions in dialog boxes.

1. Insert the Isight installation DVD, or extract the archive file of the installation media and go to the directory:
   On Windows :
   `<dvd_media_dir>\SIMULIA_Isight\Windows64\1`

2. Launch the GUI mode installation by running `setup.exe`

3. If necessary, click **Allow** on the **User Account Control** dialog to confirm the execution of the installation program.

4. Click **Next** to advance through each panel of the installer, responding to the prompts as needed.
   - In the first panel, choose the destination directory into which the software will be installed.
   - In the second panel, select SIMULIA Execution Engine Station as the product you want to install:
     - **Isight Desktop**
     - **SIMULIA Execution Engine**
     - **SIMULIA Execution Engine Station**
     
     You can choose any combination of these components to install on different computers and servers. For more information about the SIMULIA Execution Engine, see the *SIMULIA Execution Engine Installation and Configuration Guide*.

5. If you want to create a connection profile (.cpr file) to connect SIMULIA Execution Engine Station to a SIMULIA Execution Engine server during installation, select the appropriate application server and click **Next** to enter the server host name and port number.
   - **Server Host Name**. The host name of the computer running the SIMULIA Execution Engine server application.

   **Note**: If this SIMULIA Execution Engine will be accessed from computers in multiple network domains (for example, domain1.xxx.com and domain2.xxx.com), you must specify the fully qualified host name (for example, host.domain1.xxx.com).
• **Server Port Number**

The Station application will connect to SIMULIA Execution Engine, so it must have a connection profile specific to the application server getting used. The .cpr file must be located in the following directory:

```
<isight_install_dir>/config/
```

Additional connection profiles can be created later by using the **Edit Logon Profile** tool. See *Creating a SIMULIA Execution Engine Connection Profile* for more details.

6. Select the type of license server software you will use, or skip the licensing selection for now:

   - Dassault Systemes License Server
   - FLEXnet License Server
   - Skip this for now

   If you (or your system administrator) already have the FLEXnet or Dassault Systèmes license server installed and running, specify the server computer's host name and port number. This information is used to contact the license server and create a licensing client file that references the server. If you have installed a redundant license server triad, enter the host name and port for all three machines.

7. Verify the information on Summary panel that corresponds to the entries made earlier (and do some modifications if necessary).

8. Click **Finish** to allow the installer to complete installation of SIMULIA Execution Engine Station.

9. After the installer completes, you can edit the values in the `station.properties` file to customize the behavior of the station. For more information, see *Configuring SIMULIA Execution Engine Station Properties*.

   The `station.properties` file is installed into the `\config\` subdirectory:

   ```
   <station_install_dir>\config\n   ```

   In particular, you should decide whether you want to change the following properties:

   - **Station Affinities.** Any affinity setting in addition to the default of station name and platform. For more information on affinities, see *About Station Affinities*.
   - **Default log level.** The default setting is *Info*. For more information on these settings, see *About Log Message Detail Levels*.
   - **Temp directory.** The default is the current user’s temporary directory. If you want to change this setting, be sure to select a directory that has the following characteristics:
     - Is not a temporary file system. This disk space must never be reclaimed automatically. This rules out any directories that are cleared during a reboot or during an automatic disk space cleanup.
     - Preferably on a local disk on the station host system. If there is insufficient local storage space, a NAS device can be used; however, this setup is not recommended.

   ⚠️ **Important:** You must change this setting if you plan on using the SIMULIA Execution Engine station security (Run-As) feature. You must use a directory that can be accessed by all users (for example, `c:\temp`). For more information on Run-As specifications, including how to change the station temporary directory after an installation, see *About File System Security With Run-As*. For more information on determining or changing directory permissions, contact your local system administrator.
Installing a SIMULIA Execution Engine Station as a Service

You can set up the SIMULIA Execution Engine station to run as a service. Only one station can be run as a service on any given computer.

Installing a SIMULIA Execution Engine Station as a Service

Manually installing a SIMULIA Execution Engine as a service involves running a command, which is included with your station installation, and specifying a SIMULIA Execution Engine connection when prompted.

1. Log in as an Administrator or a user with administrative privileges.
2. Open a Command Prompt dialog box.
3. Type the following command:
   For Windows 64-bit:
   `<isight_install_dir>\win_b64\code\command\installstation.bat`
   A short message appears in the console, and the SIMULIA Execution Engine Logon dialog box appears.
4. Select the connection profile for the SIMULIA Execution Engine you want this SIMULIA Execution Engine station to use, and enter the logon ID and password to be used.
   You will not log in to the SIMULIA Execution Engine at this time; the information is stored for later use when the service is started.
5. Click OK.
   A SIMULIA Execution Engine station has now been configured and installed as a service. Now you need to access the Windows Services interface and start the station’s service.

Starting a SIMULIA Execution Engine Station as a Service

Once you install the station’s service, you need to start the service from your computer’s Services dialog box. The process for accessing this dialog box varies based on the operating system you are using.

1. Access the Services dialog box.
   • If your station is running on Windows Server 2008:
     a) Click the Start button, point to Administrative Tools and click Services.
     b) Click Continue.
   • If your station is running on Windows 7:
     a) Click the Start button, and click Control Panel.
     b) Click System and Security, and click Administrative Tools.
     c) Double-click Services.

2. From the Services dialog box, locate the service named Fiper Station, and click it to select it.
3. Click Start Service on the Services dialog box toolbar, and wait for the service to start.
   If there are any problems, a log file (station.log) can be reviewed. This file is located in the following directory:
   `<station_temporary_directory>\<hostname>\`
The SIMULIA Execution Engine station should now be running, and it will appear in the stations list of the SIMULIA Execution Engine (you can see it using the Dashboard or the Command Line Client stationstatus command).

### Stopping a SIMULIA Execution Engine Station Service

If at any point you need to shut down a station running as a service (including prior to uninstalling the station), you need to do so from the Services dialog box.

1. Access the Services dialog box as described in *Starting a SIMULIA Execution Engine Station as a Service*.
2. Locate the service named Fiper Station, and click it to select it.
3. Click Stop Service on the Services dialog box toolbar, and wait for the service to stop (this process will take a bit longer than starting the service because the SIMULIA Execution Engine station must first wait for running work items to finish.

### Uninstalling a SIMULIA Execution Engine Station Service

Uninstalling a SIMULIA Execution Engine that is running as a service involves running a command, which is included with your station installation.

1. Verify that the station service is stopped as described in *Stopping a SIMULIA Execution Engine Station Service*.
2. Open a Command Prompt dialog box.
3. Type the following command:
   For Windows 64-bit:
   ```
   <isight_install_dir>\win_b64\code\command\uninstallstation.bat
   ```
   The SIMULIA Execution Engine station has now been removed as a service.
4. If desired, confirm the removal of the station service by refreshing the Services dialog box and verifying that the service called Fiper Station is no longer listed.

### Uninstalling a SIMULIA Execution Engine Station

You can permanently remove the SIMULIA Execution Engine station at any time, as conditions warrant. This process involves stopping the station and the license server, removing the SIMULIA Execution Engine station software, and deleting any temporary directories and files that are left behind.

The procedure for uninstalling the SIMULIA Execution Engine station consists of the following steps:

1. If you are running a license server on the same computer as the SIMULIA Execution Engine station, stop the license server program.
2. Removing the SIMULIA Execution Engine station software.
3. Removing the temporary directories and files created by the SIMULIA Execution Engine station.

### Stopping the Station

Before you can remove the station software, you need to verify that the station is no longer running.

1. Login as Administrator or a user with administrative privileges.
2. Verify that the SIMULIA Execution Engine station you are removing is stopped (if the station was installed as a service, verify that the associated service has been stopped).
3. If you are running the SIMULIA Execution Engine license server on the same computer as the SIMULIA Execution Engine station, stop the license server before uninstalling the station.
4. You now need to remove the SIMULIA Execution Engine software as described in *Removing the SIMULIA Execution Engine Station Software*.

### Removing the SIMULIA Execution Engine Station Software

The process of removing the SIMULIA Execution Engine station software differs based on the operating system of the computer running the station. In general, you will launch the uninstallation wizard to remove the station software.

1. Uninstall/remove **Dassault Systemes Simulia Isight 5.x** using the Windows control panel **Uninstall a program**.
2. Close the control panel.

   🔄 **Note:** If a message appears informing you that Isight has been removed and that you must restart your system to complete the uninstallation process, be sure to perform the restart before continuing to the next section.

3. Remove any temporary files or directories created by the SIMULIA Execution Engine station. See *Deleting Station Temporary Directories and Files on Windows*.

4. **Optional:** Instead of the Windows control panel, you can use the following batch file to uninstall the SIMULIA Execution Engine:

   ```
   <station_install_dir>\Uninstall.bat
   ```

5. You now need to remove any temporary directories or files that the SIMULIA Execution Engine station created as described in *Deleting Station Temporary Directories and Files on Windows*.

### Deleting Station Temporary Directories and Files on Windows

Once you have removed the SIMULIA Execution Engine station software, you need to delete the temporary directories and files created by the SIMULIA Execution Engine station.

1. Navigate to the directory that contains the top level of your SIMULIA Execution Engine installation directory. For example, if you installed the SIMULIA Execution Engine in `C:\SIMULIA\Execution Engine\2016`, navigate to the `C:\SIMULIA\Execution Engine` directory.
2. Delete the **2016** directory and all of its contents.
3. Navigate to the following directory, where `<user_name>` is the name of the user who installed and uninstalled the SIMULIA Execution Engine station:

   - Windows Server 2008/7: `C:\Users\<username>`

4. Delete the following items:

   - the `fiper.preferences` file
   - the `fiper` directory

5. Navigate to the following directory, where `<user_name>` is the name of the user who installed and uninstalled the SIMULIA Execution Engine station:
6. Delete the fiper directory.
7. Navigate to the location of the station temporary directory.
8. Delete the entire SIMULIA Execution Engine station directory (typically the same as the computer name running that station).

The SIMULIA Execution Engine station removal process is complete.
Installing a SIMULIA Execution Engine Station on Linux

You can install a SIMULIA Execution Engine station on Linux platforms.

Installing the Station Software on Linux in Graphical User Interface (GUI) Mode

To install SIMULIA Execution Engine Station on Linux, you run the Isight installation wizard.

In the GUI mode, the Isight 2016 installer prompts you to answer questions in dialog boxes. On Linux, the dialog boxes are based on Motif (or OpenMotif) and require a graphics card on the Linux machine.

Installing Isight in GUI mode on Linux, also requires an X-Windows display, either local or remote over a network. If the DISPLAY environment variable is not set, the installer will not execute.

1. Log in as the user that will install the software.
   - If you are not installing as root, it is recommended that you review the information in Installing as an Administrator (Windows) or Non-Root User (Linux) before beginning your installation.
   - If you log in as a normal, non-root user and enable root/superuser privileges with the su command, you must use the “su -” command (su space dash) to read the root profile. Otherwise, the installer may fail because required administrator utilities will not be in the executable path (PATH).

2. Insert the Isight installation DVD, or extract the archive file of the installation media and go to the directory:
   - On Linux:
     `<dvd_media_dir>/SIMULIA_Isight/Linux64/1`

3. Launch the GUI mode installation by running the command:
   - On Linux:
     `./startGUI.sh`

4. Click Next to advance through each panel of the installer, responding to the prompts as needed.
   - In the first panel, choose the destination directory into which the software will be installed.
   - In the second panel, select SIMULIA Execution Engine Station as the product you want to install:

     Isight Desktop
     SIMULIA Execution Engine
     SIMULIA Execution Engine Station

     You can choose any combination of these components to install on different computers and servers. For more information about the SIMULIA Execution Engine, see the SIMULIA Execution Engine Installation and Configuration Guide.

5. If you want to create a connection profile (.cpr file) to connect SIMULIA Execution Engine Station to a SIMULIA Execution Engine server during installation, select the appropriate application server and click Next to enter the server host name and port number.
• **Server Host Name.** The host name of the computer running the SIMULIA Execution Engine server application.

   **Note:** If this SIMULIA Execution Engine will be accessed from computers in multiple network domains (for example, domain1.xxx.com and domain2.xxx.com), you must specify the fully qualified host name (for example, host.domain1.xxx.com).

• **Server Port Number**

The Station application will connect to SIMULIA Execution Engine, so it must have a connection profile specific to the application server getting used. The .cpr file must be located in the following directory:

`<isight_install_dir>/config/`

Additional connection profiles can be created later by using the **Edit Logon Profile** tool. See *Creating a SIMULIA Execution Engine Connection Profile* for more details.

6. Select the type of license server software you will use, or skip the licensing selection for now:

   Dassault Systemes License Server
   FLEXnet License Server
   Skip this for now

If you (or your system administrator) already have the FLEXnet or Dassault Systèmes license server installed and running, specify the server computer's host name and port number. This information is used to contact the license server and create a licensing client file that references the server. If you have installed a redundant license server triad, enter the host name and port for all three machines.

7. Verify the information on Summary panel that corresponds to the entries made earlier (and do some modifications if necessary).

8. Click **Finish** to allow the installer to complete installation of SIMULIA Execution Engine Station.

9. After the installation completes, you can edit the values in the `station.properties` file to customize the behavior of the station. For more information, see *Configuring SIMULIA Execution Engine Station Properties*.

   The `station.properties` file is installed into the `/config/` subdirectory:

   `<station_install_dir>/config/`

   In particular, you should decide whether you want to change the following properties:

   • **Station Affinities.** Any affinity setting in addition to the default of station name and platform. For more information on affinities, see *About Station Affinities*.

   • **Default log level.** The default setting is `Info`. For more information on these settings, see *About Log Message Detail Levels*.

   • **Temp directory.** The default is the current user’s temporary directory. If you want to change this setting, be sure to select a directory that has the following characteristics:

      • Is not a temporary file system. This disk space must never be reclaimed automatically. This rules out any directories that are cleared during a reboot or during an automatic disk space cleanup.

      • Preferably on a local disk on the station host system. If there is insufficient local storage space, a NAS device can be used; however, this setup is not recommended.

   **Important:** You must change this setting if you plan on using the SIMULIA Execution Engine station security (Run-As) feature. You must use a directory that can be accessed by all users. For more information on Run-As specifications, including how to change the station temporary directory after an installation, see *About File System Security With Run-As*. For more information on determining or changing directory permissions, contact your local system administrator.
Enabling the SIMULIA Execution Engine Station Security Feature (Run-As)

If the SIMULIA Execution Engine station software is installed by a non-root user, you must issue two commands as root to allow the station security feature to work properly.

If the SIMULIA Execution Engine station software is installed by a non-root user on Linux, follow the steps below to change permissions on the SMAFIPplaunch file.

1. Log on to the system running the SIMULIA Execution Engine as root.
2. Change directory (cd) to the following directory:
   
   ```
   <see_install_dir>/<os_dir>/code/bin/
   ```

   where the `<os_dir>` subdirectory is the operating system on which the station is running:

   ```
   linux_a64 for 64-bit Linux
   ```

3. Give the following commands:

   ```
   chown root SMAFIPplaunch
   chmod 4711 SMAFIPplaunch
   ```

   For other configuration steps required, see Configuring SIMULIA Execution Engine Stations for Run-As on Linux.

Installing a SIMULIA Execution Engine Station as a Service Manually

The SIMULIA Execution Engine station can be set up to run as a service (daemon process) on Linux.

To install the SIMULIA Execution Engine station as a service manually, run the following command (as a root user):

```
<SEE_install_dir>/<os_dir>/code/command/station.service install
```

For example:

```
/opt/SIMULIA/ExecutionEngine/5.x/linux_a64/code/command/station.service install
```

The normal SIMULIA Execution Engine logon dialog will display, allowing you to select the SIMULIA Execution Engine to which you want to connect and the SIMULIA Execution Engine logon and password. A startup file is installed in `/etc/rc3.d/S95station`.

The following information should also be noted when installing a station as a service using this command:

- To uninstall the station, run the following command (as a root user):
  ```
  <SEE_install_dir>/<os_dir>/code/command/station.service uninstall
  ```

- The initial installation of the SIMULIA Execution Engine station as a service does not start the station. You must start it manually immediately following the installation using the following command (after this manual station start, the station will start automatically after subsequent system reboots):
  ```
  <SEE_install_dir>/<os_dir>/code/command/station.service start
  ```

- You can stop a station as a service with the following command:
  ```
  <SEE_install_dir>/<os_dir>/code/command/station.service stop
  ```
• If you want the station to run as a non-root user, edit the `station.service.template` file before installing the station as a service and change the following line:

```
STATION_USER=
```

to have the name of the user you want the station to run. This does not have to be the same user name used to log on to the SIMULIA Execution Engine.

### Uninstalling a SIMULIA Execution Engine Station

You can permanently remove the SIMULIA Execution Engine station. This process involves stopping any running SIMULIA Execution Engine interfaces and the license server, removing the SIMULIA Execution Engine software, and deleting any temporary directories and files that are left behind.

The procedure for uninstalling the SIMULIA Execution Engine station consists of the following steps:

1. If you are running a license server on the same computer as the SIMULIA Execution Engine station, stop the license server process.
2. Removing the SIMULIA Execution Engine station software.
3. Removing the temporary directories and files created by the SIMULIA Execution Engine station.

### Stopping the SIMULIA Execution Engine Station

Before you can remove the station software, you need to verify that the station is no longer running.

1. Verify that you are logged in as the same user that installed the SIMULIA Execution Engine station.
2. Stop any interactive stations.

   **Note:** If you installed a SIMULIA Execution Engine station as a service, you must stop and uninstall it manually before removing the SIMULIA Execution Engine. For more information, see *Installing a SIMULIA Execution Engine Station as a Service Manually*.

3. Log in as root (obtain root permissions).
4. If the license server is running on the same computer as the SIMULIA Execution Engine station, you need to remove the license server.
5. You now need to remove the SIMULIA Execution Engine station software as described in *Removing the SIMULIA Execution Engine Station Software*.

### Removing the SIMULIA Execution Engine Station Software

You can now remove the SIMULIA Execution Engine station software.

1. Navigate to the following directory:

   `<station_install_dir>/`

2. Execute the following command:

   ```bash
   ./*
   ```

   **Important:** You must be logged in as the same user that installed the SIMULIA Execution Engine station or the uninstaller will not be able to completely remove the software.

3. Log out of your system, and log back into your system.
4. You now need to remove any temporary directories or files that the SIMULIA Execution Engine created as described in *Deleting Temporary Files and Other Files*.

**Deleting Temporary Files and Other Files**

Once you have removed the SIMULIA Execution Engine station software, you need to remove the temporary directories and files created by the SIMULIA Execution Engine station.

1. Log in as root (obtain root permissions). Although it is not necessary to be root to delete all the temporary files, it is necessary for some of the files.

2. Navigate to the directory that contains the top level of your SIMULIA Execution Engine installation. For example, if you installed SIMULIA Execution Engine in `/opt/SIMULIA/Execution Engine/2016`, navigate to the `/opt/SIMULIA/Execution Engine` directory.

3. Delete the 2016 directory and all of its contents.

4. Navigate to the `$HOME` directory for the user that installed the SIMULIA Execution Engine.

5. Delete the following items (if they are present):
   - `.fiper.preferences` file (notice the leading “.” in the file name)
   - `dashboard.log` file
   - `Piperinstall.log` file
   - `Piperuninstall.log` file
   - `gateway.log` file

6. Navigate to the following directory:
   `/var/tmp`

7. Delete the `flexlm.log` file (if it is present).

8. Navigate to the location of the station temporary directory.

9. Delete the entire SIMULIA Execution Engine station directory (typically the same as the computer name running that SIMULIA Execution Engine station).

   The SIMULIA Execution Engine station removal process is complete.
Configuring an Oracle Database

If you will use your own Oracle database with the SIMULIA Execution Engine application, you must create the Oracle instance and configure the database user account, tablespace, and tables. This must be done before installing the SIMULIA Execution Engine server.

These instructions apply to Oracle 11gR2.

If you will use the built-in Derby database instead of Oracle, skip the instructions in this section.

Creating the Database in Oracle

You use the Oracle Database Configuration Assistant to create the database for your SIMULIA Execution Engine. The steps provided here are for Oracle 11gR2 databases. Other database products will have different administrative tools to accomplish these tasks.

1. Access the Database Configuration Assistant using one of the following methods:
   - Windows: Click the Start button, point to All Programs/Oracle - OraDb11g_home1/Configuration and Migration Tools, and click Database Configuration Assistant.
   - Linux: Navigate to the <oracle_install_directory>/bin directory, and execute the dbca file.

   The Welcome message appears.

2. Click Next.

   The Operations screen appears.

3. Verify that Create a Database is selected.

4. Click Next.

   The Database Templates screen appears.

5. Verify that General Purpose or Transaction Processing is selected.

6. Click Next.

   The Database Identification screen appears.

7. Enter the database name in the Global Database Name text box (for example, SEE).

8. Verify that the database name appears in the SID text box.

9. Click Next.

   The Management Options screen appears.

10. Verify that Configure Enterprise Manager is selected.

11. Verify that Configure Database Control for local management is selected.

12. Click Next.

   The Database Credential screen appears.

13. Verify that Use the Same Administrative Password for All Accounts is selected.

14. Enter the password (for example, seeadmin) in the Password and Confirm Password text boxes.
15. Click Next.
   The Storage Options screen appears.

16. Verify that File System is selected.
17. Click Next.
   The Database File Locations screen appears.

18. Verify that Use Database File Locations from Template is selected.
19. Click Next.
   The Recovery Configuration screen appears.

20. Verify that Specify Flash Recovery Area is selected.
21. Click Next.
   The Database Content screen appears.

22. Click Next.
   The Initialization Parameters screen appears.

23. Verify that the Memory tab is selected.
24. Do the following:
   a. Verify that the Typical button is selected.
   b. Set the Memory Size to the maximum amount of memory to be used by Oracle. The amount specified depends on the database system’s hardware capabilities and what other applications will be run on the system. A computer dedicated as a database server should allocate all available physical memory, less the operating system requirements, to Oracle. For example, a dedicated Windows server with 4 GB of memory should allocate about 3 GB to Oracle, leaving 1 GB for the operating system.
   c. Verify that Use Automatic Memory Management is selected.

25. Click the Sizing tab.
26. Type 300 in the Processes text box.
27. Click the Character Sets tab.
28. Click Use Unicode (AL32UTF8).
29. In the National Character Set list, select the following:
   a. Windows: UTF8 - Unicode 3.0 UTF-8 Universal character set
   b. Linux: UTF

30. Click the Connection Mode tab.
31. Verify that Dedicated Server Mode is selected.
32. Do the following:
   a. Click Next.
      The Security Settings screen appears.
   b. Click Next.
      The Automatic Maintenance Tasks screen appears.

33. Click Next.
   The Database Storage screen appears.
34. Click Next.
   The Create Options screen appears.

35. Verify that Create Database is selected.
36. Click Finish.
   A Confirmation message appears.

37. Click OK.
   A message appears when the database is created.

38. Make a note of the URL given as the Database Control URL. It is the address you will use to access the
   Enterprise Manager to create tables and a user account (for example, https://hostname.yourcompany.com:1158/em).
39. Click Exit.

---

### About Oracle URLs and Port Numbers

To access the Oracle Enterprise Manager web-based console, you must specify the URL and port number for
the correct database.

The URL address for connecting to the database uses the following general format:

```
https://hostname.yourcompany.com:portnumber/em
```

In Oracle 11gR2, use https instead of http in this URL.

To determine the port number for your Oracle database, navigate to the `<oracle_install_directory>/
cfgtoollogs/dbca/<database_name>` directory, and open the `emConfig.log` file. To obtain the full
URL for the database, including the port number, enter the following as the search criteria in your text editor:

```
The Database Control URL is
```

---

### Creating Tablespaces and Defining User Information

Using the Oracle Enterprise Manager web-based console, you need to create a tablespace for the SIMULIA
Execution Engine data and define the user that will access the database.

1. Log on to the computer that contains the Oracle software.
   Although you can access the database from any computer on your network using a web browser, this procedure
   is written assuming that you are directly accessing the system running Oracle.

2. Verify that you have created the SIMULIA Execution Engine-specific database using the procedure described
   in Creating the Database in Oracle.
   You must create a database following these steps to ensure that the database contains the correct internal
   settings.

3. Access the database with the Oracle Enterprise Manager, using one of the following methods, based on your
   operating system:
   - Windows: Click Start, point to All Programs, Oracle - OraDb11g_home1, and click Database Control - SEE. (In this example, SEE is the name of the database you are accessing.)
• Linux: Use a browser to open the following page:

https://hostname.yourcompany.com:portnumber/em

For more information about determining the port number for your database, see About Oracle URLs and Port Numbers.

4. Type SYS as the user name in the corresponding text box, and enter the password you specified when you created the database.

5. From the Connect As list, select SYSDBA.

6. Click Login.

7. Verify that you have accessed the correct database.

   The screen that appears after you log in shows the database name next to the Database Instance string (in the upper left corner of the screen).

8. At the top of the console, access the Server information (Oracle 11gR2), and click Server.

9. In the Storage list, click Tablespaces.

   The Tablespaces page opens.

10. On the right side of the console, click Create.

    The Create Tablespace page appears.

11. In the Name text box, type the following:

    FIPERTS1

12. Click Set as default permanent tablespace.

13. In the Datafiles area, click Add.

    The Add Datafile page appears.

14. In the File Name text box, type the following entry:

    FIPERDF1.DBF

15. In the File Size text box, type 2048, and verify that MB is selected from the corresponding list.

16. Click Automatically extend datafile when full (AUTOEXTEND).

17. In the Increment text box, type 100, and select MB from the corresponding list.

18. In the Maximum File Size area, verify that Unlimited is selected.

19. Click Continue.

    The Create Tablespace page appears.

20. Click OK.

    A message appears indicating that the object was created successfully.

21. At the top of the left side of the console, click Database Instance.

    The Server page appears.

22. In the Security list, click Users.

    The Users page appears.

23. Click Create.
The Create User page appears.

24. In the Name text box, type seeadmin.
25. From the Profile list, verify that DEFAULT is selected.
26. In both the Enter Password and Confirm Password text boxes, type the password for the seeadmin user (you can use seeadmin for simplicity, if desired).
27. Click the icon adjacent to the Default Tablespace text box.

The Search and Select window appears.
28. Verify that FIPERTS1 is selected.
29. Click Select.

You are returned to the Create User page.
30. Click Roles.
31. On the right side of the console, click Edit List.

The Modify Roles screen appears.
32. From the Available Roles list, select Resource.
33. Click Move to move the role to the Selected Roles list.
34. Click OK.

You are returned to the Create User page.
35. Click Object Privileges.
36. From the Select Object Type list, select View, and click Add.
37. Click the icon adjacent to the Select View Objects text box.

The Select View Objects window appears.
38. From the Schema list, select SYS.
39. In the Search View Name text box, type the following entry:

   DBA_PENDING_TRANSACTIONS

40. Click Go.

The object is now listed at the bottom of the window.
41. Click DBA_PENDING_TRANSACTIONS, and click Select.

The Add View Objects Privileges page appears.
42. In the Available Privileges list, click Select.
43. Click Move to move the item to the Selected Privileges list.
44. Click OK.

You are returned to the Create User page.
45. Click OK.

A message appears indicating the object was created successfully.
46. In the top right corner of the console, click Logout to exit the Enterprise Manager.
47. Proceed to Updating the Database.
**Updating the Database**

You must update your database to verify that it will run correctly with the SIMULIA Execution Engine. To update the database, you need to log in as a sysdba and execute several `grant` commands.

1. Open a Command Prompt dialog box (terminal window on Linux).
2. If you are updating a database that is running on a Linux system, verify that the `ORACLE_SID` environment variable is set to the correct database.
3. Execute the following command to connect to Oracle’s SQLPlus utility, where `<password>` is the sys user password specified during database creation:

   ```
   sqlplus sys/<password> as sysdba
   ```

   If you followed the instructions in *Creating the Database in Oracle* to create the database, this password is probably `seeadmin`.

   **Note:** If Oracle is not defined in your path or your `ORACLE_HOME` environment variable is not set, you will have trouble executing the command. Set the necessary system information, or navigate to the `<oracle_install_directory>in` directory, and execute the command from the directory. If you have more than one database running on your system, it may be necessary to set the `ORACLE_SID` environment variable to ensure that you are connecting to the correct database.

   You are connected to the SQLPlus utility.

4. Execute the following commands by typing each command individually and pressing the Enter key:

   ```
   grant select on pending_trans$ to public;
   grant select on dba_2pc_pending to public;
   grant select on dba_pending_transactions to public;
   grant execute on dbms_system to seeadmin;
   ```

   **Note:** The `seeadmin` user name in the last command refers to the Oracle user account that was defined when the database was created. Be sure that your database uses the same user name or substitute the appropriate user name in this command.

5. Type `exit` to close the SQLPlus utility.

   The database update is complete.

6. Continue to *Creating the Database Tables*.

---

**Creating the Database Tables**

The final step in initializing the database is to create the database tables. These tables are created by executing the `createtables` command that is included with your SIMULIA Execution Engine installation.

Be sure to execute the command in the following procedure on the computer containing the database (the computer running Oracle).

1. Open a Command Prompt dialog box (terminal window on Linux).
2. Navigate to the following directory:

   `<SEE_install_dir>/<os_dir>/reffiles/MAPIserver/db/oracle/`

   where `<os_dir>` is one of the following:

   • `/win_b64/` for Windows 64-bit
   • `/linux_a64/` for Linux 64-bit

3. Type one of the following commands, based on your operating system (where `user_name` is the name of the user account created for the database (usually `seedadmin`), `password` is this user’s password, and `databasename` is the name of the database that will hold the tables):

   • Windows: `createTables user_name password databasename`
   • Linux: `./createTables user_name password databasename`

   When using the `createTables` script, you should note the following:

   • If you created a database whose name is greater than eight characters, only type the first eight characters of the database name when using the script. For example, if your database is called `seedatabase`, you should only type `seedatab` when using the `createTables` command.
   • If your database is running on a Linux system, be sure that your `ORACLE_HOME` environment variable is set to your Oracle installation directory and that the path to the Oracle `bin/` directory is in your path. If these variables are not set properly, the script will not function correctly.

4. Verify correct script operation by examining the `createTables.log` file, which is located in the same directory as the `createTables` command itself.

Installing the Oracle JDBC Driver

If you are using an Oracle database, you must install the proper Java Database Connectivity (JDBC) driver file.

1. Obtain a copy of the Oracle JDBC driver file, which is typically named `ojdbc6.jar`. You can get the file either from your Oracle installation or from `http://www.oracle.com`.

2. Copy the file into the `<SEE_install_dir>/config/tomcat/lib/` directory of the SIMULIA Execution Engine installation.
TomEE Configuration

After you have successfully installed the TomEE server, you can start the server and the database.

Configuring TomEE with Derby or Oracle

You can use the TomEE Configuration Utility to set up the application server with either database.

Configure TomEE with Derby

1. Start the TomEE Configuration Utility with the following batch file or script:
   - On Windows:
     `<SEE_install_dir>\config\tomcat\bin\config.bat`
   - On Linux:
     `<SEE_install_dir>/config/tomcat/bin/config.sh`

   This step is not necessary if the installer has already started the configuration utility.

2. Select Derby as the database.

3. Select either Configure existing database or Configure new database.

4. If you chose Configure existing database, browse to the existing Derby installation directory.

5. If you chose Configure new database, Derby will be installed at the default location.

6. Click Next to define the LDAP configuration.

7. Select one of the following:
   - Secured Mode (configure TomEE in secured mode).
   - Unsecured Mode (configure TomEE in unsecured mode).
   - Skip LDAP Configuration (skip the LDAP security configuration if you have already configured the mode and do not intend to change it).

8. If you chose Secured Mode, enter the following information:
   - Connection URL: The location and the port number for the LDAP server. Typically, the port is 389.
     Use the following format:
     `ldap://10.10.10.10:389/`
     or
     `ldap://ldapservername:389/`
   - Connection Name: The distinguished name that is used to connect to the LDAP repository. For more information, consult your local system administrator. You can also find the information at the following URL: http://msdn.microsoft.com/en-us/library/aa366101%28v=vs.85%29.aspx.
   - Connection Password: The password for the user name.
   - User Base: The base entry for a user search. For more information, consult your system administrator.
   - Role Base: The base entry for a role search. For more information, consult your system administrator.
9. Assign user(s) to the required roles by entering a list of user names separated by semicolons.

10. Click Next to define the TomEE configuration.

11. Enter the TomEE Start Port. The default is 8080.

12. Enter the EJBD Port. The default is 4201.

This port number will be reflected in the tomeecpr connection profile file, which will be used by all SIMULIA Execution Engine clients to connect to the SIMULIA Execution Engine server.

13. Select Install TomEE as a service to configure the TomEE process as a service.

14. Click Finish and then OK to complete the utility.

Configure TomEE with Oracle

The TomEE Configuration Utility also supports the use of an Oracle database.

**Before you begin:** Before starting these steps, be sure you have completed all the steps described in *Configuring an Oracle Database*.

1. Start the TomEE Configuration Utility with the following batch file or script:

   - On Windows:
     ```bash
     <SEE_install_dir>\config\tomcat\bin\config.bat
     ```

   - On Linux:
     ```bash
     <SEE_install_dir>/config/tomcat/bin/config.sh
     ```

   This step is not necessary if the installer has already started the configuration utility.

2. Select Oracle as the database.

3. Under Oracle Database Configuration with TomEE, enter the following values for your Oracle database:
   - Hostname
   - Port
   - SID
   - Username
   - Password

4. In File Manager Setting, enter the path for the file manager.

5. Click Next to define the LDAP configuration.

6. Select one of the following:
   - **Secured Mode** (configure TomEE in secured mode).
   - **Unsecured Mode** (configure TomEE in unsecured mode).
   - **Skip LDAP Configuration** (skip the LDAP security configuration if you have already configured the mode and do not intend to change it).

7. If you chose **Secured Mode**, enter the following information:
   - **Connection URL**: The location and the port number for the LDAP server. Typically, the port is 389.
     
     Use the following format:
     ```bash
     ldap://10.10.10.10:389/  
     ```
     
     or
     ```bash
     ldap://ldapservername:389/  
     ```
• **Connection Name**: The distinguished name that is used to connect to the LDAP repository. For more information, consult your local system administrator. You can also find the information at the following URL: [http://msdn.microsoft.com/en-us/library/aa366101(v=vs.85).aspx](http://msdn.microsoft.com/en-us/library/aa366101(v=vs.85).aspx).

• **Connection Password**: The password for the user name.

• **User Base**: The base entry for a user search. For more information, consult your system administrator.

• **Role Base**: The base entry for a role search. For more information, consult your system administrator.

8. Assign user(s) to the required roles by entering a list of user names separated by semicolons.

9. Click **Next** to define the TomEE configuration.

10. Enter the **TomEE Start Port**. The default is 8080.

11. Enter the **EJBD Port**. The default is 4201.

   This port number will be reflected in the `tomee.cpr` connection profile file, which will be used by all SIMULIA Execution Engine clients to connect to the SIMULIA Execution Engine server.

12. Select **Install TomEE as a service** to configure the TomEE process as a service.

13. Click **Finish** and then **OK** to complete the utility.

## Starting TomEE

You can manually start the server.

**Before you begin**: If you are using an Oracle database, be sure that the database instance is started and configured as described in *Configuring an Oracle Database* and *Configure TomEE with Oracle*. If you are using a Derby database, be sure you have completed all the steps in *Configure TomEE with Derby*.

1. Browse to the following directory:
   
   `<SEE_install_directory>/config/tomcat/bin`

2. Do one of the following:
   
   • Windows: Double-click the `tomcat_start.bat` file.
   
   • Linux: Open a new terminal and run the `tomcat_start.sh` command.

## Starting the Derby Database

Starting TomEE automatically starts the Derby database. You can manually start the Derby database, if desired.

1. Browse to the following directory:
   
   `<SEE_install_directory>/config/tomcat/bin`

2. Do one of the following:
   
   • Windows: Double-click the `derby.bat` file.
   
   • Linux: Open a new terminal and run the `derby_start.sh` command.
Stopping TomEE

You can manually stop the server.

1. Browse to the following directory:
   
   <SEE_install_directory>/config/tomcat/bin

2. Do one of the following:
   - Windows: Double-click the tomcat_shutdown.bat file.
   - Linux: Open a new terminal and run the tomcat_shutdown.sh command.

Configuring Client Authentication

You must configure user authentication in your TomEE server.

About Client Authentication

The client authentication feature enables basic application server security by requiring any client connecting to
the SIMULIA Execution Engine supply credentials (for example, user ID and password).

The credentials are verified against a security domain defined by the SIMULIA Execution Engine administrator.
If the credentials pass the security check, the logon is allowed; otherwise, it is rejected. A client that passes this
security check is said to be authenticated, which means that the identity of the client has been established.

A client is any program running on any computer in the network that attempts to contact the SIMULIA Execution
Engine. SIMULIA Execution Engine clients include the Isight Design Gateway, SIMULIA Execution Engine
stations, the SIMULIA Execution Engine Dashboard, and the SIMULIA Execution Engine Command Line
Client. Each of these applications must provide valid user credentials to log on to the SIMULIA Execution
Engine and to perform any SIMULIA Execution Engine-related operations.

Once a client has provided valid credentials and is authenticated, those credentials can be used to determine
access to specific resources and information. All other SIMULIA Execution Engine security features are built
upon the authenticated credentials, which is why enabling this feature is a prerequisite to all other SIMULIA
Execution Engine security features.

About Security Roles

There are three types of SIMULIA Execution Engine user roles provided, each of which is identified by a security
role.

Each security role to which a user belongs enables that user to use a particular set of SIMULIA Execution Engine
features. The SIMULIA Execution Engine security roles are as follows:

- **fiperuser.** This role enables basic access to the SIMULIA Execution Engine system. All users (including
  administrators) must be associated with this security role. Users in this role can run the Isight Design Gateway
  and all other client programs except for SIMULIA Execution Engine stations. When using the Dashboard,
  users with this access level cannot view or change Access Control List configurations or shutdown SIMULIA
  Execution Engine stations.

- **fiperadmin.** This role enables access to SIMULIA Execution Engine administrative features. Users with
  this security role must also have the **fiperuser** security role. SIMULIA Execution Engine administrators can
  use all features of the Dashboard, shutdown SIMULIA Execution Engine stations remotely using the Dashboard
  or the Command-Line Client, and to perform other restricted administrative functions.
• **fiperstation.** This role provides the ability to run a SIMULIA Execution Engine station. Users with this security role also have the **fiperuser** security role and are the only users allowed to run a SIMULIA Execution Engine station (which can perform work on behalf of all users of the SIMULIA Execution Engine). In production deployments, this role will be given only to restricted user IDs created solely for the purpose of running the SIMULIA Execution Engine stations.

**Using an LDAP Server for Client Authentication**

The server is configured to authenticate all incoming connection requests against a particular security back-end infrastructure by using LDAP. You can define LDAP settings and user mappings during installation by using the **TomEE Configuration Utility**.

1. Launch the **TomEE Configuration Utility** from one of the following:
   ```
   <SEE_install_directory>/config/tomcat/bin/config.bat
   
   or
   
   config.sh
   ```

2. Go to the LDAP wizard.
3. Select **Secured Mode**.
4. Enter the following information:
   • **Connection URL**: The location and the port number for the LDAP server.
   • **Connection Name**: The user name that is used to connect to the LDAP repository.
   • **Connection Password**: The password for the user name.
   • **User Base**: The base entry for a user search. For more information, consult your local system administrator.
   • **Role Base**: The base entry for a role search. For more information, consult your local system administrator.
5. Assign user(s) to the required roles by entering a list of user names separated by semicolons.
Configuration

This section describes additional steps that may be necessary to configure the application.

Setting the DSLS_CONFIG Environment Variable

If your SIMULIA Execution Engine application uses Dassault Systèmes (DS) licensing instead of FLEXnet licensing, you must set the DSLS_CONFIG environment variable to allow the application to find the licensing client configuration file (DSLicSrv.txt).

Before starting the SIMULIA Execution Engine, be sure this environment variable is set to the full path of the DSLicSrv.txt file in your installation:

```
DSLS_CONFIG=/<SEE_install_dir>/config/DSLicSrv.txt
```

For complete information about licensing, see the Isight Installation Guide.

Configuring Your FLEXnet License to Work with a Windows Firewall

If you run a FLEXnet license server that is behind a Windows Firewall, you need to edit your client license file for the SIMULIA Execution Engine to ensure that it can connect to the license server it is started.

1. Verify that the Windows Firewall has been updated so that the necessary license ports are open for the license server. For more information, contact your local system administrator.
2. Navigate to the following directory:
   
   `<SEE_install_dir>/config/
   
3. Open the license.dat file in the text editor of your choice.
4. Add the port number opened on the license server computer’s Windows Firewall to the SERVER line in your license file. For example, if port 1700 was opened on the Windows Firewall, your license SERVER line would appear similar to the example shown below:
   
   ```
   SERVER seemachine ANY 1700
   ```

   For more information on what port numbers were opened on your license server’s Windows Firewall, contact your local system administrator.
5. Save and close your license file.

Setting Station Execution Permissions for the Excel and Word Components

If you will be allowing the execution of Excel or Word components on any SIMULIA Execution Engine station that you install, you must have launch and activation permissions for Excel and Word, especially if the SIMULIA Execution Engine station is being run as a service.

1. Perform one of the following steps, based on your operating system:
   - Windows Server 2003: Click the Start button, and click Run.
• Windows Server 2008: Click the Start button.

2. Perform one of the following steps, based on your operating system:
   • Windows Server 2003: In the Open text box, type dcomcnfg, and click OK.
   • Windows Server 2008: In the Start Search text box, type dcomcnfg, and click ENTER.

   The Component Services dialog box appears.

3. On the left side of the dialog box, click Component Services.

   Folder options appear on the right side of the dialog box.


5. Double-click DCOM Config.

6. Right-click one of the following icons:
   • Microsoft Excel Application
   • Microsoft Word Document or Microsoft Office Word Macro-Enabled Document

   If you cannot find the icons, use the mmc comexp.msc /32 command. If you cannot find an entry for Microsoft Word, look for a string named {0020906-0000-0000-C000-000000000046} to right-click instead.

   However, before you alter this string’s settings, you should access the properties of this string and confirm that the local path points to the WINWORD.EXE program.

7. Select Properties.

   The Properties dialog box appears.

8. Click the Security tab.

9. In the Launch and Activation Permissions area, click Customize, and click Edit.

   The Launch Permission dialog box appears.

10. Click Add.

    The Select Users, Computers, or Groups dialog box appears.

11. In the Enter object names to select text box, type the necessary user name (be sure to include the computer/domain name).

    You can click Check Names to verify that the user name you entered is valid. You can also search for the name by clicking Advanced. If the user name you specify matches more than one known user, the Multiple Names Found dialog box appears, allowing you to pick the exact user.

12. Click OK.

    You are returned to the Launch Permission dialog box, and the user name you entered now appears in the list at the top of the dialog box.

13. In the Permission for <user_name> area, click Local Launch and Local Activation in the Allow column.

14. Click OK.

    You are returned to the Properties dialog box.

15. Click OK.

    You are returned to the Component Services dialog box.
16. If needed, repeat step 6 through step 15 for either Excel or Word (whichever application you didn’t configure the first time).

17. Close the Component Services dialog box.

### Setting Word and Excel Privileges for Stations Installed as a Service

If you are running a station as a service on a non-secure SIMULIA Execution Engine, you must update the station service’s properties prior to executing models that use the Excel or Word components. This action is not necessary if you are running on a secure SIMULIA Execution Engine.

1. Verify that the SIMULIA Execution Engine station is installed.
2. Verify that you have complete the procedure described in Setting Station Execution Permissions for the Excel and Word Components.
3. Access the Services dialog box as described in Installing a SIMULIA Execution Engine Station as a Service.
4. Right-click the Fiper Station entry, and select Properties.

   The Properties dialog box appears.

5. Click the Log On tab.
6. Click This account, and enter the account name and password for the account that was given privileges in Setting Station Execution Permissions for the Excel and Word Components.
7. Click OK to save your changes.
8. Close the Services dialog box.

### Setting Excel or Word Options for Windows Stations

This section explains the steps to take when you are executing Microsoft Excel or Word components on SIMULIA Execution Engine Windows Stations.

When you have concurrent Excel jobs executing on SIMULIA Execution Engine stations running in service mode, the stations can fail because the cache overflow or the idle timeout limits are exceeded. You may see the following errors:

- `com.engineous.sdk.exception.SDKException:Error loading Add-ins`
- `Maximum execution time exceeded for component`

To prevent the job from failing, you must complete all the steps in this section. It is important to restart the computer after you complete all the steps.

### Configuring the Excel Properties and Component

You can configure Excel component execution options such as timeout value and when to close the workbook.

1. With a model open in the gateway that contains the Excel component, right-click the Excel component icon.

2. Click Properties, and then click the Execution tab.
3. In the General Options Timeout text box, enter 0.

   If set to 0, no timeout limit is enforced. For more information, see Configuring the Execution Properties.
4. Click OK to save your changes.

5. From the gateway, double-click the Excel component.
   a. Select the Advanced tab.
      Click Close workbook to close the opened workbook (when not selected, Excel and the workbook remain open after the model is executed), and then from the adjoining list select when job completes.
      For more information, see Using Advanced Options.
   b. Repeat the above step for each Excel component.

6. Click OK to save your changes and to close the Excel Component Editor.

**Configuring station.properties Values**

You can add several options to the station.properties file to reduce the risk of jobs failing when executing Excel or Word components on Windows stations.

1. Add the following properties to the station.properties file. You can vary the values according to the load.
   - fiper.station.substation.starttime=300000
   - fiper.station.substation.launchtimeout=300000
   - fiper.station.substation.keepalive.interval.ms=120000
   - fiper.security.substation.cache.size=<double to the number of expected users>+5

   For more information, see Configuring SIMULIA Execution Engine Station Properties.

2. Save the station.properties file.

**Replacing a Process Level Token**

You can replace a process level token to reduce the risk of jobs failing when executing Excel or Word components on Windows stations.

1. For each user who will start a SIMULIA Execution Engine station, and the user to the local Administrators group and grant the user the privilege Replace a process level token in the Local Security Policy dialog box.
   For more information, see Configuring SIMULIA Execution Engine Stations for Run-As on Windows.

2. Save your changes.

**Verifying that the Desktop Folder Exists**

You must ensure that the desktop folder exists on the stations running the jobs.

1. Verify that the following directory exists:
   - For MS Office 2007 (32-bit), 2010 (32-bit), 2013 (64-bit) C:\Windows\SysWOW64\config\systemprofile\Desktop
   - For 2010 (64-bit) C:\Windows\System32\config\systemprofile\Desktop

2. Verify that the directory is writeable for all run-as users.
Configuring the Excel or Word Macro Security Settings
You can configure Excel macros to reduce the risk of jobs failing when executing Excel or Word components on Windows stations.

1. Open a file:
   • From Excel, open a worksheet.
   • From Word, open a document.

2. From the File menu, select Options, and then select Customize Ribbon.
3. From the Main Tabs list, select Developer.
4. Select OK.
5. From the Developer tab, select Macro Security, and then select Macro Setting.
6. From the Developer Macro Setting list, select Trust access to the VBA project object model.
7. Close Excel or Word.

Disabling Excel Add-ins
You can disable the Excel add-ins to reduce the risk of jobs failing when executing Excel components on Windows stations.

1. Click the Windows Start button, and then select Control Panel.
2. From the Control Panel, select Uninstall a program.
   a. Select Add or Remove Features, and click Continue.
   b. Expand Microsoft Excel.
   c. Expand Add-ins.
   d. Disable all the add-ins.
   e. Click Continue to configure the setup.
   f. Click Close.
4. Close the Control Panel.

Editing the Registry
You can edit the Microsoft registry to reduce the risk of jobs failing when executing Excel components on Windows stations.

1. Click the Windows Start button, and in the search text box enter regedit to access the Registry Editor (regedit.exe).
   The Registry Editor appears.

2. Navigate to the following directory:
   HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager\Memory Management
3. Select SessionViewSize.
   a. In the Value data text box, enter 80.
   b. Click OK.

4. Navigate to the following directory:
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager\SubSystems
a. Select Windows.
b. In the Value data text box, locate the Windows SharedSection parameter.
The parameter has the following formats:
   • SharedSection=aaaa,bbbb
   • SharedSection=aaaa,bbbb,cccc
   • SharedSection=aaaa,bbbb,cccc,dddd
c. Add or edit the third value as follows:
   Windows SharedSection=aaaa,bbbb,10240
d. Click OK.
5. Close the Registry Editor.

Editing User Account Controls
You can change the user account controls to reduce the risk of jobs failing when executing Excel or Word components on Windows stations.
1. Click the Windows Start button, and then select Control Panel.
2. From the Control Panel, select User Accounts, and then select Change User Account Control settings.
a. Move the slider bar to Never notify.
b. Click OK.
3. Close the Control Panel.

Creating an Environment Variable
You can create an environment variable that has read and write permissions for all SIMULIA Execution Engine users.
1. Click the Windows Start button, and then select Control Panel.
2. Select System and Security, and then select Advanced System Settings.
3. Click Environment Variables, and select New.
a. In the Variable name text box, enter the following:
   FIPER_TEMP
b. In the Variable value text box, enter a value, such as the following:
   C:\SIMULIA\StationTemp\2016
4. Give read and write permission to all SIMULIA Execution Engine users.
5. Click OK.
6. Close the Control Panel.
7. Restart the computer.
Configuring Security

This section describes the security options available with the SIMULIA Execution Engine and how to configure them.

About SIMULIA Execution Engine Security

The SIMULIA Execution Engine is a distributed computing infrastructure with a wide range of security features implemented at different levels, at different points in the infrastructure, and using different operating system and middleware facilities. This section describes how an administrator can use some of these features to secure the overall computing environment.

The SIMULIA Execution Engine administrator can pick and choose which features to implement and can add features over time to improve the security of the system. It might be desirable to start with an open system for prototyping or proof-of-concept implementations and then apply more security features as the needs of your organization dictate. It is important for the SIMULIA Execution Engine administrator to read and understand the significance of each security feature to decide if it is appropriate for a specific environment.

Some aspects of SIMULIA Execution Engine security are provided by the native operating systems (such as file access security), some are provided by database and application server middleware, and some are built into the SIMULIA Execution Engine system itself. Some basic knowledge of all of these areas is necessary to make a determination of which features should be applied in a given environment. In addition, a basic understanding of the security tools provided by those systems is also essential to configure and deploy SIMULIA Execution Engine security features. This section provides step-by-step instructions for activating these security features, as well as a discussion of the systems involved, which will aid the SIMULIA Execution Engine administrator in determining the proper configuration for a specific computing environment.

Some security features described in this section interact with other SIMULIA Execution Engine features or have prerequisites. Such interactions and prerequisites are described in each section as appropriate.

About Database Security

Regardless of which SIMULIA Execution Engine security features are used, the SIMULIA Execution Engine database is always protected by the application server and database middleware layers.

The only access to the database is via the application server. The application server is configured with the proper credentials to access the database and maintains the only secure connection with the database. The database is not directly exposed to any SIMULIA Execution Engine user. The SIMULIA Execution Engine does not support direct database connections, but it does not prevent them if the database administrator has given such access using tools outside of the SIMULIA Execution Engine.

About the Open (Non-secure) Configuration Option

The default installation of the SIMULIA Execution Engine provides an open environment. In this type of environment, no security is provided and user access is not controlled via user names and passwords.

Most other security features of the SIMULIA Execution Engine are disabled or ineffective in this mode. For example, although Access Control Lists can be defined, the control lists are not useful since all users share a single user ID. Furthermore, SIMULIA Execution Engine Federation (B2B) features will not function in this open configuration. This mode of operation is useful to verify correct operation of the system during a new installation or for prototyping and proof-of-concept environments.
Configuring SIMULIA Execution Engine Security

You must configure user authentication and other security features in your Java application server.

About Client Authentication

The client authentication feature enables basic application server security by requiring that any client connecting to the SIMULIA Execution Engine supply credentials (e.g., user and password).

About SIMULIA Execution Engine Access Control Lists

The SIMULIA Execution Engine provides a means to limit access to specific data and information stored in the SIMULIA Execution Engine database. In particular, SIMULIA Execution Engine library objects (models and components) and job results can be protected with Access Control Lists (ACLs). An access control list contains a set of permission levels and names of users or groups.

No explicit administrative action is needed to enable the Access Control List feature. This feature is always available in the SIMULIA Execution Engine. The ACL feature should, however, be configured as described in this section to achieve the level of default permissions as required by the organization.

This feature is not useful if each user cannot be distinguished and authenticated. Thus, the client authentication capability of SIMULIA Execution Engine is a prerequisite for using Access Control Lists.

The following permission levels are available:

- **ALTER**. The user or group has full access to the object, including the ability to edit the object's permissions. The object can be fetched (copied to a local library), new versions of the object can be published to the library, and any version of the object can be deleted from the library.

- **MODIFY**. The user or group has all the accessibility granted with the ALTER option, with the exception of editing the object's permissions.

- **READ**. The user or group can only load or use the object by reference. Although the model and its contents (components, simulation process flows, parameters, etc.) can be viewed and altered, and the model itself can be executed, no new versions of the model can be published to the library.

- **REFERENCE**. This protection level applies only to models stored in the SIMULIA Execution Engine library. This protection level is a limited read access that provides information about the model inputs and outputs but does not provide any access to the model structure or internal configuration. If this level of permission is set for a user who incorporates a published model into another model, the content of the referenced model is available and it can be executed by the reference.

- **NONE**. The user or group will have no access to the published object. Any model that references this object cannot be used.

🌟 Important: User names are case-sensitive.

For instructions on how to apply these permissions to library data and jobs, see *Setting Default Permissions* in the *Isight User’s Guide*.

An ACL system administrator can define new groups, add and remove users from groups, define default system-wide permissions, and add and remove other ACL system administrators. The Dashboard is used by the ACL system administrator to configure the ACL system settings. For more information on this interface, see *Using the Dashboard*.

The ACL system administrator should then use the Dashboard's **Access Control** tab to define the system-wide default permission settings. This tab is shown in the following figure.
Only an ACL system administrator will see the **Access Control** tab on the Dashboard. When the SIMULIA Execution Engine is first installed, the only users who are considered ACL system administrators are those who have been assigned the `fiperadmin` security role. These users can add other ACL system administrators using the **System Administration** subtab; these users will have complete control over all published objects and assigned permissions in the SIMULIA Execution Engine but will not have any administrative privileges.

The **System Default** subtab defines the permissions that will be applied to any object (model, component, job) that has no explicit permission for the requesting user.

For the most secure system, the **System Default** tab should have **All other users** set to **NONE**. This action will prevent access by any user to any data to which they are not otherwise given explicit permission to use. Setting this value to **ALTER** will give all users access to all data unless the author of the data explicitly prevents it.

**Configuring Station (Run-As) Security**

This section describes the SIMULIA Execution Engine station Run-As feature and how to configure it. This feature provides additional security at the station level of the SIMULIA Execution Engine.

The Run-as feature is considered an additional level of security above client authentication.

**About Station Run-As Security**

The station Run-As feature provides a means for work executed on SIMULIA Execution Engine stations to run in the security context of the job submitter. When this feature is not used, work done on SIMULIA Execution Engine stations on behalf of SIMULIA Execution Engine users is run in the single security context of the user that started the station.

The term “security context” refers to the operating system level security information about a particular user (e.g., it is the operating system’s identification of a particular user and the user’s associated permissions to system resources such as files, network resources, etc.). When any program is started, the operating system associates the program (process) with the security context of a particular user (usually the user that started it). The process has access only to the resources that the user is authorized for at the operating system level. For example, the process would only be able to access files for which the user had appropriate file permissions.
When the Run-As feature is not active, all work is performed in a single operating system process, meaning that SIMULIA Execution Engine jobs could, in theory, access system resources (such as files) on behalf of a user when that user did not have permission to access the file. Through the SIMULIA Execution Engine infrastructure, users could retrieve a file from the SIMULIA Execution Engine station for which they would not typically have permission. In an extreme case it would be possible to write an Isight component (such as the Script component) to access other user’s in-progress work on the SIMULIA Execution Engine station where the component executes. This situation can be partially mitigated by running stations with restricted user that have a minimal set of file access permissions.

The SIMULIA Execution Engine Run-As feature prevents jobs from accessing any resource to which the original job submitter does not have valid operating system level permissions. In particular, the job cannot access files for which the submitter’s own user ID does not have permissions, including other users’ in-progress work on the same station. From an operating system point of view, the work is run in a process that is started with the job submitter’s security context and, therefore, has only that user’s resource permissions.

When the Run-As feature is enabled, SIMULIA Execution Engine stations examine each incoming work item request. The work item contains the job submitter’s credentials in encrypted form (encryption techniques are described in About User Credential Encryption). The job submitter’s credentials are authenticated against the security domain (realm) configured by the system administrator. If the job submitter’s credentials do not authenticate, the work request is rejected. When the job submitter’s credentials are authenticated, a new process is started using those credentials. This secondary process is known as a “substation,” and it will perform the requested work on behalf of the job.

The substation process performs the work required for the job including any access to system resources (such as files). If the job attempts to access files for which the submitter does not have proper operating system permissions, the file access is denied by the operating system. All temporary files created by the SIMULIA Execution Engine as part of running the job will be protected. Only the job submitter has read or write access to the work-in-progress files. Thus, different users’ jobs will not be able to access the files of other users on the same station.

Substation processes are started for each distinct user on an as-needed basis in each SIMULIA Execution Engine station system in the network. The substation process continues to exist (suspended) after it completes a work assignment. If another piece of work for that same user arrives at that SIMULIA Execution Engine station, the already running substation process is awakened and reused. If a substation process remains inactive for a period of time, it will be terminated automatically. Inactive substations may also be terminated when a threshold on the number of processes is reached. When the main SIMULIA Execution Engine station shuts down, all substation processes that it started are also shut down.

About User Credential Encryption

For the SIMULIA Execution Engine station to start a new process (substation) on behalf of the SIMULIA Execution Engine user (job submitter), it must have the user’s credentials or be running as root (Linux only).

Credentials (password) are supplied by the user when any SIMULIA Execution Engine client connects to the SIMULIA Execution Engine. Those credentials are captured when a job is submitted and used by the stations to start processes (substations) in the job submitter’s security context.

For this procedure to be secure, it is necessary to transmit the user’s credentials to the SIMULIA Execution Engine (where they were captured during the log on process), and from the SIMULIA Execution Engine to the SIMULIA Execution Engine station where they will be used to start substation processes. The SIMULIA Execution Engine uses industry-standard PKI encryption technology with strong 1024-bit encryption for transmitting and storing user credentials. The SIMULIA Execution Engine uses a public/private key system to ensure secure communications.

The way in which credentials are used on SIMULIA Execution Engine stations is described below and shown in the following figure. Numbers in the figure correspond to the numbered steps below.
Figure 3: SIMULIA Execution Engine Encryption Process

1. When the SIMULIA Execution Engine is first started, it generates a secure random private/public key pair.
2. The SIMULIA Execution Engine connects to the database using the credentials defined by the administrator in the application server configuration.
3. On some system in the network, the SIMULIA Execution Engine station is started and generates its own secure random private/public key pair. The public key is sent to the SIMULIA Execution Engine, where it is stored with other station configuration details. The private key never leaves the SIMULIA Execution Engine station.
4. The user logs on to a local computer somewhere in the network and starts an interface that will access the SIMULIA Execution Engine (for example, the Isight Design Gateway).
5. The user provides credentials (user and password) for connecting to the SIMULIA Execution Engine.
6. The application server authenticates the user against the configured SIMULIA Execution Engine security domain. For more information on how this domain is configured, see Configuring SIMULIA Execution Engine Security.
7. The client program retrieves the SIMULIA Execution Engine’s public key. The SIMULIA Execution Engine’s private key never leaves the SIMULIA Execution Engine and is not available to any clients.
8. The client program encrypts the user credentials with the SIMULIA Execution Engine’s public key.
9. The client program submits a job and includes the encrypted credentials with the job request. The SIMULIA Execution Engine stores the encrypted credentials with the job details.
10. At some later time, the SIMULIA Execution Engine dispatches a work request for the job to a particular SIMULIA Execution Engine station. The user credentials (stored in the job details) are decrypted with the SIMULIA Execution Engine’s private key, and are then re-encrypted with the station’s public key.
11. The work request is sent to the station with the encrypted credentials.
12. When the work request is received by the SIMULIA Execution Engine station, the user credentials are decrypted with the station’s private key.
13. The SIMULIA Execution Engine station launches a new process (substation) by authenticating the user to the local operating system. If authentication fails, the new process is not created and the work request fails.
14. The new substation process (running in the security context of the job submitter) performs the requested work.

About Securing the SIMULIA Execution Engine Station File System

For a secure SIMULIA Execution Engine operating environment, it is important to consider the local file system of the SIMULIA Execution Engine station computer.

This file system is, in general, available to any SIMULIA Execution Engine job that runs on the station. The SIMULIA Execution Engine job can attempt to read or write any part of the file system, including network attached drives.
It is important, therefore, to properly secure the local file system against inadvertent or malicious use by a SIMULIA Execution Engine job. SIMULIA Execution Engine stations using the Run-As feature behave differently (from a file system security point of view) than stations running without this feature enabled.

**About File System Security Without Run-As**

Without the station Run-As feature, the SIMULIA Execution Engine station runs as a single process with the security identity of the user that starts the process. The station, therefore, has access to exactly the same set of files as the user that starts it.

If that user has complete access to the entire file system, all users that run jobs on that station also have access to that computer’s entire file system. SIMULIA Execution Engine jobs on such a system can read or write any file including operating system files or private user data.

The first step in securing the file system in this environment is to use a dedicated, restricted user to run the SIMULIA Execution Engine station process. The restricted user should be given access only to the parts of the file system needed for proper operation of the station and jobs that run there. In general, this setup requires read access to basic operating system files, read access to the SIMULIA Execution Engine installation directories, and read/write access to the SIMULIA Execution Engine station temporary directory. The temporary directory can be specified in the `station.properties` file, which is located at the top level of the SIMULIA Execution Engine or station installation directory. In general, it is best to specify a custom location for the temporary directory rather than using the default location, which is the user’s temporary directory.

The station will keep all work-in-progress files in the temporary directory. To prevent access to these files, that directory should be restricted such that only the dedicated station user has read/write access to it. All other users should have no access to this directory. This arrangement will prevent anyone from logging on to the station computer and having access to work-in-progress files.

In this environment it is impossible to prevent a SIMULIA Execution Engine job from (possibly) accessing another user’s unrelated work-in-progress files. All work-in-progress files are created by the same dedicated user, and all will be accessible from any other SIMULIA Execution Engine jobs. However, it would require some explicit effort for a SIMULIA Execution Engine job to access unrelated work-in-progress files, because they are kept in separate subdirectories. The station Run-as security feature must be used to prevent this type of file access.

**About File System Security With Run-As**

When the station Run-As feature is enabled, each user’s work is executed in that user’s security context. Access to files on the SIMULIA Execution Engine station computer will be dictated by the file system permission of the user who submitted the job.

In general, the system running the station must be configured to allow read access for all users to parts of the file system needed to run the station, which includes the basic operating system files and the SIMULIA Execution Engine installation directory. The administrator can choose to make these directories readable for all users or only for those users that will be running SIMULIA Execution Engine jobs on the station. The SIMULIA Execution Engine installation directory should not be configured with write permission for users running jobs.

The station will keep all work-in-progress files in the temporary directory. Within that directory, each substation (user) will create a subdirectory with permissions that allow access only by the submitting user. This prevents one user’s job from accessing work-in-progress files from another user’s job.

The station should be configured with an explicit temporary directory by specifying a directory name in the `station.properties` file (located at the top level of the SIMULIA Execution Engine or station installation directory). On Windows, this directory should be configured with the following permissions for all users that will run jobs on the station:

- read
- write
• execute
• create folders
• create files

Note: You can grant all the necessary permissions using the Modify option available from the Security tab on the directory’s Properties dialog box. For more information, contact your local system administrator.

All other users should have no access to this directory to prevent casual users logged on to the system from accessing work-in-progress files. If the system is physically secure or there is no threat from logged on users, these permissions can be granted to all users.

About Run-As Security Limitations
Some limitations exist with regard to using the SIMULIA Execution Engine station Run-As security feature. These limitations should be reviewed prior to activating the feature.

The following station Run-As security limitations exist:

• On Windows 7 and Windows 2008 Server, a Run-As station should be started as a service as user LOCAL SYSTEM or LOCAL SERVICE. If a Run-As station is started interactively, it must be started as a user in the Administrators group who has been granted privilege Replace a Process Level Token and must be started by right-clicking and selecting Run as Administrator (or be started from a command window that was started with Run as Administrator). Be aware that there are known problems with the Excel and Word components on Run-As stations on Windows Vista and later—the component will sometimes hang, leaving multiple Excel or Word processes running on the machine. It is recommended that Excel and Word not be run on Run-As stations on Windows Vista or later. Either use an earlier version of Windows, or use a non-Run-As station (see Station-Specific Run-As Behavior and Setting Station-Specific Run-As Options).

• On Windows, a Run-As station may create 2 substation processes for each user. These substations are created differently so they have different operating system privileges. One is specifically for the OSCommand and Simcode component, and the other is for COM components like Excel and Word. This is necessary because a station spawned one way cannot create Windows Job objects for OSCommand cleanup, and a station spawned the other way cannot create a COM server for Excel or Word. No specific action is required, but be aware that there may be twice as many substation processes as expected. Forcing all Excel and Word components to execute on a non-Run-As station (see previous item) avoids the extra substations on Run-As machines, and also avoids a problem with Excel on Windows Vista and later.

Configuring the Run-As Feature
All encryption key generation and management for the SIMULIA Execution Engine Run-As feature is automatic and requires no configuration. The system administrator only needs to enable the feature (it is disabled by default) and specify the security domains for SIMULIA Execution Engine stations to authenticate user credentials.

Unless each station is manually configured, all stations will use the same security domain (realm) to authenticate user credentials. It is possible to have different stations authenticate to different security domains, but each station can use only a single domain.

When the SIMULIA Execution Engine and the stations use separate security domains, it is necessary that users have a common user name and password for all domains in which their job executes. It is not possible to authenticate a single job with multiple user names and passwords.

Configuring the SIMULIA Execution Engine for Run-As
To enable the station Run-As feature, you first need to configure your SIMULIA Execution Engine properties file to recognize the feature when a station using it connects to the server.
1. Verify that client authentication is enabled in your SIMULIA Execution Engine. Client authentication must be enabled to use the Run-As feature.

2. Navigate to the following directory of your SIMULIA Execution Engine installation:
   
   ```
   <see_install_dir>/config/
   ```

3. Open the `acs.properties` file using the text editor of your choice.

4. Locate the following lines at the end of the file:
   
   ```
   #fiper.security.runas.enabled=true
   #fiper.security.runas.domain=
   ```

5. Remove the leading `#` character from the `fiper.security.runas.enabled` setting.
   If the `#` character is not present for the first setting, your SIMULIA Execution Engine is already configured to use Run-As. This option was probably enabled during the SIMULIA Execution Engine installation process. Proceed to Configuring SIMULIA Execution Engine Stations for Run-As.

6. If you are installing the SIMULIA Execution Engine on Windows, do the following:
   a. Remove the leading `#` character from the `fiper.security.runas.domain=` setting.
   b. Add the name of the appropriate Windows domain (which all Windows-based SIMULIA Execution Engine stations will use to authenticate user credentials) to the end of the setting (after the `=` character). Unless a SIMULIA Execution Engine station is explicitly configured otherwise, this domain name will be used by all Windows-based stations.

7. Save and close the `acs.properties` file.

8. Stop and restart .
   This step is necessary for the changes to the `acs.properties` file to take effect.

### Configuring SIMULIA Execution Engine Stations for Run-As

Once you have updated the SIMULIA Execution Engine properties file to recognize the Run-As feature, you need to update the individual station installations to use the feature.

This process varies based on the operating system the station is using. The following instructions describe the configuration steps for Windows-based and Linux-based stations.

This procedure must be performed on each computer that will run a SIMULIA Execution Engine station.

### Configuring SIMULIA Execution Engine Stations for Run-As on Windows

To configure a Windows-based station to use the Run-As security feature, you need to update the station’s properties file, set permissions on the station’s temporary directory, and update the system rights for the users that will be executing on the station.

**Note:** This procedure must be performed on each system that will run a SIMULIA Execution Engine station.

1. Perform one of the following steps, based on your security arrangement:
   - If you plan on using the same security domain for the SIMULIA Execution Engine station as specified for a Windows-based SIMULIA Execution Engine, proceed to step 5. This SIMULIA Execution Engine setting is described in Configuring the SIMULIA Execution Engine for Run-As.
   - If you plan on using a different security domain as specified for a Windows-based SIMULIA Execution Engine, proceed to step 2.
If you are connecting to a Linux-based SIMULIA Execution Engine, proceed to step 2.

2. Edit the `<see_install_dir>\config\station.properties` file and remove the leading # character from the following setting:

```
#fiper.security.station.domain
```

3. Enter the appropriate Windows domain by replacing the DEV string with your Windows domain name. Do *not* replace the entire line.

   The SIMULIA Execution Engine station will now use this domain to authenticate users’ credentials instead of the domain specified in the `acs.properties` file.

4. Save and close the `station.properties` file.

5. Grant read, write, execute, create folders, and create files access (or full access) on the SIMULIA Execution Engine station temporary directory to all users that may submit jobs. This temporary directory is specified in the `station.properties` file. For more information on locating this setting in the `station.properties` file, see *About File System Security With Run-As*.

   The steps necessary for granting this access to the SIMULIA Execution Engine station temporary directory differ slightly across Windows operating systems and may require special access rights. For more information, contact your local system administrator.

   This step must be performed on all systems that will be running SIMULIA Execution Engine stations.

6. Locate the system user rights as described below (based on your operating system):

   - *Windows Server 2003*: Navigate to *Start* / *Administrative Tools* / *Local Security Policy*; and from the *Local Security Settings* dialog box, access the *User Rights Assignment* settings under the *Local Policies* option.
   
   

7. For each user who will start a SIMULIA Execution Engine station, add the user to the local Administrators group and grant the user the privilege *Replace a process level token* in the *Local Security Policy* dialog box.

   For more detailed information on setting these user rights, contact your local system administrator.

8. Log out and log back on to the system. This step is necessary for the privilege changes to be recognized.

9. Repeat step 1 through step 8 for each system that will be running a SIMULIA Execution Engine station.

   **Important:** If the station is configured as a Windows service, the user starting the service must either be the default service account `LocalService` or be a member of the Administrators group who has been granted the *Replace a process level token* privilege. A non-administrator account will not work. For more information on these users settings and their privileges, contact your local system administrator.

10. Stop and restart TomEE for the associated SIMULIA Execution Engine. For more information, see *TomEE Configuration*.

    The Run-As configuration is complete. Your SIMULIA Execution Engine and stations will now use Run-As security.
Configuring SIMULIA Execution Engine Stations for Run-As on Linux

To configure a Linux-based station to use the Run-As security feature, you need to update the station’s properties file and add a new file to the system’s `/etc` directory. This procedure must be performed on each system that will run a station.

1. Verify that you installed the SIMULIA Execution Engine station as root.

   For the Run-As feature to work on a Linux based operating system, the SIMULIA Execution Engine installation must have been performed as root. If you did not install the SIMULIA Execution Engine station as root, follow the instructions in *Enabling the SIMULIA Execution Engine Station Security Feature (Run-As)*.

2. Open the `<see_install_dir>/config/station.properties` file using the text editor of your choice.

   This file is located in the top level of the SIMULIA Execution Engine installation directory.

3. Verify that the leading `#` character has been removed from the `fiper.station.tempdir` setting.

4. Set the `fiper.station.tempdir` value to some location that is world-writable—for example, something similar to `/var/tmp/<SEETempDir>`.

   **Important:** If this directory already exists, it must not be owned by another user.

5. Save and close the `station.properties` file.

6. To run a SIMULIA Execution Engine station with the Run-As feature on Linux, create a file (in mode 644) called `fiper` in the `/etc/pam.d/` directory with the contents shown below. On Red Hat Linux, create the file `fiper` as root.

   **Red Hat Linux:**

   ```
   #%PAM-1.0
   auth       include    system-auth
   account    include    system-auth
   password   include    system-auth
   session    include    system-auth
   ```

   **SUSE Linux:**

   ```
   #%PAM-1.0
   auth       include    common-auth
   account    include    common-account
   password   include    common-password
   session    include    common-session
   ```

   **Note:** The contents specified above are the standard settings. To verify that your settings are the same, examine the matching contents of the `login` file, located in the same `/etc/pam.d` directory. You can also copy this `login` file, rename it `fiper`, and then edit it to only contain the settings specified above.

7. Repeat step 2 through step 6 (as necessary based on the computer’s operating system) for each system that will be running a SIMULIA Execution Engine station.

8. Restart the TomEE server.

   The Run-As configuration is complete. Your SIMULIA Execution Engine and stations will now use Run-As security.
Setting Station-Specific Run-As Options

You can determine, at the station level, whether or not a station will use Run-As security.

For example, you can turn on the feature for the SIMULIA Execution Engine but disable it for specific stations in your environment. This feature allows you to have a mix of Run-As enabled and Run-As disabled stations that use the same SIMULIA Execution Engine. Furthermore, when a station is running on Linux, you can set it to execute in Run-As mode but without a password.

To set the station-specific Run-As option, you use the property fiper.station.runas in the station.properties file. For more information about this option, see Station-Specific Run-As Behavior.
Using SIMULIA Execution Engine Interfaces

This section describes the interfaces that a system administrator can use to control and monitor the SIMULIA Execution Engine and the SIMULIA Execution Engine stations. The interfaces include the station itself, the Dashboard, and the Command Line Client. These interfaces are intended for use by system administrators who install and maintain the SIMULIA Execution Engine system.

Alternatively, the SIMULIA Execution Engine WebTop interface is intended for use by Isight end-users who will perform basic operations such as running models and viewing results. These end-users should refer to the SIMULIA Execution Engine WebTop Guide.

Using the SIMULIA Execution Engine Station

SIMULIA Execution Engine stations are computers on the network that have been registered with the SIMULIA Execution Engine to provide services to the system and to handle the execution of workitems. The station software consists of a framework for receiving workitems, communicating with the library, executing components, and returning results.

The SIMULIA Execution Engine station is a long-running process that performs work on behalf of the SIMULIA Execution Engine. An instance of the station must be run on any computer that is to act as part of the distributed and parallel execution environment.

The SIMULIA Execution Engine station is a multithreaded Java application that can handle many concurrent requests. For an overview of the complete SIMULIA Execution Engine system, including stations, see About the SIMULIA Execution Engine Environment.

You can customize the operation of a SIMULIA Execution Engine station using the station.properties file or by using the equivalent command line arguments when you start the station. See Configuring SIMULIA Execution Engine Station Properties for more details.

About the SIMULIA Execution Engine Station Interface

The SIMULIA Execution Engine station interface shows you various items including status information and logging status.

The station interface appears similar to the example below.
This interface provides you with the following information and options:

- The name and working directory of the station.
- Any affinities that are defined for the station, including predefined affinities. See About Station Affinities for more information.
- The Detail level list allows you to filter the information that is displayed in the Station Log area. The following logging levels are available: Debug, Info, Warning, Error, and SysError. See About Log Message Detail Levels for more information.
- The Clear button allows you to clear the text displayed in the Station Log area.
- The Status button provides you with station information, such as the number of threads used and available, the number of workitems in progress, and the available station memory.
- When activated, the Auto scroll option automatically displays the most recent text messages in the Station Log area.
- The Shutdown button closes the station interface and stops all associated processes. See Shutting Down a SIMULIA Execution Engine Station for more information.
- The Renew Lease button forces the station to report its status to the SIMULIA Execution Engine. This notification process lets the SIMULIA Execution Engine know that the station is still functioning properly, which helps eliminate long delays associated with attempting to contact a nonfunctional station. This process is performed automatically after a certain period of time as defined by the fiper.station.leaseinterval property in the station.properties file. For more information on this setting, see Lease Interval.
- The numbers next to the SIMULIA Execution Engine connection button (in the lower-right corner of the interface) show the number of busy threads, the maximum number of threads, and the number of active threads.
- The lower-right corner of the interface displays the SIMULIA Execution Engine connection information. The name of the SIMULIA Execution Engine and the user name (if security is enabled) are displayed. Clicking this button displays additional station information in a separate dialog box.

**About Log Message Detail Levels**

The SIMULIA Execution Engine provides system monitoring and debugging information in log files, like all Java enterprise applications. You can filter the amount of information that is logged, choosing more or less details.

You can choose any of the following settings:

- **Debug.** These messages are intended for debugging system or component code. Typically, these messages are meaningful only to the program developers. This level can produce a large quantity of messages, which can affect system performance. This option is the lowest level, providing the most information. Using this setting will send all types of messages to your log file: debug, informational, warning, error, and system error.
- **Info.** These messages contain routine status or other informational items that are not generally significant. Choosing this setting will provide all Info messages as well as all Warning, Error, and SysError messages.
- **Warning.** These messages indicate a condition of which the end user should be aware but do not generally indicate a failure. Choosing this setting will provide all Warning messages as well as all Error and SysError messages.
- **Error.** These messages indicate an error condition that was caused by the end user, operational data, or some other condition that can be corrected. Choosing this setting will provide all Error messages as well as all SysError messages.
• **SysError.** These messages indicate a software system failure. The message may indicate that some part of the infrastructure has become unusable (for example, a database has gone down), or it may indicate a programming error. These errors should be reported to system administrators for analysis. This option is the highest level, providing the smallest amount of messages. Using this setting will send only system error messages to your log file.

### About Station Affinities

You can define affinities to control which workitems are dispatched to a particular SIMULIA Execution Engine station. If a station is to have specific affinity keywords associated with it, you must include the keywords in the `station.properties` file.

Affinities are properties that SIMULIA Execution Engine stations may be declared to have. Individual components in a model may be set to have one or more affinities, so that workitems for running these components may be dispatched only to stations declared to have matching affinities. For example, an Excel component is preset to have the Windows affinity because Excel runs only on Windows; hence, the component will be executed only on a station running on a Windows computer.

Predefined affinities include the station name, OS type, OS name, OS version, and OS architecture. In addition to the predefined affinities, you can set custom and required affinities.

#### Custom Affinities

A SIMULIA Execution Engine administrator can define custom affinities to control workitem dispatches in the local SIMULIA Execution Engine environment. For example, if NASTRAN runs on only one computer, a station running on that computer could be declared to have the affinity “NASTRAN” so that all workitems for NASTRAN jobs would automatically be dispatched there. By default, a station has no custom affinities.

#### Required Affinities

A SIMULIA Execution Engine administrator can define required affinities for stations. For example, you can define a required affinity so that only NASTRAN jobs will be dispatched to the station running on the computer with NASTRAN.

A required affinity becomes a precondition to a workitem to be dispatched to the station. Thus, only components with that affinity will be sent to that station. To define a required affinity in the `station.properties` file, add a “+” before the affinity. A required affinity is indicated by a `+affinity` entry in the properties file.

For example, suppose you have a single SIMULIA Execution Engine station with `APPL_X` installed. If you only define a normal affinity for `APPL_X` on that station, you can then use that affinity in your SIMULIA Execution Engine model to make sure that the component that runs that application—for example, a Simcode component—will get dispatched to that station. This is normal affinity behavior. However, this does not prevent other, unrelated work from also being dispatched to your `APPL_X` station; for example, other task, calculator, or Simcode components. These workitems can deplete resources on this station that you may prefer be dedicated to just running `APPL_X`.

To create a SIMULIA Execution Engine station dedicated solely to `APPL_X` work, you can specify the station affinity as `+APPL_X`. This means that not only will components with this affinity be dispatched to this station, but components that do not have that affinity will be excluded from being dispatched to the station.

You can also give a SIMULIA Execution Engine station more than one required affinity. Therefore, a component must have both of the required affinities to run on that station.

### Station Status Reporting in the Dashboard

The Dashboard shows the current status of any stations connected to the SIMULIA Execution Engine.

The possible status conditions are:
Running: The station is active and either processing workitems or ready to accept new workitems.

Quieting: The shutdown process has been started and no more workitems can be submitted to the station. However, the station may still have workitems left to execute before it can safely shut down—workitems that were submitted but not completed when the shutdown process was started. Depending on the number of pending workitems, the station may stay in this state for a long time.

Shutdown: The station is not accepting new workitems or processing any submitted workitems. It is inactive. Any station that has ever been connected to the SIMULIA Execution Engine remains on this list, so that you can easily see if a station that should still be active has become inactive.

Unknown: A station may be in this state when the connection to the station has been lost or a station is in the process of reconnecting after the SIMULIA Execution Engine has been stopped and restarted. The station will remain in this state until it reconnects with the SIMULIA Execution Engine to report its status or the station is determined to be in the shutdown state.

About Running Multiple Stations on a Single Host Computer

In most cases you should install only one instance of the SIMULIA Execution Engine station on a single computer. You can, however, install and run more than one if desired.

In general, a single SIMULIA Execution Engine station manages all the resources of a single computer, including running multiple simultaneous simulation process flows. The station adjusts its workload level automatically for multiple CPU systems. For information on manually adjusting the concurrency (loading) level of the computer, see Concurrency Limit.

It may be desirable in some cases to run multiple SIMULIA Execution Engine stations on the same physical computer. In this case each station must have a distinct station name. By default, the station name is the host computer name; however, the station name can be specified manually. To manually specify the name, see Station Name.

Unique names are required even if the stations are connecting to different SIMULIA Execution Engine servers. Only one station on a computer can be configured to run as a service/daemon; the others must be started manually.

Starting a SIMULIA Execution Engine Station

For a station to be useful, the SIMULIA Execution Engine must be running on a server on your network, and the SIMULIA Execution Engine library must be preloaded with the basic system metamodels (components).

You can start a station from the command line or from the Start menu on Windows. You can also start the station in console mode, in which you set the connection profile and login information from the command line.

Starting a Station

You can start a SIMULIA Execution Engine station from the command line or from the Start menu in Windows. You must connect the station to a SIMULIA Execution Engine using a connection profile.

1. Start the station using one of the following options:
   - Windows: Click the Start button, point to All Programs / SIMULIA Execution Engine x.x, and click Station.
   - From a command prompt, navigate to the SIMULIA Execution Engine installation directory (if necessary), and execute one of the following commands:
     - Windows: <SEE_install_dir>\os_dir\code\bin\station.exe
     - Linux: <SEE_install_dir>/os_dir/code/bin/station
Note: There are numerous command line options available with the SIMULIA Execution Engine station. For more information, see *Station Properties/Arguments Quick Reference*. You can also view the command line options by running `station -help` at the command line.

The Logon Station dialog box appears.

2. Perform one of the following actions:
   - To connect to a predefined SIMULIA Execution Engine, select the desired profile from the **Connection profile** list.
   - If you want to alter an existing profile, select it from the **Connection profile** list, and click the **Edit** button. The **Profile Editor** dialog box appears, allowing you to change the profile settings.
   - If the connection profile you want to use is not present in the list, you can create one by clicking the **Add** button. The **Profile Editor** dialog box appears.

3. Specify a user ID and password in the corresponding text boxes.
4. Click **OK**.

   The **Station** dialog box appears.

5. Verify that no error messages appear in the **Station Log** area of the interface. If no errors appear, the station is ready to be used by the SIMULIA Execution Engine.

6. Verify that the name of the SIMULIA Execution Engine appears in the lower-right corner of the dialog box. You can click this area to view detailed information about the SIMULIA Execution Engine connection.

**Starting a Station in Console Mode**

You can set up the station to start and log on using console mode. In this mode you set the connection profile and login information from the command prompt.

1. Open a Command Prompt dialog box (terminal window on Linux), and navigate to one of the following directories, based on your operating system:
   - **Windows**: `<SEE_install_dir>`\os_dir\code\command\<br>   - **Linux**: `<SEE_install_dir>`/os_dir/code/command/

2. Type one of the following commands, based on your operating system:
   - **Windows**: `station.bat logonmode:console`<br>     (Be sure not to use `station.exe`)
   - **Linux**: `station logonmode:console`

   If you are starting the station using `telnet` or other environments where there is no display available, add the `nogui:true` option to the command line shown above. This option also suppresses the entire station interface, allowing the station to execute on a system using no display.

   You are prompted for your SIMULIA Execution Engine connection profile.

3. Enter your desired profile. For example, type:

   `seecomputer.cpr`

4. Enter your user name and password.
Important: The password is not hidden. It can be viewed by anyone.

If you used the nogui: true option, a message appears indicating that the station is running. You must leave this Command Prompt dialog box (terminal window) open to keep the station operational. To shut down the station at a later time, use the CTRL+C keyboard command.

Shutting Down a SIMULIA Execution Engine Station
You can manually stop an active station.

1. Display the station dialog box on your screen, if it is not visible.
2. Click Shutdown.
   This action closes the station interface and stops all associated processes.
3. If the station does not shut down in a reasonable amount of time, click the Force button to force it to shut down within a few seconds. This action will prematurely kill any workitems in progress on the station.

Restarting a SIMULIA Execution Engine Station Remotely
Several interfaces give you the ability to stop and restart a station remotely. This feature is useful since it allows you to essentially reboot a station without accessing the computer that is running the station.

This feature does not allow you to restart a station that has been completely stopped (shutdown). It only allows for an immediate stop/restart of the station.

The following restrictions and limitations should be noted when using this feature:

• Restarting a station that was initially launched with an interactive logon will always require manual logon from the computer when the station is running.
• Non-responsive, non-running stations (in the shutdown state) and stations using the LSF DRM option are not supported.

Before remotely restarting a station, you must ensure that the logon settings (user name, password, and connection profile) will be provided using one of the following methods:

• Using station property file settings (as described in Configuring SIMULIA Execution Engine Station Properties)
• Using command line arguments (as described in Configuring SIMULIA Execution Engine Station Properties)
• Specifying information at the prompt when the station is installed as a service. For more information on running the station as a service, see Installing a SIMULIA Execution Engine Station as a Service (Windows) or Installing a SIMULIA Execution Engine Station as a Service Manually (Linux).

You can restart a station in any of the following ways:

• In the Dashboard, right-click on any station in the station list and choose either Restart selected stations or Restart all stations.
• In the Command Line Client, use the restartstation command. See Using the Command Line Client in the Isight User’s Guide for complete information.

Configuring SIMULIA Execution Engine Station Properties
You can customize the operation of a SIMULIA Execution Engine station using the station.properties file or by using the equivalent command line arguments when you start the station.
When a station starts, it reads the `station.properties` file and configures itself according to the property settings. If you include command line arguments when starting the station, the arguments override any settings in the `station.properties` file.

The command line options can be used in any order and in any combination. If you use more than one command line argument, and two are in conflict with each other, the later argument overrides the earlier one.

The `station.properties` file is located in the following directory of the SIMULIA Execution Engine installation:

```
<see_install_dir>/config/station.properties
```

The file can be opened and edited using any text editor. When setting a property, be sure to remove the `#` character at the beginning of the line. All of the properties defined in the `station.properties` file (and their command line equivalents) are optional. All file paths in the `station.properties` file (and the `acs.properties` file) must include the forward slash on both Linux and Windows operating systems.

For a quick reference listing of all station properties and command line arguments, see Station Properties/Arguments Quick Reference. Detailed descriptions of each property and argument are provided in succeeding sections—use the cross-reference links given in the quick reference table for each.

### Station Properties/Arguments Quick Reference

The table below provides a quick reference for the station properties and command line arguments, with links to each individual description. The `station.properties` file also contains information and instructions.

<table>
<thead>
<tr>
<th><code>station.properties</code> entry</th>
<th>Equivalent Command Line Argument</th>
<th>Full Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>fiper.station.affinity</code></td>
<td><code>affinity:&lt;affinity_name&gt;</code></td>
<td>Affinities</td>
</tr>
<tr>
<td><code>fiper.station.allowedusers</code></td>
<td><code>allowedusers:&lt;list_of_users&gt;</code></td>
<td>Allowed Station Users</td>
</tr>
<tr>
<td><code>fiper.station.concurrency</code></td>
<td><code>concurrency:&lt;number&gt;</code></td>
<td>Concurrency Limit</td>
</tr>
<tr>
<td><code>fiper.station.description</code></td>
<td><code>desc:&lt;text_string&gt; or &quot;&lt;text_string&gt;&quot;</code></td>
<td>Station Description</td>
</tr>
<tr>
<td><code>fiper.station.jms.persistent</code></td>
<td><code>not available</code></td>
<td>Persistent JMS Messaging</td>
</tr>
<tr>
<td><code>fiper.station.leaseinterval</code></td>
<td><code>leaseinterval:&lt;minutes&gt;</code></td>
<td>Lease Interval</td>
</tr>
<tr>
<td><code>fiper.stationlogfile</code></td>
<td><code>logfile:&lt;log_file_name&gt;</code></td>
<td>Log File Location</td>
</tr>
<tr>
<td><code>fiper.station.loglevel</code></td>
<td>`loglevel:[debug</td>
<td>info</td>
</tr>
<tr>
<td><code>fiper.station.maxquiescetime</code></td>
<td><code>not available</code></td>
<td>Shutdown Timeout Command</td>
</tr>
<tr>
<td><code>fiper.station.name</code></td>
<td><code>name:&lt;station_name&gt;</code></td>
<td>Station Name</td>
</tr>
<tr>
<td><code>fiper.station.nogui</code></td>
<td>`nogui:[true</td>
<td>false]`</td>
</tr>
<tr>
<td><code>fiper.station.retrydelay</code></td>
<td><code>not available</code></td>
<td>Windows Service Retry Delay</td>
</tr>
<tr>
<td><code>fiper.station.runas</code></td>
<td><code>not available</code></td>
<td>Station-Specific Run-As Behavior</td>
</tr>
<tr>
<td><code>fiper.station.saveLogOnError</code></td>
<td><code>not available</code></td>
<td>LSF Station Error Logs</td>
</tr>
<tr>
<td>station.properties entry</td>
<td>Equivalent Command Line Argument</td>
<td>Full Description</td>
</tr>
<tr>
<td>-------------------------</td>
<td>---------------------------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>fiper.station.serviceconfigfile</td>
<td>serviceconfigfile:&lt;filename&gt;</td>
<td>Windows Service Configuration File</td>
</tr>
<tr>
<td>fiper.station.tempdir</td>
<td>tempdir:&lt;directory&gt;</td>
<td>Temporary Directory</td>
</tr>
<tr>
<td>fiper.station.workdir</td>
<td>not available</td>
<td>Working Directory</td>
</tr>
<tr>
<td>fiper.logon.prompt</td>
<td>logonprompt:[yes</td>
<td>no]</td>
</tr>
<tr>
<td>fiper.logon.profile</td>
<td>profile:&lt;cpr_filename&gt;</td>
<td>Connection Profile</td>
</tr>
<tr>
<td>fiper.logon.prop.user</td>
<td>user:&lt;username&gt;</td>
<td>User Name</td>
</tr>
<tr>
<td>fiper.logon.prop.pw</td>
<td>pw:&lt;password&gt;</td>
<td>Password</td>
</tr>
<tr>
<td>fiper.grid.ssh.configfile</td>
<td>not available</td>
<td>Grid Distributed Execution</td>
</tr>
<tr>
<td>fiper.file.RootA</td>
<td>not available</td>
<td>Shared File System Symbolic Root Directories for File Parameters</td>
</tr>
<tr>
<td>fiper.security.station.domain</td>
<td>not available</td>
<td>Security Domain for Run-As Mode on Windows</td>
</tr>
<tr>
<td>fiper.security.substation.cache.size</td>
<td>not available</td>
<td>Substation Cache Size</td>
</tr>
<tr>
<td>fiper.security.substation.idle.timeout</td>
<td>not available</td>
<td>Idle Substation Timeout</td>
</tr>
<tr>
<td>fiper.security.station.domain</td>
<td>not available</td>
<td>Security Domain for Run-As Mode on Windows</td>
</tr>
<tr>
<td>fiper.security.substation.cache.size</td>
<td>not available</td>
<td>Substation Cache Size</td>
</tr>
<tr>
<td>fiper.security.substation.idle.timeout</td>
<td>not available</td>
<td>Idle Substation Timeout</td>
</tr>
<tr>
<td>not available</td>
<td>configfile:&lt;filename&gt; or @filename</td>
<td>Configuration File of Command Line Arguments</td>
</tr>
<tr>
<td>not available</td>
<td>-help</td>
<td>Command Line Argument Help</td>
</tr>
<tr>
<td>not available</td>
<td>locale:&lt;locale_string&gt;</td>
<td>Locale</td>
</tr>
<tr>
<td>not available</td>
<td>logonmode:[GUI</td>
<td>console]</td>
</tr>
<tr>
<td>not available</td>
<td>service:[stop</td>
<td>start]</td>
</tr>
<tr>
<td>not available</td>
<td>unix_user:&lt;user_id&gt;</td>
<td>Linux User for Extended Grid Credentials</td>
</tr>
<tr>
<td>not available</td>
<td>unix_pw:&lt;password&gt;</td>
<td>Linux Password for Extended Grid Credentials</td>
</tr>
<tr>
<td>not available</td>
<td>win_user:&lt;user_id&gt;</td>
<td>Windows User for Extended Grid Credentials</td>
</tr>
<tr>
<td>not available</td>
<td>win_domain:&lt;domain_name&gt;</td>
<td>Windows Domain for Extended Grid Credentials</td>
</tr>
<tr>
<td>not available</td>
<td>win_pw:&lt;password&gt;</td>
<td>Windows Password for Extended Grid Credentials</td>
</tr>
</tbody>
</table>
Station Name
This property allows you to arbitrarily name a SIMULIA Execution Engine station. By default, the name is taken from the local IP host name.

Every station must have a name so that the SIMULIA Execution Engine can identify all stations connected to it. Station names also allow users to set component affinities to force certain components to run only on certain stations. It is possible to run multiple SIMULIA Execution Engine stations on one computer only if they have unique names.

- **Property name:** `fiper.station.name=<station_name>`
- **Command line argument:** `name:<station_name>`

🌟 **Important:** Do not use this property to change the name of a station when using LSF. You must use the default station name.

Station Description
This property allows you to enter a description of the station, which appears in the Dashboard interface.

- **Property name:** `fiper.station.description=<text_string>` or `"<text_string>"`
- **Command line argument:** `desc:<text_string>` or `"<text_string>"`

If the description contains spaces, you must surround it with quotation marks.

Command line examples:

```
station desc:FiperStation1
station desc:"Station Number One"
```

Allowed Station Users
This property allows you to specify the users that can access the station. The list should consist of user names separated by commas. By default, this property value is blank, allowing all users to access the station.

- **Property name:** `fiper.station.allowedusers=<list_of_users>`
- **Command line argument:** `allowedusers:<list_of_users>`

Command line example:

```
station allowedusers:seadmin,joeuser
```

Affinities
This property is used to specify custom affinities for the station. The value is one or more affinity keywords, separated by spaces.

- **Property name:** `fiper.station.affinity=<affinity_name>`
- **Command line argument:** `affinity:<affinity_name>`

See [About Station Affinities](#) for more information about station affinities.
Property file example:

```
fiper.station.affinity=DAEMON CFD
```

Command line example:

```
station affinity:Word
```

You can use multiple affinity arguments to specify more than one affinity. Alternatively, you can use one argument with the keywords separated by spaces and enclosed in double quotation marks. For example:

```
station affinity:CFD affinity:DOCS
```

or

```
station "affinity:CFD DOCS"
```

**Lease Interval**

Stations are required to periodically report their status to their SIMULIA Execution Engine. This notification process lets the SIMULIA Execution Engine know that the station is still functioning properly, which helps eliminate long delays associated with attempting to contact a nonfunctional station. This property allows you to specify how many minutes elapse before the station reports its status to the SIMULIA Execution Engine. The default setting is 10 minutes. It is highly recommended that you not change this value. You can also perform this operation manually using the **Renew Lease** button on the station interface.

- **Property name:** `fiper.station.leaseinterval=<minutes>`
- **Command line argument:** `leaseinterval:<minutes>`

Command line example:

```
station leaseinterval:30
```

**Default Logging Level**

Stations log various messages, of varying levels of importance, to a scrolling text area on the station interface (when running in GUI mode) and, optionally, to a log file. This property specifies the types of message displayed. Basically, you are selecting the amount of information that you want to appear on the station interface and in the log file.

By default, only warnings, application errors, and internal system errors are reported. When the station interface appears, the **Detail level** list shows the log level specified by this property.

For more information about logging levels, see **About Log Message Detail Levels**.

- **Property name:** `fiper.station.loglevel=[debug|info|warn|error|syserror]`
- **Command line argument:** `loglevel:[debug|info|warn|error|syserror]`

This setting must be a word, not a number. The debug level provides the most information, while the syserror level provides the least information.

Property file example:

```
fiper.station.loglevel=info
```
Command line example:

```bash
station loglevel:error
```

If logging is enabled, you can set the maximum size and number of log files using these properties:

- `fiper.logging.maxSizeKB`
- `fiper.logging.numBackups`

## Log File Location

This property allows you to specify the name and location of the log file that the station will generate. The information in the log file is the same as the information that appears in the station interface.

By default, the station logs are written to the following file:

```
{fiper.station.tempdir}/{fiper.station.name}/station.log
```

and backup files `station1.log`, `station2.log`, etc. This property lets you override the default.

- **Property name**: `fiper.station.logfile=<log_file_name>`
- **Command line argument**: `logfile:<log_file_name>`

Give the full path to the desired directory, plus the file name; for example:

```bash
station logfile:/logs/station_log.txt
```

## Temporary Directory

Stations often write temporary files while executing SIMULIA Execution Engine jobs. This property names the path/directory where the temporary files are written.

- **Property name**: `fiper.station.tempdir=<directory>`
- **Command line argument**: `tempdir:<directory>`

Command line example:

```bash
station tempdir:c:\simulia\Execution Engine\2016\station_tmp\`
```

If you are using the Run-As mode, the default temporary directory should be a directory that is accessible to all users. For example:

- **Windows**: `c:\temp`
- **Linux**: `/var/tmp/`

The directory you specify must be on a disk with sufficient free space to hold all files that will be written by SIMULIA Execution Engine jobs. The default location for this directory is the user’s temporary directory (Windows) or is defined by the `TMPDIR` environment variable (Linux).

By default, the station logs are written to the following file in the temporary directory:

```
{fiper.station.tempdir}/{fiper.station.name}/station.log
```

and backup files `station1.log`, `station2.log`, etc.
Working Directory

The working directory is used by the station to store working files. If you do not specify a working directory, the temporary directory (fiper.station.tempdir) is used instead.

Temporary file systems are often subject to periodic automatic cleanup. This is not appropriate for the directory specified by fiper.station.tempdir, but it is appropriate for fiper.station.workdir. If a station is running in an environment where there is only a little disk space available on non-temporary file systems and a great deal of space available on temporary file systems, fiper.station.tempdir should point to a directory on a non-temporary file system and fiper.station.workdir should point to a directory on a temporary file system.

- **Property name:** fiper.station.workdir=<directory>
- **Command line argument:** not available

Property file example:

```
fiper.station.workdir=/scratch/SEE/stationwork
```

If you are using Run-As mode, the working directory should be accessible to all users. The working directory must be on a disk with sufficient free space to hold all working files that will be written by SIMULIA Execution Engine jobs.

Running a Station Without a GUI

This property controls whether stations are run with or without a graphical user interface (GUI). With a GUI, all log messages are displayed on the GUI, in the dialog box. Without a GUI, the log messages are written to standard output or to the log file if one is named.

When running a station without a GUI on Windows, the station should be started with the command station.bat instead of station.exe. Using the correct file allows the user to be prompted for logon credentials and to see station messages in the command window where the station is started. To shutdown the station, press Ctrl-C in the command window. If station.exe is used to start the station, it is run as a window-less background process, and the user will not be able to supply logon credentials or stop the station.

- **Property name:** fiper.station.nogui=[true|false]
- **Command line argument:** nogui:[true|false]

By default, a station runs in GUI mode.

Logon Mode

This argument allows you to determine if the logon prompt appears in GUI mode (the default) or through a console (command line).

- **Property name:** not available in station.properties file
- **Command line argument:** logonmode:[GUI|console]

This argument is useful for starting a station through a remote session (such as telnet) where there is no GUI capability. When it is used, you will be prompted to log on using prompts in the command line interface.

Command line example:

```
station.bat logonmode:console
```

On Windows, this argument only works with the station.bat executable file.
Logon Prompt
This property allows you to specify whether the Logon dialog box appears when the station is started.

If you use this property to prevent the Logon dialog box from appearing, you must specify a profile name, user name, and password using the corresponding properties or command line arguments—see Logging Into a Station Without Prompt. Failure to specify this information will cause the station to start incorrectly.

- **Property name:** fiper.logon.prompt=[yes|no]
- **Command line argument:** logonprompt:[yes|no]

The default value of this property is yes.

Connection Profile
This property allows you to specify the connection profile (.cpr file) that the station will use when it starts.

- **Property name:** fiper.logon.profile=<cpr_filename>
- **Command line argument:** profile:<cpr_filename>

If you specify only this command line argument, the Logon dialog box will appear and you will have to specify the name and password for the connection profile.

Command line example:
```
station profile:see.cpr
```

If you receive an error that the connection profile cannot be found, use the full path to the .cpr file. For example:
```
station profile:c:\simulia\Execution Engine\2016\see.cpr
```

User Name
This property allows you to specify the user that will log on to the station.

- **Property name:** fiper.logon.prop.user=<username>
- **Command line argument:** user:<username>

If you specify only this argument but not the logonprompt: no option, the Logon dialog box will appear but the user name will already be defined.

Password
This property allows you to specify the password of the user that will log on to the station.

- **Property name:** fiper.logon.prop.pw=<password>
- **Command line argument:** pw:<password>

If you specify only this argument, the Logon dialog box will appear but the password will already be defined.

Command line example:
```
station pw:beatlejuice
```
**Linux User for Extended Grid Credentials**
This argument allows you to specify a Linux user name when logging in using the extended grid credentials for SIMULIA Execution Engine.

- **Property name**: not available in `station.properties` file
- **Command line argument**: `unix_user:<user_id>`

**Linux Password for Extended Grid Credentials**
This argument allows you to specify a Linux password when logging in using the extended grid credentials for SIMULIA Execution Engine.

- **Property name**: not available in `station.properties` file
- **Command line argument**: `unix_pw:<password>`

**Windows User for Extended Grid Credentials**
This argument allows you to specify a Windows user name when logging in using the extended grid credentials for SIMULIA Execution Engine.

- **Property name**: not available in `station.properties` file
- **Command line argument**: `win_user:<user_id>`

**Windows Domain for Extended Grid Credentials**
This argument allows you to specify a Windows domain name when logging in using the extended grid credentials for SIMULIA Execution Engine.

- **Property name**: not available in `station.properties` file
- **Command line argument**: `win_domain:<domain_name>`

**Windows Password for Extended Grid Credentials**
This argument allows you to specify a Windows password when logging in using the extended grid credentials for SIMULIA Execution Engine.

- **Property name**: not available in `station.properties` file
- **Command line argument**: `win_pw:<password>`

**Starting the Station as a Windows Service**
This argument allows you to start the station as a service on Windows operating systems, if the station was installed as a service.

- **Property name**: not available in `station.properties` file
- **Command line argument**: `service:[stop|start]`

For more information about this type of installation, see [Installing a SIMULIA Execution Engine Station as a Service](#) (for Windows stations) and [Installing a SIMULIA Execution Engine Station as a Service Manually](#) (for Linux stations).
Command line example:

```
station service:start
```

**Windows Service Configuration File**

This property specifies the name of the service configuration file when running as a service on Windows. Do not change this setting manually. It is set by the station installer (`installstation.bat`).

- **Property name**: `fiper.station.serviceconfigfile`
- **Command line argument**: `serviceconfigfile:<service_config_filename>

For more information on this type of installation, see *Installing a SIMULIA Execution Engine Station as a Service*.

**Windows Service Retry Delay**

If a station is running as a service on Windows, you can use this setting to determine how often the station should attempt to reconnect to the SIMULIA Execution Engine if the connection is broken.

The time is set in seconds.

- **Property name**: `fiper.station.retrydelay=<seconds>`
- **Command line argument**: not available

Property file example:

```
fiper.station.retrydelay=60
```

**LSF Station Error Logs**

This property determines whether to save the station LSF log file when an error occurs, if you are using the LSF distributed resource management (DRM) mode.

- **Property name**: `fiper.station.saveLogOnError`
- **Command line argument**: not available

This property can be set to `true` or `false` (the default).

Property file example:

```
fiper.station.saveLogOnError=false
```

If this property is set to `true`, the station LSF log information is saved in the following file in the temporary directory when an error occurs:

```
{fiper.station.tempdir}/station_{LSF-job-ID}.log
```

**Grid Distributed Execution**

You can use this property to configure remote copy and remote run commands for distributed execution. See the `station.properties` file for instructions.

- **Property name**: `fiper.grid.ssh.configfile`
- **Command line argument**: not available
**Concurrency Limit**

This property allows you to specify the maximum number of execution threads for the station.

A station can run only a limited number of workitems concurrently. Once that limit is reached, further workitems received must be queued until an active one terminates. Limiting concurrency prevents the station application from overloading the computer it is running on.

By default, the concurrency is 2 times the number of processors on the computer. It is recommended that you do not change this setting. For more information, contact SIMULIA Technical Support.

- **Property name**: fiper.station.concurrency
- **Command line argument**: concurrency:<number>

**Shared File System Symbolic Root Directories for File Parameters**

These properties allow you to adjust how the SIMULIA Execution Engine references a shared file system for Isight file parameters.

- **Property names**:
  - fiper.file.root.RootA
  - fiper.file.root.RootB
  - fiper.file.root.RootC
  etc.

- **Command line argument**: not available

Property file example:

```
fiper.file.root.RootA=/net/host/home/RootA
```

When this directory path is defined in the station.properties file, you can create an Isight file parameter with a path similar to

```
{rootA rootA/path/filename.ext}
```

In this example the instance of RootA would be replaced with the actual path for this file system.

Shared or network file systems are often named differently on different computers. For example, a user’s home directory might be /home/user on the user’s Linux workstation and /net/host/user on other Linux workstations. At the same time, it might be mounted as H: on the user’s Windows workstation and be available as //host/user on other Windows computers.

The shared file system feature of SIMULIA Execution Engine allows it to adjust how it references a shared file to account for these differences. Instead of using an absolute path, which will be incorrect on some computers, the file is referenced as a path relative to a symbolic root directory. On each computer, the symbolic root directory is set to the location where that computer mounts the shared file system. Each time the file is referenced—by the Design Gateway, Runtime Gateway, or a station—the local symbolic root value is used to build the absolute path to the file that is appropriate for that computer.

**Note**: The names of all physical directories must be written using the forward slash (/) as a path separator, even on Windows operating systems.

Symbolic roots are used through the Design Gateway Files tab and defined using the preferences options.
The following examples could be defined on Windows:

```
fiper.file.root.CFDfiles=//server1/CDFfiles
fiper.file.root.docs=D:/Documents
```

For details about file parameters, see Configuring a File as the Source or Destination in the Isight User’s Guide.

**Station-Specific Run-As Behavior**

By default, the SIMULIA Execution Engine station Run-As behavior follows the Run-As configuration of the SIMULIA Execution Engine to which the station is connected. This property can be used to force the station to run with Run-As disabled even if the SIMULIA Execution Engine has Run-As enabled. It can also be used to force the station to run with Run-As enabled, although the station will not run unless the SIMULIA Execution Engine has the Run-As feature enabled.

- **Property name**: fiper.station.runas
- **Command line argument**: not available

Property file example:

```
fiper.station.runas=disabled
```

This property can have any of the following values:

- **disabled**: This option turns off the Run-As feature for the station. It is valid only if the Run-As feature is active on the SIMULIA Execution Engine that the station is using.
- **enabled**: This option turns on the Run-As feature for the station. It is valid only if the Run-As feature is active on the SIMULIA Execution Engine that the station is using.
- **unsecured**: On Linux only, this option allows you to run a station in a Run-As environment without specifying a password.

**Security Domain for Run-As Mode on Windows**

The security domain controls authentication of user credentials on the station when the station security feature (Run-As) is enabled. By default, the domain configured in the SIMULIA Execution Engine is used.

- **Property name**: fiper.security.station.domain
- **Command line argument**: not available

This property applies only when Run-As mode is enabled for the station.

This property applies only on Windows, and is ignored on Linux. On Linux, stations always authenticate against the domain configured by the system administrator.

**Substation Cache Size**

This property allows you to specify the maximum number of substation processes to be cached.

If the number of substation processes exceeds the substation cache size limit, the unused processes are shut down.

By default, the cache size is 2 times the concurrency limit (see *Concurrency Limit*).

- **Property name**: fiper.security.substation.cache.size
- **Command line argument**: not available
Idle Substation Timeout
This property allows you to specify the idle timeout period (in minutes) for substation processes. If a substation process is idle for longer than the timeout period, the process is shut down.
By default, the timeout is 5 minutes.

- **Property name**: fiper.security.substation.idle.timeout
- **Command line argument**: not available

Shutdown Timeout Command
The station defers shutdown until all work on the station has completed. By default, the station waits indefinitely or until a forced shutdown occurs. Editing this property will cause all work that is active when the timeout occurs to be abandoned and marked as failed. Changing this setting may also cause subsequent jobs to fail. It is recommended that you do not change this setting.

- **Property name**: fiper.station.maxquiescetime
- **Command line argument**: not available

Command Line Argument Help
You can use `-help` on the command line to show a dialog box that describes all of the command line arguments available. You can also use the following equivalent arguments to display this dialog box: `-h`, `?`, or `/?`.

- **Property name**: not available in station.properties file
- **Command line argument**: `-help`

Example:

```
station -help
```

Configuration File of Command Line Arguments
This argument allows you to read a set of other command line arguments from a specified file. The file must be a text file containing command line arguments as they would normally appear on a command line.

- **Property name**: not available in station.properties file
- **Command line argument**: `configfile:<filename>`
  or `@filename`

You can use either of the two forms of this argument; they are interchangeable.

Command line examples:

```
station configfile:myargs.txt
station @see.txt
```

For example, if you have a file `see.txt` in your home directory that contains the following entries:

```
profile:ExecutionEngine1 logonprompt:no
# provide user name and password to connect
user:seeadmin pw:seeadmin
```
You can now start a station connected to ExecutionEngine1 with the following command:

```
station @see.txt
```

You can also mix and match the contents of the file with the actual command line option at the command prompt. For example:

```
station @see.txt logonprompt:yes
```

For the example file see.txt shown above, this argument would show the Logon dialog box (logonprompt option) even though the file provided everything necessary and also instructed the Logon dialog box to not appear.

The arguments file you create can contain one or more lines. Each line can contain one or more command line arguments. Arguments with spaces must be enclosed in single or double quotation marks. Comments lines are allowed—any line starting with # is ignored. Empty lines are also ignored.

**Locale**

This argument sets the locale (language environment) for the current session. It is useful only when testing support for a language. You can also use the shorter version of this argument: -l.

- **Property name**: not available in station.properties file
- **Command line argument**: locale:<locale_string>

Command line examples to set the locale to German:

```
station locale:de_DE
station -l de_DE
```

**Persistent JMS Messaging**

Persistent JMS messaging allows the SIMULIA Execution Engine to recover running jobs from an unexpected server shutdown due to power loss, network connectivity interruption, etc.

- **Property name**: fiper.station.jms.persistent
- **Command line argument**: not available

Property file example:

```
fiper.station.jms.persistent=true
```

The SIMULIA Execution Engine often communicates with stations using a Java asynchronous messaging infrastructure known as JMS. By default, this infrastructure is configured to make a best effort at delivering every message to its intended recipients. The default value of true changes the station side of the infrastructure configuration such that it guarantees delivery of every message to its intended recipients. It does this by changing the message consumers to use Durable Subscriptions on the Topics to which the station is listening.

There are also corresponding configuration entries in the acs.properties file:

- **fiper.acs.jms.persistent**. This property must also be left set to true to change the server side of the infrastructure configuration to guarantee delivery. It does this by changing the delivery mode on the message producers to Persistent.
- **fiper.acs.jms.ttl**. This property sets the time-to-live value (in minutes) for persistent messages. The default value for this entry is 0, which means that the messages are kept in the cache until they are successfully delivered to all known recipients.
Logging Into a Station Without Prompt

You can use a set of properties to automate the SIMULIA Execution Engine station connection and login process.

To set up the station.properties file to skip the connection profile (.cpr file) and login screen, edit the following properties to set the values as shown:

```properties
fiper.logon.prompt=no
fiper.logon.profile=<full_path_to_cpr_file>
fiper.logon.prop.user=<user_name>
fiper.logon.prop.pw=<user_password>
```

Example:

```properties
fiper.logon.prompt=no
fiper.logon.profile=c:/simulia/Execution Engine/2016/see.cpr
fiper.logon.prop.user=seeadmin
fiper.logon.prop.pw=seeadmin
```

You can also use the equivalent command line arguments to accomplish the same configuration; for example:

```
station profile:see_system.cpr logonprompt:no user:joeschmoe
pw:joeschmoe
```

⭐ Important: Be sure to use forward slashes (/) in the profile path.

Using the Dashboard

The Dashboard is a program that displays the current status of the SIMULIA Execution Engine. It lists the running stations, number of running jobs, workitems on each station, and licenses being used by the SIMULIA Execution Engine.

To run the Dashboard on a computer other than the one containing the SIMULIA Execution Engine installation, you must install the SIMULIA Execution Engine.

About the Dashboard Interface

The Dashboard interface shows you varied status information about the SIMULIA Execution Engine, the stations, access control, and license usage. The Dashboard allows you to perform various management and monitoring operations on the SIMULIA Execution Engine and the stations.

The Dashboard below shows that two active and one inactive stations are currently associated with this SIMULIA Execution Engine.
The Stations tab of the Dashboard dialog box shows information in the following columns.

- **Station Name.** The name of the SIMULIA Execution Engine station. By default, the station name is the same as the name of the computer running the station. However, you can change the name as described in Station Name.

- **Status.** The current availability of the station, which can be any of the following states: Running, Quieting, Shutdown, or Unknown. See Station Status Reporting in the Dashboard for definitions of these conditions.

- **Description.** A text description of the station, which can be used to more easily identify specific stations. For more information on how to define a station’s description, see Station Description.

- **DRM Mode.** The DRM mode used by the station: either Fiper or LSF.

- **O/S.** The operating system of the computer running the station.

- **Host.** The name of the computer where the station is running.

- **Workload.** How much work the station is currently performing. This measurement is shown with a colored bar.

   **Note:** You can sort the station list by clicking any of the column headings.

The SIMULIA Execution Engine tab of the Dashboard dialog box shows information such as server type, release number, and configuration properties. You can also click the All server properties check box to view additional configuration properties of the SIMULIA Execution Engine.

The Access Control tab of the Dashboard dialog box lets you define and manage user access controls. See Managing Access Control for SIMULIA Execution Engine Users for more details.

The License tab of the Dashboard dialog box shows you how many instances of a particular license feature are currently being used. See Viewing License Usage Information for more details.

### Starting the Dashboard

You can start the Dashboard interface from the command line or from the Start menu in Windows.

1. Start the Dashboard program using one of the following options:
   - **Windows:** Click the Start button, point to All Programs / SIMULIA Execution Engine x.x, and click Dashboard.
From a command prompt, navigate to the SIMULIA Execution Engine installation directory (if necessary), and execute one of the following commands:

- **Windows**: `<SEE_install_dir>\<os_dir>\code\command\dashboard.exe`
- **Linux**: `<SEE_install_dir>/<os_dir>/code/command/dashboard`

The **Logon Dashboard** dialog box appears.

2. Select the desired connection profile from the **Connection profile** list.
3. Specify the user ID and password in the corresponding text boxes, and click **OK**.

   The **Dashboard** dialog box appears, with the **Stations** tab selected.

### Command Line Options for the Dashboard

You can customize the operation of the Dashboard using command line arguments when you start the Dashboard with the `dashboard.exe` (or `dashboard`) command. The command line arguments available are described below.

The arguments can be used in any order and in any combination. If you use more than one command line argument and two are in conflict with each other, the later argument overrides the earlier one.

#### @FILENAME

This argument allows you to read a set of Dashboard command line arguments from a specified file. The file must be a text file containing command line arguments as they would typically appear on a command line.

For example, if you had a file `acs.txt` in your home directory that contained the following entries:

```plaintext
profile:executionengine1 logonprompt:no
# provide user name and password to connect
user:fiperacs pw:fiperacs
```

Then you could start a Dashboard connected to ExecutionEngine1 with the following command:

```
dashboard @acs.txt
```

You can also mix and match the contents of the file with the actual command line option at the command prompt. For example:

```
dashboard @acs.txt logonprompt:yes
```

Using the contents of the file created earlier, this argument would show the **Logon** dialog box (**logonprompt** command) even though the file provided everything necessary and also instructed the **Logon** dialog box to **not** appear.

The file you create can contain one or more lines. Each line can contain one or more command line argument. Arguments with spaces must be enclosed in single or double quotation marks. Comments lines are allowed (any line starting with `#` is ignored). Empty lines are also ignored.

This argument is interchangeable with the `configfile` argument.
-help

This argument opens a dialog box that displays all the arguments discussed in this section. You can also use the following arguments to open this dialog box: -h, ?, or /?

Example: dashboard -help

leaseinterval

leaseinterval: <seconds>

This argument allows you to specify how many minutes elapse before the Dashboard renews its status with the SIMULIA Execution Engine. For more information, see Lease Interval.

Example: dashboard leaseinterval: 30

locale

locale: <locale_string>

This argument sets the locale (language environment) for the current session. It is useful only when testing support for a language. You can also use the shorter version of this argument: -l.

Example (German): dashboard locale: de_DE
Example (German): dashboard -l de_DE

logfile

logfile: <log_file_name>

This argument allows you to specify the location and name of the log file that the Dashboard will generate.

Example: dashboard logfile: \temp\dashboard_log.txt

loglevel

loglevel: [debug|info|warn|error|syserror]

This argument allows you to specify the log level of the Dashboard log file. In essence, you are selecting the amount of information that you want to appear in the log file. The debug level shows the most information while the syserror shows the least. For more information on these log levels, see About Log Message Detail Levels.

Example: dashboard loglevel: debug

logonprompt

logonprompt: [yes|no]

This argument allows you to specify if the Logon dialog box appears when the Dashboard is started. If you use this argument to stop the Logon dialog box from appearing (the no option), you must specify a profile name, user name, and password using the arguments in this list. If you do not specify this information, the Dashboard will not start properly.
Example: dashboard profile:acs_system.cpr logonprompt:no user:fiperacs pw:fiperacs

**profile**

profile:<connection_profile_filename>

This argument allows you to specify the connection profile that the Dashboard will use. If you specify only this argument, the Logon dialog box will appear, and you will have to specify the name and password for the connection profile.

Example: dashboard profile:acscomputer.cpr

*Caution:* If you receive an error that the connection profile cannot be found, use the full path to the profile file. For example:

dashboard profile:c:\simulia\Execution Engine\2016\acs_system.cpr

**pw**

pw:<password>

This argument allows you to specify the password of the user that will log on to the Dashboard. If you specify only this argument, the Logon dialog box will appear but the password will already be defined.

Example: dashboard pw:fiperacs

**unix_pw**

unix_pw:<password>

This argument allows you to specify a Linux password when logging in using the SIMULIA Execution Engine extended grid credentials.

**unix_user**

unix_user:<user_id>

This argument allows you to specify a Linux user name when logging in using the SIMULIA Execution Engine extended grid credentials.

**user**

user:<user_id>

This argument allows you to specify the user that will log on to the Dashboard. If you specify only this argument but not the logonprompt:no option, the Logon dialog box will appear but the user name will already be defined.

Example: dashboard user:fiperacs
win_domain

win_domain:<domain_name>

This argument allows you to specify a Windows domain name when logging in using the SIMULIA Execution Engine extended grid credentials.

win_pw

win_pw:<password>

This argument allows you to specify a Windows password when logging in using the SIMULIA Execution Engine extended grid credentials.

win_user

win_user:<user_id>

This argument allows you to specify a Windows user name when logging in using the SIMULIA Execution Engine extended grid credentials.

Viewing Connection Information

You can use the Dashboard to see network connection information for the SIMULIA Execution Engine as well as any connected stations.

1. Open the Dashboard dialog box as described in Starting the Dashboard.
2. Click the button in the lower-right corner that shows the name of the SIMULIA Execution Engine (and the current user name if security is enabled).

   Click the button to refresh the information displayed on the Dashboard. The refresh is also done automatically every few seconds.

Viewing Station Information

You can use the Dashboard to view details about any stations connected to the SIMULIA Execution Engine.

Before you begin: You must have the fiiperadmin security role to view station information.

1. Open the Dashboard dialog box as described in Starting the Dashboard.
2. Click a SIMULIA Execution Engine station name in the list near the top of the Stations tab.

   The Station Details area in the bottom area of the tab is populated with information about the selected SIMULIA Execution Engine station.

Controlling Station Workitems

You can use the Dashboard to stop, pause, or resume the execution of any workitems on a station.

Before you begin: You must have the fiiperadmin security role to perform these operations.

1. Open the Dashboard dialog box as described in Starting the Dashboard.
2. Click a SIMULIA Execution Engine station name in the list near the top of the Stations tab.
3. If any workitems are listed, you can stop, pause, or resume the associated job using the corresponding buttons.
You can also view the details of a workitem or select and cancel one or more workitems (for example, a workitem taking too long to finish or one that appears unable to finish).

**Shutting Down, Restarting, or Deleting a Station**

You can shut down, restart, or delete a station from the Dashboard, using the right-click (context) menu.

**Before you begin:** You must have the *fiperadmin* security role to perform these operations.

1. Open the Dashboard dialog box as described in *Starting the Dashboard*.
2. Right-click the SIMULIA Execution Engine station name in the list near the top of the Stations tab. The following commands are available:
   
   - **Shutdown selected station(s).** This option shuts down the selected stations.
   - **Shutdown all stations.** This option shuts down all of the listed stations that are not already in the shutdown state.
   - **Restart selected station(s).** This option allows you to stop and restart selected stations. For more information on using this option, including its limitations, see *Restarting a SIMULIA Execution Engine Station Remotely*.
   - **Restart all stations.** This option allows you to stop and restart all stations that can use this option. For more information on using this option, including its limitations, see *Restarting a SIMULIA Execution Engine Station Remotely*.
   - **Delete selected station(s).** This option allows you to delete inactive (shutdown) stations that you no longer want included in the station list.

Use these commands with care as shutting down a station that is actively running workitems can result in lost work.

**Important:** The restart options cannot be used to simply restart stopped stations from the Dashboard. These options essentially execute a remote station reboot—i.e., stopping a running station and then restarting it remotely.

**Managing Access Control for SIMULIA Execution Engine Users**

You can use the Dashboard to define and manage user access controls.

**Before you begin:** To perform these operations, you must have the *fiperadmin* security role or you must have been added to the system administrator list. For information on how to add additional users as system administrators, see *Managing Access Control – System Administration*.

1. Open the Dashboard dialog box as described in *Starting the Dashboard*.
2. Click the Access Control tab.
3. Use the buttons on the Access Control tab to define access information. You can add, remove, rearrange, or clear the information on a specific subtab or on all the subtabs.

   These controls are applied to each specified user for all objects that are access-controlled, as opposed to published object and job controls, which are applied to each specified user for each controlled object individually.

   The purpose of each subtab follows from the procedure for calculating the permission assigned to a given user on a given controlled object. You should order the access control assignments as follows:

1. Assignments to user U in the System Administration list, if any
2. Assignments to user U in the System Override list, if any
3. Assignments to user U in the list for object X, if any
4. The default assignment for all users for object X, if any
5. Assignments to user U in the System Default list, if any
6. The default assignment for all users in the System Default list

Note: If there is more than one assignment to user U in any list, the one closest to the top of the list is used; this can occur implicitly when Groups are defined.

The access control for user U on object X is whichever assignment can be found closest to the top of this ordering.

Important: User names are case-sensitive.

Managing Access Control – System Administration

You use the Dashboard to fix access rights for specified users.

Before you begin: To perform these operations, you must have the fiperadmin security role or you must have been added to the system administrator list.

1. Open the Dashboard dialog box as described in Starting the Dashboard.
2. Click the Access Control tab, then the System Administration subtab. This subtab is used to fix access rights for specified users, rights that cannot be altered by assigning rights to those users for specific objects.

For example, a user assigned the ALTER permission in either table would have unrestricted access to every published object and every job. A user assigned the REFERENCE permission would be able to use any published model but not be able to see inside them. Finally, a user assigned the NONE permission would be locked out of the SIMULIA Execution Engine.

Note: If you want to allow additional users (other than the system administrator defined during the initial SIMULIA Execution Engine installation) access to the Dashboard’s Access Control tab, add the user names to the System Administration subtab. The next time the users open the Dashboard, they will see the Access Control tab. Only users added with the ALTER permission will be able to further edit this list of users.

The System Administration subtab has one extra function. Any user with entries in this tab (and only users with this characteristic) is granted access to these tabs when accessing the Dashboard.

3. If desired, you can use any of the following subtabs:
   • System Override. This subtab works in conjunction with the System Administration subtab to fix access rights for specified users.
   • System Default. This subtab is used to assign “fallback” access rights (rights that users will have on specific objects unless they are assigned rights specifically for those objects). These rights include the global default rights which all users will have in the absence of any user-specific rights assignments.
   • SIMULIA Execution Engine Groups. This subtab allows SIMULIA Execution Engine system administrators to collect a set of user names into a single named entity, so that access rights may be assigned jointly to all users in the group. For more information on setting group options for models, see Setting Default Permissions in the Isight User’s Guide.
Viewing License Usage Information

You can use the Dashboard to show you how many instances of a particular license feature are currently being used.

Before you begin: You must have the fiperadmin security role to perform these operations.

1. Open the Dashboard dialog box as described in Starting the Dashboard.
2. Click the License tab.
   
   You can see how many instances of a particular license feature are currently being used by all users using the same license or license server as the SIMULIA Execution Engine. Therefore, the information shown may not be specific to the SIMULIA Execution Engine you are currently using. If more than one SIMULIA Execution Engine is using the same license server, the license information returned will be for both SIMULIA Execution Engines.

3. Select an item from the License feature list.
   
   Information about that license feature is displayed in the area below the list. This information is current immediately after the license feature is selected from the list. However, the information may change at any time. Click Refresh to verify that you are looking at the most up-to-date information. This option is especially useful if you leave the Dashboard running while performing some other task, and then return to it after a period of time.

Using the Command Line Client

The Command Line Client is a console (character mode) program that provides simple text-based access to most functions of the SIMULIA Execution Engine. To run the Command Line Client on a computer other than the one containing the SIMULIA Execution Engine installation, you must install the SIMULIA Execution Engine software.

The command-line client can be run in single-command mode or in prompting mode. In single-command mode a single command (with arguments) is supplied. The requested command is run, any output is displayed, and the client terminates and returns control to the shell. In prompting mode the client acts like a command shell itself, prompting for commands and only terminating when the quit command is executed.

For complete information about the options available when using the Command Line Client, see Using the Command Line Client in the Isight User’s Guide

Starting the Command Line Client

You can start the Command Line Client from the command line or from the Start menu in Windows.

1. Start the Command Line Client using one of the following options:
   
   - **Windows**: Click the Start button, point to All Programs / SIMULIA Execution Engine xx, and click Command Line.
   - From a command prompt, navigate to the SIMULIA Execution Engine installation directory, and execute one of the following commands:
     - **Windows**: `<SEE_install_dir>\<os_dir>\code\command\fipercmd.bat`  
     - **Linux**: `<SEE_install_dir>/os_dir/code/command/fipercmd`
   
   The Logon dialog box appears.
2. To log on to the SIMULIA Execution Engine, first do one of the following:
   - To connect to a predefined SIMULIA Execution Engine, select the desired profile from the **Connection profile** list.
   - If you want to alter an existing profile, select it from the **Connection profile** list, and click the button. The **Profile Editor** dialog box appears, allowing you to change the profile settings.
   - If the connection profile you want to use is not present in the list, you can create one by clicking the button. The **Profile Editor** dialog box appears.

   **Note:** You can set up the Command Line Client to allow you to log in using the console mode. In this mode, you set the connection profile and login information from the command prompt. For more information on this option, see **Using the Command Line Client** in the Isight User’s Guide.

3. Specify a user ID and password in the corresponding text boxes.

4. Click **OK**.
   
   The Command Line Client is ready when the > prompt appears.

   There are numerous commands available for this interface. For example, to display a list of all jobs in the SIMULIA Execution Engine, use the following command:

   ```
   jobstatus
   ```

   For more information on the other options available when using the Command Line Client, see **Using the Command Line Client** in the Isight User’s Guide.

   **Note:** The typical command-shell editing keys can be used in the Command Line Client: up/down arrows to recall previous commands, right/left arrows to edit a command.

5. Type **quit** to exit the Command Line Client.
Generating Reports of SIMULIA Execution Engine License Usage

You can use the licusage utility to generate reports of SIMULIA Execution Engine license usage history.

About License Usage Reports

You can use the licusage utility to generate reports of SIMULIA Execution Engine license usage history. The report tool reads data from the Dassault Systèmes license server log file or the FLEXnet debug log file and customizes the report according to your choice. The example below shows how to enter the options. You must enter -- before each option. For information about each option, see License Report Utility Options.


In the context of this report tool, a license session is defined as a licensed job executing on a SIMULIA Execution Engine product feature, which has a checkout time and a checkin time.

Running the licusage Utility

You can use the licusage utility to generate reports of SIMULIA Execution Engine usage history.

1. To start using the licusage utility, navigate to the following directory:
   
   `<SEE_install_directory>/<operating_system>/code/command/` where the `<operating_system>` is one of the following:

<table>
<thead>
<tr>
<th><code>&lt;operating_system&gt;</code></th>
<th>For</th>
</tr>
</thead>
<tbody>
<tr>
<td>win_b64</td>
<td>64-bit Windows</td>
</tr>
<tr>
<td>linux_a64</td>
<td>64-bit Linux</td>
</tr>
</tbody>
</table>

2. At a command line, type licusage, then enter the options that you want to use.
   
   For information about the available options, see License Report Utility Options.

License Report Utility Options

This section provides reference information on the options that you can use to control the output from the SIMULIA Execution Engine license usage utility.
General Options
You can use the following command line options to generate reports of SIMULIA Execution Engine license usage history.

--log
This option specifies the path and the file name of the log files to be read.

The default location of a DSLS log file is \ProgramData\DassaultSystemes\LicenseServer\LogFiles\ on Windows or /var/DassaultSystemes/LicenseServer/LogFiles/ on Linux. For FLEXnet, you must specify the debug log file. If the log file is located on a remote server, you must copy it to a directory that is accessible by your local computer. This option is required. Multiple log files can be specified in two ways:

- Use the --log option multiple times: --log first.log --log second.log
- Give a comma-separated list of the file names: --log first.log,second.log

--logtype
This option specifies the type of license server being used with SIMULIA Execution Engine: dsls or flexnet. This option is required and is case-sensitive (must be lowercase).

--start
This option specifies the starting date and time for the reporting period.

If the --start option is omitted, the reporting period begins with the oldest recorded item in the log files. The date and time must be specified in one of the following formats:

- \d\-\m\m\-\y\y\y\y \h\h:\m\m:\s\s
  For example, start 01-Jan-2013 09:00:00 indicates a starting time of 9:00 AM on January 1, 2013. Specifying a time is required, including hours, minutes, and seconds. The hour field (\h\h) must be entered in 24-hour format.
- --xhours or --xdays
  Use this format to pick a time or day in the past. For example, --start --6hours specifies a start time of six hours ago.

--end
If the --end option is not specified, the reporting period ends with the most recent recorded item in the log files. The date and time must be specified in one of the following formats:

- \d\-\m\m\-\y\y\y\y \h\h:\m\m:\s\s
  For example, --end 31-Jan-2013 18:00:00 indicates an ending time of 6:00 PM on January 31, 2013. Specifying a time is required, including hours, minutes, and seconds. The hour field (\h\h) must be entered in 24-hour format.
- --xhours or --xdays
  Use this format to pick a time or day in the past. For example, --end --3hours specifies an ending time of three hours ago.
--type
This option specifies the type of report desired: export or query.

If this option is omitted, the default is export. An export report provides the basic licensing session information, usually in tabular format or comma-separated values (CSV). The data can be optionally filtered or sorted. A query report lets you look at sessions in aggregate; you can operate on the raw data to calculate such things as maximum usage, peak usage, and averages. A query report divides the total time period into equal sized buckets. You choose the size of each time bucket with the --duration option. You can also use an export report to save the raw data to a CSV file, import it into an Excel spreadsheet, and perform custom calculations using your own tools.

--accessor
This option specifies a comma-separated list of accessors to be read from the log data.

Accessors are the fields or columns of the output report. For example: --accessor username,checkout,duration. This option is required.

--list_accessors
This option specifies a list of the available report accessors.

The basic accessors are as follows:

- duration—duration of the license session, in seconds
- username—username
- checkout—checkout date
- feature—Dassault Systèmes license feature (trigram); for example, IGF
- quantity—number of licenses requested
- project—custom project names or numbers recorded from the lmprojectenvironment file parameter
- session—the internal session object that represents the licensing job; this is useful only when using the peak aggregator or designing your own custom aggregators

--filter
This option filters the report data.

The required format for this option is as follows:

--filter accessor:value

The report output is filtered to include only data records for which accessor=value. You can include multiple accessor:value pairs in a comma-separated list; for example,

--filter username:tsmith,hostname:zulu

This example would produce a report showing license checkouts only from the user tsmith on the computer zulu. To create other filters, use the --custom option with a Python program.

--list_filters
This option provides a list of the available filters.

The one built-in filter takes the form --filter accessor:value. If you create any custom filters using the -custom option, they will appear in this list.
Export Options

You can use the following command line options to sort the output of the licusage utility.

--sort
This option sorts the data chronologically, from oldest to newest. The required format for this option is --sort date.

--list_sorters
This option lists the available sorters.
The one built-in sorter is date. If you create any custom sorters using the --custom option, they will appear in this list.

Query Options

You can use the following command line options for query reports from the licusage utility.

--aggregator
This option specifies a comma-separated list of aggregator functions to be applied to the accessor values in each time bucket. For example, --aggregator max,average. This option is required if --type query is used.

--list_aggregators
This option obtains a list of the available aggregator functions. The basic aggregators are as follows:

• max—maximum value of an accessor in each time bucket
• min—minimum value of an accessor in each time bucket
• sum—sum of all accessor values in each time bucket
• average—average value of accessor in each time bucket
• len—number of items in each time bucket
• peak—maximum number of licenses in use; must operate on the session accessor

Not all aggregators will work with all assessors. For example, you can successfully calculate the maximum duration of sessions, but attempting to find the average value of usernames is meaningless.

--duration
This option specifies the size of each time bucket for query reports, in minutes, hours, days, weeks, or months. If this option is omitted, the default is 24 hours. Examples: --duration 1hour, --duration 2days

Output Options

You can use the following command line options to format the output of the licusage utility.

--output
This option specifies the path and file name of the file to which the report will be written. If this option is omitted or --output is used, the default is to write to standard output (showing the report in your command prompt or shell window).

--format
This option specifies the output format to use.
**--list_formats**

This option obtains a list of the available output formats. The basic formats are as follows:

- **table**—a plain table with columns for each accessor and/or aggregator selected
- **csv**—comma-separated values

**Advanced Options**

You can use the following command line option to provide custom definitions in the **licusage** utility:

**--custom**

This option specifies a Python program file containing your custom definitions.

**Examples**

This section contains examples of how to use the **licusage** utility to view license usage information.

The following example generates a simple export type of report showing checkout date, username, product feature, and quantity.

```
licusage --log today.log --logtype dsls --type export --accessor checkout,username,feature,quantity
```

<table>
<thead>
<tr>
<th>Checkout Date</th>
<th>Username</th>
<th>Feature</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>2012-Sep-07 12:03:40</td>
<td>bbagins</td>
<td>IGF</td>
<td>10</td>
</tr>
<tr>
<td>2012-Sep-07 12:03:52</td>
<td>gandalf</td>
<td>IGF</td>
<td>20</td>
</tr>
<tr>
<td>2012-Sep-07 12:03:53</td>
<td>bbagins</td>
<td>IGF</td>
<td>20</td>
</tr>
<tr>
<td>2012-Sep-07 12:03:56</td>
<td>gollum</td>
<td>IGD</td>
<td>1</td>
</tr>
</tbody>
</table>

The following example generates a query report that calculates the average number of license tokens used in each 1-hour time bucket.

```
licusage --log today.log --logtype dsls --type query --accessor quantity --aggregator average --duration 1hour
```

<table>
<thead>
<tr>
<th>Bucket</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>2012-Dec-06 12:37:53</td>
<td>25</td>
</tr>
<tr>
<td>2012-Sep-06 13:37:53</td>
<td>19</td>
</tr>
<tr>
<td>2012-Sep-06 14:37:53</td>
<td>8</td>
</tr>
<tr>
<td>2012-Sep-06 15:37:53</td>
<td>18</td>
</tr>
<tr>
<td>2012-Sep-06 16:37:53</td>
<td>13</td>
</tr>
<tr>
<td>2012-Sep-06 17:37:53</td>
<td>34</td>
</tr>
</tbody>
</table>

The example below adds to the query report to calculate the total number of checkouts, the average checkout quantity, and the maximum checkout quantity for all sessions in **today.log** grouped into 1-hour buckets.

```
licusage --log today.log --logtype dsls --type query --accessor quantity,quantity,quantity --aggregator count,average,max --duration 1hour
```

<table>
<thead>
<tr>
<th>Bucket</th>
<th>Count</th>
<th>Average</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>2012-Dec-06 12:37:53</td>
<td>25</td>
<td>25</td>
<td>25</td>
</tr>
<tr>
<td>2012-Sep-06 13:37:53</td>
<td>19</td>
<td>19</td>
<td>19</td>
</tr>
<tr>
<td>2012-Sep-06 14:37:53</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>2012-Sep-06 15:37:53</td>
<td>18</td>
<td>18</td>
<td>18</td>
</tr>
<tr>
<td>2012-Sep-06 16:37:53</td>
<td>13</td>
<td>13</td>
<td>13</td>
</tr>
<tr>
<td>2012-Sep-06 17:37:53</td>
<td>34</td>
<td>34</td>
<td>34</td>
</tr>
</tbody>
</table>
The example below adds one or more columns to the query report to show the maximum duration of the sessions in each bucket.

```
licusage --log today.log dsls --type query --accessor quantity,quantity,quantity,duration --aggregator count,average,max,max --duration 1hour
```

<table>
<thead>
<tr>
<th>Bucket</th>
<th>Number of Items</th>
<th>Average</th>
<th>Maximum</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>2012-Sep-19 12:37:53</td>
<td>108</td>
<td>25</td>
<td>80</td>
<td>1497</td>
</tr>
<tr>
<td>2012-Sep-19 13:37:53</td>
<td>98</td>
<td>19</td>
<td>80</td>
<td>1220</td>
</tr>
<tr>
<td>2012-Sep-19 14:37:53</td>
<td>46</td>
<td>46</td>
<td>66</td>
<td>77</td>
</tr>
<tr>
<td>2012-Sep-19 15:37:53</td>
<td>114</td>
<td>114</td>
<td>50</td>
<td>482</td>
</tr>
<tr>
<td>2012-Sep-19 16:37:53</td>
<td>129</td>
<td>129</td>
<td>50</td>
<td>449</td>
</tr>
<tr>
<td>2012-Sep-19 17:37:53</td>
<td>74</td>
<td>74</td>
<td>66</td>
<td>3552</td>
</tr>
<tr>
<td>2012-Sep-19 09:37:53</td>
<td>2</td>
<td>50</td>
<td>50</td>
<td>3272</td>
</tr>
</tbody>
</table>
Basic Troubleshooting

This section describes the log files that the SIMULIA Execution Engine provides to assist you with resolving errors that occur during installation and usage. It also includes basic troubleshooting information.

User Login Names Containing Punctuation

SIMULIA Execution Engine interfaces do not run correctly if they are started using a user login name (user ID) that contains punctuation marks—most notably !, #, ?, and : (exclamation point, pound sign, question mark, and colon). Because JAVA uses URLs internally to locate JAR files, these characters cause the URL to be misinterpreted. The problem is most severe on Windows-based systems, where the default temporary directory is inside a directory named after the user name.

To avoid this problem, you must force the SIMULIA Execution Engine installation and all temporary files into directories that do not contain these characters. First, verify that the SIMULIA Execution Engine is not installed in a directory that contains any of these characters. Second, manually set your temporary directory to a directory that does not contain these characters. On Windows, set the TEMP environment variable. On Linux, set the TMPDIR environment variable. For more information on setting environment variables on your system, contact your local system administrator.

If the appropriate environment variable cannot be set, use the variable FIPER_TEMP.

Finally, if you cannot set either the system environment variables or the FIPER_TEMP variable, create a file called fiper.bat (Windows) or .fiper.sh (Linux) in your home directory. Use this file to set the environment variable FIPER_TEMP to a “safe” directory name. The contents of the new file should be similar to the following examples:

- Windows (fiper.bat)
  
  ```
  set FIPER_TEMP=C:\TEMP\bang-user
  ```

- Linux (.fiper.sh)
  
  ```
  FIPER_TEMP=/var/tmp/bang-user
  export FIPER_TEMP
  ```

Log Files for the SIMULIA Execution Engine

Log files may be useful when attempting to determine certain issues with the SIMULIA Execution Engine system.

The following log files are generated either by the SIMULIA Execution Engine or TomEE:

- **station.log**: This SIMULIA Execution Engine-generated file matches the information displayed on the SIMULIA Execution Engine station interface, including connection information and execution details. It is located in the temporary directory specified in the station.properties file in the top level of the SIMULIA Execution Engine installation directly. A subdirectory based on the SIMULIA Execution Engine station name is created in this temporary directory. This subdirectory contains the log file.

  The default location for this temporary directory is as follows:
Configuring the Windows Firewall

To ensure that your SIMULIA Execution Engine will function correctly and be able to communicate with other computers in your network, you need to configure the Windows Firewall for specific port exceptions. You only need to perform this action on the computer running the SIMULIA Execution Engine. You do not need to alter the Windows Firewall settings on systems running a SIMULIA Execution Engine station or Isight, unless these computers are also acting as a license server.

You may need to alter the Windows Firewall settings as described in Configuring the Windows Firewall for TomEE.

Configuring the Windows Firewall for TomEE

You need to open several ports to ensure that your Tomee-based SIMULIA Execution Engine will function correctly behind the Windows Firewall.

🌟 Important: These instructions assume that you are using TomEE's default port settings. To ensure that your port settings match those in this procedure, contact your local TomEE administrator.

1. Click Start, point to Control Panel / Network Connections, and click Local Area Connection.
   The Properties dialog box appears.
2. Click the Advanced tab.
3. In the Windows Firewall area, click Settings.
   The Windows Firewall dialog box appears.
4. Click the Exceptions tab.
   This tab allows you to define programs and ports that are not impacted by the Windows Firewall. You need to specify certain ports for the SIMULIA Execution Engine.
5. Click Add Port.
   The Add a Port dialog box appears.
6. Enter the TomEE information. For more information, consult your local system administrator.
7. Click OK.
   You are returned to the Exceptions tab, and the new port exception is added to the list. You may have to scroll down to see it because the exceptions are listed alphabetically.
8. Click OK to close the Windows Firewall dialog box.
9. Click OK to close the Properties dialog box.
Resolving Publishing Errors on Windows

You can set the environment variable TEMP to avoid errors when using the SIMULIA Execution Engine publishall program.

When running programs on Windows, your user name is part of the default temporary directory path C:\Program Files\user_name\Local Settings\Temp.

If the path to the temporary directory contains any of the following characters:
$ @ # & % ? !
the Java ClassLoader cannot read files in that directory and the SIMULIA Execution Engine publishall program will produce the following error:

cannot open super metamodel
com.engineous.component.Plugin

If your user name contains any of the restricted characters listed above, you can avoid this problem by setting the environment variable TEMP to point to a directory that does not contain any special characters (such as C:\temp). For more information on setting this environment variable, contact your local system administrator.

Linux-based SIMULIA Execution Engine Stops Functioning Correctly

If you installed the SIMULIA Execution Engine on a Linux system, verified that it was fully functional, stopped using it for a time, and then were unable to run any jobs on it (your jobs stop running before the first work item is executed), the problem may be caused by the automatic removal of your SIMULIA Execution Engine temporary directory.

The default setting for this directory is /tmp. However, some temporary directories on Linux are automatically “cleaned” from time to time. If this temporary directory (or its contents) are deleted, your SIMULIA Execution Engine will stop functioning correctly.

Since the settings for your file system are unique to your environment, it is highly recommended that you consult your local system administrator to determine a stable part of your file system for this temporary directory.

You can alter this temporary directory setting after your installation by changing the fiper.system.temp setting in the acs.properties file. This file is located at the top level of your SIMULIA Execution Engine installation directory. Once you alter this setting, you must restart your SIMULIA Execution Engine so that the newly defined directory is used by the system.

Changing Your SIMULIA Execution Engine Passwords

Passwords for various accounts are stored inside the SIMULIA Execution Engine or SIMULIA Execution Engine station configurations in the locations described below.

Some or all of the passwords must be updated if the password for one of the SIMULIA Execution Engine utility user accounts is changed.

- The password for an account used to log a station running as a service in to the SIMULIA Execution Engine is stored in the station service configuration.
On Linux, this password is stored in the `FiperStation` file, which is located in the `/etc/init.d` directory. This file is a copy of the `station.service` file, which is located in the `<SEE_install_directory>/<operating_system>/code/command` directory.

On Windows, the password is stored in the `Wrapper.conf` file, which is located in the `<SEE_install_directory>/<operating_system>/code/command` directory.

To fix the password on all operating systems:

1. Change the password of the account.
2. Uninstall the station as a service on the system.
3. Reinstall the station as a service, supplying the new password.

Symptom: If this password is wrong, the station service will not start. You can view error messages in the `station.log` file (located in the station temporary directory, which was specified during the station installation and is listed in the `station.properties` file at the top level of the SIMULIA Execution Engine installation directory).

**Note:** If a station running as a service on Windows is set up to run as a user other than the default `LOCAL_SYSTEM`, you must update the Service configuration after changing the password. To change the password, access the Services control panel, open the Properties dialog box for the station service, click the Log On tab, and update the logon information. For more information on accessing these settings, see *Installing a SIMULIA Execution Engine Station as a Service* or contact your local system administrator. This procedure must be performed in addition to uninstalling and re-installing the service as described above, since the logon user for a service is set after the service is created.
Index

I
Isight Desktop Installation
  Silent Mode 10

L
license utility 84

R
report utility 84

S
SIMULIA Execution Engine Installation
  TUI Mode 8
  SIMULIA Execution Engine stations 1, 55
Station Installation Linux
  GUI Mode 21
About SIMULIA
Dassault Systèmes SIMULIA applications, including Abaqus, Isight, Tosca, and Simulation Lifecycle Management, enable users to leverage physics-based simulation and high-performance computing to explore real-world behavior of products, nature, and life. As an integral part of Dassault Systèmes 3DEXPERIENCE® platform, SIMULIA applications accelerate the process of making highly informed, mission-critical design and engineering decisions, before committing to costly and time-consuming physical prototypes. www.3ds.com/simulia

Our 3DEXPERIENCE® platform powers our brand applications, serving 12 industries, and provides a rich portfolio of industry solution experiences.
Dassault Systèmes, the 3DEXPERIENCE® Company, provides business and people with virtual universes to imagine sustainable innovations. Its world-leading solutions transform the way products are designed, produced, and supported. Dassault Systèmes’ collaborative solutions foster social innovation, expanding possibilities for the virtual world to improve the real world. The group brings value to over 190,000 customers of all sizes in all industries in more than 140 countries. For more information, visit www.3ds.com.