DASSAULT SYSTEMES

SIMULIA

SIMULIA EXECUTION ENGINE 5.9

INSTALLATION AND CONFIGURATION GUIDE
WebSphere AND Oracle/DB2

3DEXPERIENCE

Legal Notices

Isight and the SIMULIA Execution Engine are products of Dassault Systèmes Simulia Corp., Providence, Rhode Island, USA. These products may be used or reproduced only in accordance with the terms of such license.

This documentation is subject to the terms and conditions of either the software license agreement signed by the parties, 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. No part of this documentation may be reproduced or distributed in any form without prior written permission of Dassault Systèmes or its subsidiary.

Isight, Abaqus, the 3DS logo, and SIMULIA are trademarks or registered trademarks of Dassault Systèmes or its subsidiaries in the United States and/or other countries. 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 at `<isight_install_directory>\Doc\Third_Party_Products.html`.

OPEN SOURCE PROGRAMS: This release of Isight and the SIMULIA Execution Engine uses several open source or free programs (“OS Programs”). Each such program is distributed with the Isight and the SIMULIA Execution Engine Software in binary form and, except as permitted by the applicable license, without modification. Each such program is available online for free downloading and, if required by the applicable OS Program license, the source code will be made available by Dassault Systèmes or its subsidiary upon request. For a complete list of OS Programs utilized by Isight and the SIMULIA Execution Engine, as well as licensing documentation for these programs, see `<isight_install_directory>\Doc\Third_Party_Products.html`.

Dassault Systèmes or its subsidiaries may have patents or pending patent applications, trademarks, copyrights, or other intellectual property rights covering its software and/or its documentation. No license of such patents, trademarks, copyrights, or other intellectual property rights is provided or implied except as may be expressly provided in a written license agreement from Dassault Systèmes or its subsidiary.

Preface

This section lists various resources that are available for help with using SIMULIA software, including technical engineering and systems support, training seminars, and documentation.

Platform Information

SIMULIA applications are supported on a variety of platforms.
For complete details on supported platforms, refer to
http://www.3ds.com/support/certified-hardware/simulia-system-information/

Support

Both technical engineering support (for problems with creating a model or performing an analysis) and systems support (for installation, licensing, and hardware-related problems) for Isight are offered through a network of local SIMULIA support offices.

SIMULIA Online Support System

SIMULIA Execution Engine Express is easier to install than the standard SIMULIA Execution Engine. The server and database are installed with the software and are pre-configured and ready to execute. SIMULIA provides a knowledge database of answers and solutions to questions that we have answered, as well as guidelines on how to use Abaqus, SIMULIA Scenario Definition, Isight, SIMULIA Execution Engine, and other SIMULIA products. You can also submit new requests for support. All support incidents are tracked. If you contact us by means outside the system to discuss an existing support problem and you know the incident or support request number, please mention it so that we can query the database to see what the latest action has been.

Many questions can also be answered by visiting *http://www.3ds.com/products/simulia*. The information available online includes:

- Systems information and computer requirements
- Performance data
- Status reports

- Training seminar schedule
- INSIGHTS Magazine/Realistic Simulation News Magazine
- Technology briefs
- Customer conference papers

Technical Engineering Support

Technical support engineers are available to assist in clarifying product features and checking errors by giving both general information on using the product and information on its application to specific analyses. If you have concerns about an analysis, we suggest that you contact us at an early stage, since it is usually easier to solve problems at the beginning of a project rather than trying to correct an analysis at the end.

Please have the following information ready before contacting the technical engineering support hotline, and include it in any written contacts:

- The release of Isight that are you using, which can be obtained by accessing the `VERSION` file in the following directory:

  ```
  <install_directory>\<operating_system>\reffiles\SMAFIPconfig
  ```

- The type of computer on which you are running Isight.
- The symptoms of any problems, including the exact error messages, if any.
- Any log files associated with the error.
- Workarounds or tests that you have already tried.

When contacting support about a specific problem, any available product output files may be helpful in answering questions that the support engineer may ask you.

The support engineer will try to diagnose your problem from the model description and a description of the difficulties you are having. The more detailed information you provide, the easier it will be for the support engineer to understand and solve your problem.

If the support engineer cannot diagnose your problem from this information, you may be asked to supply a model file. The data can be attached to a support incident in the online system. It can also be sent by means of e-mail, disk, or ftp. Please check the **Support** page at *http://www.3ds.com/products/simulia* for the media formats that are currently accepted.

If you are contacting us via telephone to discuss an existing problem, please give the receptionist the support engineer's name. If you are contacting us via e-mail, please include the support engineer's name at the top of any e-mail correspondence. If you are contacting us online (preferred for written communication), update the existing incident/support request for the problem.

Systems Support

Systems support engineers can help you resolve issues related to the installation and running of the product, including licensing difficulties, that are not covered by technical engineering support.

You should install the product by carefully following the instructions in the installation guide. If you encounter problems with the installation or licensing, first review the instructions in the installation guide to ensure that they have been followed correctly. If this does not resolve the problems, consult the knowledge database for information about known installation problems. If this does not address your situation, please create an incident/support request in the online system and describe your problem.

Anonymous FTP Site

To facilitate data transfer with SIMULIA, an anonymous ftp account is available on the computer `ftp.simulia.com`. Login as user `anonymous`, and type your e-mail address as your password. Contact support before placing files on the site.

Contacting Technical Support

Use the **Support** page at *http://www.3ds.com/products/simulia*, or obtain local support office contact information from the **Locations** page at *http://www.3ds.com/products/simulia*.

In addition, contact information for offices and representatives is listed in the preface.

Support for Academic Institutions

Under the terms of the Academic License Agreement we do not provide support to users at academic institutions.
Academic users can purchase technical support on an hourly basis. For more information, please see *http://www.3ds.com/products/simulia* or contact your local support office.

Training

SIMULIA offices offer regularly scheduled public training classes, including classes on Isight. SIMULIA offices also provide training seminars at customer sites. All training classes and seminars include workshops to provide practical experience with our products. For a schedule

and description of available classes, see the **Services** page at *http://www.3ds.com/products/simulia* or call your local representative.

Feedback

SIMULIA welcomes any suggestions for improvements to Isight software, the support program, or documentation.
If you wish to make a suggestion about the service or products, refer to *http://www.3ds.com/products/simulia*. Complaints should be addressed by contacting your local office or through *http://www.3ds.com/products/simulia*.

Contents

What You Need to Know Before Installing the SIMULIA Execution Engine.1
About the SIMULIA Execution Engine Environment....1
Basic Installation Steps for the SIMULIA Execution Engine....3
Installing as an Administrator (Windows) or Non-Root User (Linux)....3
What's New?....5
Prerequisites....6
Software Requirements....6
Operating System....6
Database and Java EE Application Server....6
Configuring WebSphere to Use Java 7....6
Web Browser and PDF Viewer Software....7
Hardware Requirements....7
Updating the Windows Firewall....8
Disabling Real-Time Virus Scanning....8
Migrating to SIMULIA Execution Engine 5.9 8
Installing the SIMULIA Execution Engine Server....9
Installing the SIMULIA Execution Engine....9
Uninstalling the SIMULIA Execution Engine....10
Closing the SIMULIA Execution Engine Client Applications....11
Removing the SIMULIA Execution Engine Software....11
Deleting Temporary Directories and Files on Windows....12
Deleting Temporary Directories and Files on Linux....13
Installing a SIMULIA Execution Engine Station on Windows....15
Installing the Station Software on Windows....15
Installing a SIMULIA Execution Engine Station as a Service....17
Installing a SIMULIA Execution Engine Station as a Service....17
Starting a SIMULIA Execution Engine Station as a Service....18
Stopping a SIMULIA Execution Engine Station Service....19
Uninstalling a SIMULIA Execution Engine Station Service....19
Uninstalling a SIMULIA Execution Engine Station....19
Stopping the Station....20
Removing the SIMULIA Execution Engine Station Software....20

Deleting Station Temporary Directories and Files on Windows....21
Installing a SIMULIA Execution Engine Station on Linux....23
Before You Begin....23
Shared Network Install....23
Root Privileges....23
Installing the Station Software on Linux....24
Enabling the SIMULIA Execution Engine Station Security Feature (Run-As)....26
Installing a SIMULIA Execution Engine Station as a Service Manually....27
Uninstalling a SIMIULIA Execution Engine Station....28
Stopping the SIMULIA Execution Engine Station....28
Removing the SIMULIA Execution Engine Station Software....28
Deleting Temporary Files and Other Files....29
Initializing the SIMULIA Execution Engine Database....31
Initializing an Oracle Database....31
About Oracle URLs and Port Numbers....31
Creating Tablespaces and Defining User Information....32
Updating the Database....35
Creating the Database Tables....36
Initializing a DB2 Database....37
About the DB2 User....37
Updating the Initialization File....38
Executing the Database Scripts....40
Configuration....43
Setting the DSLS_CONFIG Environment Variable....43
Configuring Your FLEXnet License to Work with a Windows Firewall....43
Setting Station Execution Permissions for the Excel and Word Components....44
Setting Word and Excel Privileges for Stations Installed as a Service....46
Setting Excel 2010 Options for Windows 7 Stations....46
Configuring the Excel Properties and Component....47
Configuring station.properties Values....47
Replacing a Process Level Token....48
Verifying that the Desktop Folder Exists....48
Configuring the Excel 2010 Macro....48
Disabling Excel 2010 Add-ins....49
Editing the Registry....49
Editing User Account Controls....50
Creating an Environment Variable....50

Configuring WebSphere.....52
About Configuring WebSphere.....52
Manually Configuring the SIMULIA Execution Engine.....53
Starting WebSphere and Determining Server Port Numbers.....53
Creating a J2C Authentication Alias for JDBC Datasources.....55
Adding JDBC Providers.....56
Creating Datasources.....58
Setting the fiperhome Variable and the Library Options.....63
Enabling the Startup Beans Service.....65
JMS Configuration - Configuring Service Integration Bus.....65
Creating JMS Destinations.....66
Creating Queues.....68
Creating Topics and the Connection Factory.....69
Creating the Activation Specifications.....72
Configuring the WebSphere JVM.....75
Setting the DSLS_CONFIG Environment Variable.....78
Deploying the SIMULIA Execution Engine EAR File.....79
Installing the WebTop and WebDashboard.....80
Automatically Configuring the SIMULIA Execution Engine.....81
Prerequisites.....81
Params.txt Quick Reference.....81
Executing the Scripts.....83
Limitations.....84
Enabling Security.....85
Restarting the SIMULIA Execution Engine in WebSphere.....85
About Starting the SIMULIA Execution Engine Server.....86
Restarting WebSphere with No Security Enabled.....86
Restarting WebSphere with Security Enabled.....87
About Stopping the SIMULIA Execution Engine Server.....88
Creating the Connection Profile and Preloading the Library.....89
Creating the Connection Profile File.....89
Publishing to the Library.....90
Understanding the acs.properties File Settings.....91
Configuring Security.....97
About SIMULIA Execution Engine Security.....97
Configuring SIMULIA Execution Engine Security.....99
About Client Authentication.....99

About SIMULIA Execution Engine Access Control Lists....100
Specifying the WebSphere Security Settings....102
Updating Windows Shortcuts for Security Authorization....108
Configuring Station (Run-As) Security....109
About Station Run-As Security....109
About User Credential Encryption....110
About Securing the SIMULIA Execution Engine Station File System....112
About Run-As Security Limitations....114
Configuring the Run-As Feature....115
Configuring the WebTop or WebDashboard for the SIMULIA Execution Engine....121
Determining Your Deployment Strategy....121
Configuration Steps for Different Architectures....121
Creating a New WebSphere Profile....123
About New Profile Port Numbers....123
Creating a Profile Using the Profile Management Tool....123
Creating a Profile Using Command Line Options....126
Deploying the Web-Based Applications....127
Configuring the Web-Based Applications....129
Configuring the Application on a Different System or Separate Profile....129
Configuring the Application in the Same Profile....138
Verifying the Installation....142
About the Application Port Number and URL....142
Viewing the Application....143
Using SIMULIA Execution Engine Interfaces....145
Using the SIMULIA Execution Engine Station....145
About the SIMULIA Execution Engine Station Interface....146
About Log Message Detail Levels....147
About Station Affinities....148
Station Status Reporting in the Dashboard and WebDashboard....149
About Running Multiple Stations on a Single Host Computer....149
Starting a SIMULIA Execution Engine Station....150
Shutting Down a SIMULIA Execution Engine Station....152
Restarting a SIMULIA Execution Engine Station Remotely....152
Configuring SIMULIA Execution Engine Station Properties....153
Using the Dashboard....172
About the Dashboard Interface....172
Starting the Dashboard....174

Viewing Connection Information....179
Viewing Station Information....179
Controlling Station Workitems....179
Shutting Down, Restarting, or Deleting a Station....180
Managing Access Control for SIMULIA Execution Engine Users....180
Managing Access Control – System Administration....181
Viewing License Usage Information....182
Using the WebDashboard....183
About the WebDashboard Interface....183
Accessing the WebDashboard in a Browser....185
Viewing Station Information....186
Restarting or Shutting Down a Station....187
Viewing Details for the SIMULIA Execution Engine....188
Working with Running Jobs....188
Searching for Jobs....190
Deleting Non-Running Jobs....191
Viewing License Usage Information....192
Using the Command Line Client....192
Starting the Command Line Client....193
Using Distributed Resource Management with the SIMULIA Execution Engine....195
Default Fiper DRM included with the SIMULIA Execution Engine....195
LSF DRM and the SIMULIA Execution Engine....196
Mixed-Mode DRM and the SIMULIA Execution Engine....199
Verifying the SIMULIA Execution Engine Configuration....200
Configuring LSF for the SIMULIA Execution Engine....201
Creating the seeadmin User on Windows....206
Creating the seeadmin User....206
Setting the seeadmin User Privileges....209
Checking Password Policies....209
Creating an Oracle Database for the SIMULIA Execution Engine....211
Creating the Database in Oracle 11gR2....211
Generating Reports of SIMULIA Execution Engine License Usage..215
About License Usage Reports....215
Running the licusage Utility....215
License Report Utility Options....216
General Options....216

Export Options....218
Query Options....219
Output Options....220
Advanced Options....220
Examples....220
Basic Troubleshooting....223
User Login Names Containing Punctuation....223
Log Files for the SIMULIA Execution Engine....224
Configuring the Windows Firewall....224
Configuring the Windows Firewall for WebSphere....225
Resolving Publishing Errors on Windows....226
Fixing Network Connection Problems....227
IP Configuration Workaround....228
Linux-based SIMULIA Execution Engine Stops Functioning Correctly....229
Changing Your SIMULIA Execution Engine Passwords....229
DB2 Package Problem....232
Copying the WebSphere JAR Files....233
Backup and Restore Procedures....234
Backing up SIMULIA Execution Engine Data....234
Restoring the SIMULIA Execution Engine....235

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.

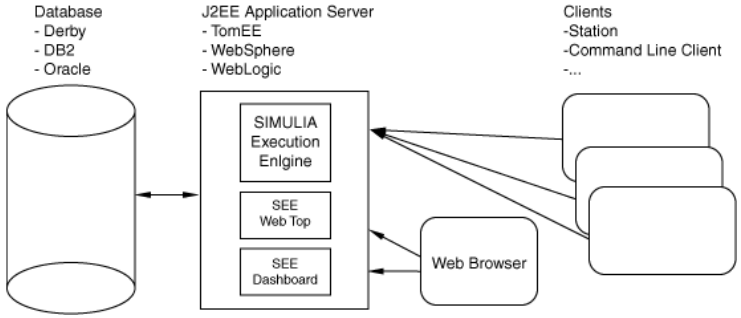


Figure 1: SIMULIA Execution Engine Architecture

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*.

WebDashboard

The SIMULIA Execution Engine WebDashboard is a Web browser-based interface that displays the current status of the SIMULIA Execution Engine. It is similar to the Dashboard interface, except that it runs in a browser and does not require you to install any Isight software. The WebDashboard 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. It also allows you to search for a particular job using specific search criteria. For more information, see *Using the WebDashboard*.

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.

Note: If you are installing only a SIMULIA Execution Engine station, these steps do not apply. For more information, see *Installing a SIMULIA Execution Engine Station on Windows* or *Installing a SIMULIA Execution Engine Station on Linux*.

1. Install the SIMULIA Execution Engine software.

   For details, see *Installing the SIMULIA Execution Engine Server*.
2. Install and upgrade (if necessary) an application server and database combination to run the SIMULIA Execution Engine on.
3. Configure the SIMULIA Execution Engine.

   For details, see *Configuration*, *Configuring WebSphere*, and *Configuring Security*.
4. Start the SIMULIA Execution Engine in the application server.

   See *Restarting the SIMULIA Execution Engine in WebSphere*.

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

Supported Versions of Middleware	SIMULIA Execution Engine now supports WebSphere 8.5.0 with Oracle 11gR2 or DB2 9.5.
DS Standard Installer	The Dassault Systemes standard installer is now used to install Isight and the SIMULIA Execution Engine.

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/*.

Database and Java EE Application Server

The SIMULIA Execution Engine must be run on one of the supported database and application server combinations.

For the list of the supported databases and application servers, see the **SIMULIA Platforms & Configuration Support** section at *http://www.3ds.com/products/simulia/support/*.

The SIMULIA Execution Engine software and WebSphere must be installed on the same computer for the SIMULIA Execution Engine to function properly. The recommended hardware configuration is to install the WebSphere application server and the Oracle or DB2 database on separate server computers, with a high-speed, low-latency network connection between the two machines. Installing and running the database and application server on the same machine is not recommended but is allowed.

Configuring WebSphere to Use Java 7

WebSphere 8.5.0.0 works on bundled IBM WebSphere SDK Java Technology Version 6.0. However, SIMULIA Execution Engine 5.9 requires 64-bit Java 7. You must manually configure WebSphere 8.5.0.0 to use WebSphere SDK Java Technology Edition Version 7.0.

1. Navigate to the `<WAS_Installation>\AppServer\bin` directory and run the following command to obtain a list of the available Java SDKs:

   ```
   managesdk -listAvailable
   ```

   Ensure that the SDK name 1.7_64 is available. If you do not have IBM WebSphere SDK Java Technology Edition Version 7.0, you can download it at the following URL:

   http://pic.dhe.ibm.com/infocenter/wasinfo/v8r5/index.jsp?topic=%2Fcom.ibm.websphere.installation.nd.doc%2Fae%2Fins_installation_jdk7_gui.html

2. Run the following command to change the default SDK to version 7.0 SDK:

   ```
   managesdk -setCommandDefault -sdkname 1.7_64
   ```

3. Run the following command to set the new profile default to version 7.0 SDK:

   ```
   managesdk -setNewProfileDefault -sdkname 1.7_64
   ```

4. Run the following command to set the existing profile default to version 7.0 SDK:

   ```
   managesdk -enableProfileAll -sdkname 1.7_64 -enableServers
   ```

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 WebSphere 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:

```
<SEE_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.

For more information, see *Configuring 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 5.9

You do not have to uninstall SIMULIA Execution Engine 5.8 or a 5.8 station before you install SIMULIA Execution Engine 5.9 or a 5.9 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.

Installing the SIMULIA Execution Engine

To install the SIMULIA Execution Engine on either Windows or Linux, you run the installation wizard.

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

Root (superuser) privileges are not required to install the 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`.

Installing on Linux 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.
2. Start the installer.

   On Windows, execute the following file:

   ```
   <dvd_media_dir>\IsightSEE5.9-1-Windows\setup.exe
   ```

   On Linux, execute the following file:

   ```
   <dvd_media_dir>/IsightSEE5.9-1-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 the products 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 desktop computers and servers.

5. **Optional:** If desired, you can create a `.cpr` connection profile file to connect stations to the SIMULIA Execution Engine server. Select the application server, and click **Next** to enter the server host name and port number.

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

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 or WebDashboard 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 or WebDashboard to close SIMULIA Execution Engine Stations and stop running jobs. For more information on using the SIMULIA Execution Engine Dashboard, see *Using the Dashboard*. For more information on using the SIMULIA Execution Engine WebDashboard, see *Using the WebDashboard*.
6. Close the SIMULIA Execution Engine Dashboard or WebDashboard.

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\5.9`, navigate to the `C:\SIMULIA\Execution Engine` directory.
2. Delete the `5.9` 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/5.9`, navigate to the `/opt/SIMULIA/Execution Engine` directory.
3. Delete the 5.9 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.

To remove the SIMULIA Execution Engine Station software, use the uninstaller wizard.

Installing the Station Software on Windows

The SIMULIA Execution Engine installation wizard guides you through the process of installing a SIMULIA Execution Engine station on any computer.

This wizard is used for both standard SIMULIA Execution Engine installations and when installing only a SIMULIA Execution Engine station.

1. Log in as the Administrator that will install the station.

   You must be an Administrator user to install the SIMULIA Execution Engine station.
2. Insert the SIMULIA Execution Engine installation DVD, or extract the archive file of the installation media.
3. Start the installer.

   On Windows, execute the following file:

   ```
   <dvd_media_dir>\IsightSEE5.9-1-Windows\setup.exe
   ```
4. If necessary, click **Allow** on the **User Account Control** dialog to confirm the execution of the installation program.
5. 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 the product you want to install:

```
SIMULIA Execution Engine Station
```

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. Click **Finish** to allow the installer to complete.
8. After installation, you can use the **Edit Logon Profile** tool to create a new `.cpr` connection profile. See "Creating a SIMULIA Execution Engine Connection Profile" in the *Isight User's Guide*. The `filename.cpr` connection profile must point the station to the SIMULIA Execution Engine server.

   The `.cpr` file must specify the following:

   - **Server 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 Type**. WebSphere, WebLogic, or TomEE.
   - **Port Number**

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\
```

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, WebDashboard, 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\5.9`, navigate to the `C:\SIMULIA\Execution Engine` directory.
2. Delete the `5.9` 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:

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

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

This section describes how to install a SIMULIA Execution Engine station on Linux platforms.

Before You Begin

This section describes information you should know before you install a SIMULIA Execution Engine station on Linux.

Shared Network Install

The SIMULIA Execution Engine station may be installed once on a network file system and then run on many different computers.

Once the SIMULIA Execution Engine station has started, it places a small load on the file server. Because a SIMULIA Execution Engine station must reliably run for large periods of time, the file system it is using should be mounted with the NFS option `hard` (not `soft`).

If a shared disk is used to run SIMULIA Execution Engine stations with the Run-As option enabled, the file system must be mounted with the NFS option `suid`. This action allows the `set-user-id` permission on program `SMAFIPplaunch` to be effective.

Root Privileges

You can install the SIMULIA Execution Engine station as a root user or a non-root user.

The SIMULIA Execution Engine station is typically installed from an account with root privileges. The SIMULIA Execution Engine station can also be installed from a nonprivileged account, but there are certain restrictions you need to be aware of when doing so:

- The default install location can be written only by a privileged user. If you are installing as a nonprivileged user, you will have to change the install directory to a directory to which you can write.
- Installing a SIMULIA Execution Engine station as a service requires root privileges. If you install from an unprivileged account, the SIMULIA Execution Engine station will

have to be started manually, and you cannot log out while the SIMULIA Execution Engine station is running.

- The station Security Feature (called *run-as-user*) requires the program `SMAFIPplaunch` to be installed with `set-user-id` root privileges. A SIMULIA Execution Engine station installed as an unprivileged user cannot connect to a SIMULIA Execution Engine that has the SIMULIA Execution Engine station security feature enabled unless the file `SMAFIPplaunch` is manually changed to have `set-user-id` root privileges. Instructions for performing this step are displayed during the installation. For more information, see *Configuring SIMULIA Execution Engine Stations for Run-As on Linux*.

Installing the Station Software on Linux

Follow the steps below to install a SIMULIA Execution Engine station on Linux.

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 SIMULIA Execution Engine installation DVD, or extract the archive file of the installation media.
3. Start the installer.

   On Linux, execute the following file:

   ```
   <dvd_media_dir>/IsightSEE5.9-1-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 the product you want to install:

     ```
     SIMULIA Execution Engine Station
     ```

5. 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.

6. Click **Finish** to allow the installer to complete.
7. After installation, you can use the **Edit Logon Profile** tool to create a new `.cpr` connection profile. See "Creating a SIMULIA Execution Engine Connection Profile" in the *Isight User's Guide*. The `filename.cpr` connection profile must point the station to the SIMULIA Execution Engine server.

   The `.cpr` file must specify the following:

   - **Server 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 Type**. WebSphere, WebLogic, or TomEE.
   - **Port Number**

8. 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/
```

   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 SIMIULIA 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:

   ```
   ./rm -rf *
   ```

   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/5.9`, navigate to the `/opt/SIMULIA/Execution Engine` directory.
3. Delete the 5.9directory 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
   - `Fiperinstall.log` file
   - `Fiperuninstall.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.

Initializing the SIMULIA Execution Engine Database

This section describes how to create and prepare an Oracle or DB2 database for use with the SIMULIA Execution Engine.

Initializing an Oracle Database

Before the SIMULIA Execution Engine is configured, an Oracle database tablespace and user name must be defined, the database must be updated, and the appropriate tables must be created within the database.

The instructions in this section assume that you have created a user called `seeadmin` in the operating system on your server computer. This user account will be used to control the configuration of the SIMULIA Execution Engine. For information about configuring this user account on Windows, see *Creating the seeadmin User on Windows*.

These instructions apply to Oracle 11*g*R2.

About Oracle URLs and Port Numbers

If you are accessing the Oracle web-based interface on Linux, you must manually specify the URL and the port number for the correct database.

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

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

In Oracle 11*g*R2, 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 an Oracle Database for the SIMULIA Execution Engine*.

   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:

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

   Uuse `https` instead of `http` in this URL. 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 an Oracle Database for the SIMULIA Execution Engine* 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>\bin` 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/SMAFIPserver/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 `seeadmin`), `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.

Initializing a DB2 Database

Before the SIMULIA Execution Engine is configured, a DB2 database must be initialized, which creates the database name and the database's user name, and the appropriate tables must be created within the database.

About the DB2 User

Before initializing a DB2 database, you need to verify that the database user has been created and that the user has been placed into the correct DB2 group.

The process for verifying these settings varies based on the operating system that is running the database.

Windows DB2

If you are using a Windows-based DB2 database with your SIMULIA Execution Engine, you need to add the database user (`seeadmin`) to a specific Windows group. These steps are not necessary if you are using an Oracle database.

1. Create a new Windows user called `seeadmin`.
2. Add the user to the local group `DB2ADMNS`. This step is necessary to allow the user to create and alter the DB2 database.

Note: The remaining procedures in this section assume that you have created a new user called `seeadmin`. If you create a user with a different name or use an existing user with a different name, be sure to use that user during all of the configuration steps.

For detailed information on the steps necessary to set up a user and set these permissions, see *Creating the seeadmin User on Windows*.

Linux DB2

If you are using a Linux-based DB2 database with your SIMULIA Execution Engine, the designated user (usually `seeadmin`) needs to be in the DB2 administration group to be able to create a database. In addition, the home directory of the SIMULIA Execution Engine designated user (`seeadmin`) needs to be "world writable" at database creation time for one of the DB2 administrative users to write a file. This arrangement is not secure and should be a temporary setting only. For more information, contact your local system administrator.

About DB2 Database Scripts

Several DB2 scripts are included with your SIMULIA Execution Engine installation. These scripts can be used to create your SIMULIA Execution Engine database, create the tables within the database, remove the contents of the database tables, or remove the tables entirely from the database.

Note: On Windows, each of these scripts uses the `.bat` extension.

The following scripts are included:

`initdb`	This script creates the SIMULIA Execution Engine database. For more information on the settings within this script, see *Updating the Initialization File*. For more information on using this script, see *Executing the Database Scripts*.
`createtables`	This script creates the SIMULIA Execution Engine tables in an existing database. For more information on using this script, see *Executing the Database Scripts*.
`cleartables`	This script cleans the tables of their contents, but it leaves them in place. You must reexecute the `publishall` command after running this script. All jobs, results, and library content are removed.
`droptables`	This script removes the SIMULIA Execution Engine tables in an existing database. It should be used when your table structure is modified because of a change in the SIMULIA Execution Engine infrastructure. It will be used rarely, if ever. The `createtables` script should be run after running this script.

Updating the Initialization File

Before running the database creation script, you need to verify that the default information provided in the database initialization file is correct for your environment.

1. Verify that you are logged in as the same user that installed the DB2 software. Contact your database or system administrator to determine database user information.
2. Navigate to the following directory:

   `<SEE_install_dir>/<os_dir>/reffiles/SMAFIPserver/db/db2/`

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

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

3. Open one of the following files in the text editor of your choice:

   - Windows: `initdb.bat`
   - Linux: `initdb`

4. Review the following settings and, if necessary, update them appropriately:

   `FIP_DB`. The name of the SIMULIA Execution Engine database. The default setting is `fiper`.

   `FIP_USER`. The user with DB2 administrative privileges. The default setting is `fiperacs`.

   `FIP_SERVERSIZE`. The size of the system running the database. One of the following options should be used:

   - `small`. Used for a computer with 1 GB of memory.
   - `large`. Used for a server with at least 1.5 GB of memory.

   `FIP_TABLEDIR`. The directory in which you want to create buffer pool storage. This setting should be changed if you installed DB2 in a location other than `C:\DB2` (Windows) or `/opt/IBM` (Linux). The default settings are `C:\DB2\NODE0000\FIPER` (Windows) or `/opt/IBM/db2` (Linux).

   `FIP_Territory`. The location that controls localization of the database. The default setting is `us`. This entry must be set to one of the following options:

   - `us` (United States)
   - `ca` (Canada)
   - `cn` (China)
   - `fr` (France)
   - `de` (Germany)
   - `jp` (Japan)

- `kr` (South Korea)
- `tw` (Taiwan)
- `gb` (United Kingdom)

For additional territory location codes, see your DB2 documentation.

For example, to update this setting to use the Germany option, your entry would appear as follows:

```
SET FIP_TERRITORY=de
```

5. Continue by running the database initialization scripts; see *Executing the Database Scripts*.

Executing the Database Scripts

Two database scripts must be executed to initialize your DB2 database and to create the tables needed by the SIMULIA Execution Engine. These scripts are included with your SIMULIA Execution Engine installation.

1. Verify that you are logged in as the same operating system user that installed the DB2 software.
2. Open a Command Prompt dialog box (terminal window on Linux).

   Important: If you are running your database on Windows Server 2008 or Windows 7, the database scripts must be run from a Command Prompt dialog box with full administrator privileges. To launch this type of Command Prompt dialog box, locate the `cmd.exe` file using Windows Explorer (this file is usually located in the `C:\Windows\system32` directory), right-click the file, and select **Run as Administrator**. When prompted, click **Continue** on the **User Account Control** dialog box.

3. Navigate to the following directory:

   <SEE_install_dir>`/`*<os_dir>*`/reffiles/SMAFIPserver/db/db2/`

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

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

4. Type one of the following commands, based on your operating system:

   - Windows: `initdb.bat`

- Linux: `./initdb`

A message appears indicating the database was created.

5. Verify that no errors occurred during the script execution.

   This script creates a log file (`initdb.log`) in this same directory, which can be examined for more information.

6. Type one of the following commands, based on your operating system:

   - Windows: `createtables.bat`
   - Linux: `./createtables -all`

   You are prompted to enter your password.

7. Type the password of the user specified using the `FIP_USER` setting as described in *Updating the Initialization File*, and press **ENTER**.

   A message appears indicating that the tables were created.

8. Verify that no error messages appeared when the scripts were executed.

   This script creates a log file (`createtables.log`) in this same directory, which can be examined for more information.

9. Navigate to the top level directory of your DB2 installation.
10. Open the `db2cli.opt` file in the text editor of your choice.
11. Add the following lines anywhere within the file:

    ```
    [common]

    Patch2=50
    ```

12. Save and close the file.
13. From a Command Prompt dialog box (terminal window on Linux), navigate to the following directory:

    ```
    <db2_install_directory>/bin/
    ```

14. Stop and start DB2 by typing the following commands one after the other at the command prompt.

    ```
    db2stop

    db2start
    ```

15. If your DB2 database is running on a Windows system, reboot the system.

    This step is necessary before proceeding to the next section.

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.
4. Double-click **Computers**, and double-click **My Computer**.
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 2010 Options for Windows 7 Stations

This section explains the steps to take when you are executing Microsoft Excel 2010 components on SIMULIA Execution Engine Windows 7 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 2010 components on Windows 7 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.substation.launchtimeout=300000`
   - `fiper.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 2010 components on Windows 7 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:

   `C:\Windows\SysWOW64\config\systemprofile\Desktop`

2. Verify that the directory is writeable for all run-as users.

Configuring the Excel 2010 Macro

You can configure Excel macros to reduce the risk of jobs failing when executing Excel 2010 components on Windows 7 stations.

1. From Excel 2010, open a worksheet.
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.

Disabling Excel 2010 Add-ins

You can disable the Excel 2010 add-ins to reduce the risk of jobs failing when executing Excel 2010 components on Windows 7 stations.

1. Click the Windows Start button, and then select **Control Panel**.
2. From the **Control Panel**, select **Uninstall a program**.
3. Navigate to Microsoft Office 2010, and click **Change**.
   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 2010 components on Windows 7 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 2010 components on Windows 7 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\5.9
```

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 WebSphere

This section describes how to set up and start the SIMULIA Execution Engine using the WebSphere application server and an Oracle or DB2 database.

About Configuring WebSphere

This section describes the configuration of the SIMULIA Execution Engine including configuring WebSphere and starting the application.

The configuration procedures in this section assume that you created a user called `seeadmin` to control the configuration of the SIMULIA Execution Engine.

To fully configure the SIMULIA Execution Engine application within WebSphere, you must perform the following main steps:

1. Follow all of the instructions under *Manually Configuring the SIMULIA Execution Engine* or *Automatically Configuring the SIMULIA Execution Engine*.
2. Optionally configure usernames, passwords, and other security features as described in *Enabling Security* and in *Configuring Security*.
3. Stop and restart WebSphere as described in *Restarting the SIMULIA Execution Engine in WebSphere*, to force WebSphere and the SIMULIA Execution Engine to recognize the configuration changes.
4. Preload the basic system metamodels for Isight and the SIMULIA Execution Engine, as described in *Creating the Connection Profile and Preloading the Library*.

The `acs.properties` file options are described in *Understanding the acs.properties File Settings*.

About WebSphere URL Port Numbers

The WebSphere Administrative console is accessed via a web browser using a URL that is specific to the system running WebSphere. You need to determine the correct port number before accessing the console, especially if you are using a Linux system.

A sample URL for accessing the console is shown below:

```
http://seecomputer:9060/ibm/console
```

By default, the WebSphere Administrative console uses the following port numbers: 9060 (non-secure) and 9043 (secure). However, these port numbers are incremented by one if multiple installations of WebSphere are present on the system.

If neither of these default port numbers works, examine the following file for your SIMULIA Execution Engine:

```
<websphere_install_directory>\AppServer\profiles\<profileName>\logs\
AboutThisProfile.txt
```

The port number settings for your Administrative console are listed within this log file, which is located in the directory shown above for default installations of WebSphere; your directory may differ if you are using a nondefault WebSphere installation.

Manually Configuring the SIMULIA Execution Engine

Once you have initialized the database, you can begin configuring the SIMULIA Execution Engine within the application server.

The SIMULIA Execution Engine is built upon several basic services supplied by commercial products. Each of these services must be installed as a product itself. These services must then be configured (where they interact) to know about each other and to configure the interaction. In this configuration, each product is installed according to its own setup program and is then configured as described in this guide.

Note: WebSphere will not work properly when installed on a computer with a host name that contains underscores or dashes.

Starting WebSphere and Determining Server Port Numbers

Before accessing the WebSphere console, you need to start the application server. Once the server is started, you need to take note of some WebSphere port numbers that are used later in the configuration process.

When WebSphere is installed, it can use nondefault port numbers if there is another WebSphere server on the same computer or if custom port numbers were specified during the installation.

1. Verify that you have initialized your database as described in *Initializing the SIMULIA Execution Engine Database*.
2. Perform one of the following actions to start the WebSphere server:

- Windows: Click the **Start** button, point to **All Programs**/**IBM WebSphere**/**IBM WebSphere Application Server V8.5**/**Profiles**/**AppSrv01**, and then click **Start the server**.
- UNIX/Linux: Navigate to the `<websphere_install_directory>/AppServer/bin` directory and execute the `./startServer.sh server1` command.

The application server is started.

3. Once the server is running, perform one of the following actions:
   - Windows: Click the **Start** button, point to **All Programs**/**IBM WebSphere**/**IBM WebSphere Application Server V8.5**/**Profiles**/**AppSrv01**, and then click **Administrative console**.
   - UNIX/Linux: Open a web browser and navigate to the following page:

     ```
     http://localhost:portnumber/ibm/console
     ```

     For more information on determining the port number for your WebSphere installation, see *About WebSphere URL Port Numbers*.
4. In the **User ID** text box, type a login name.

   You can use any login name you want. However, for consistency this procedure will use the name *seeadmin*.

   When security is turned on later in this configuration procedure, you will need a user ID and password that is valid for the WebSphere server.
5. Click **Log in**.

   The full console appears.
6. On the left side of the console, click **Servers**.
7. Click **Server Types**.

   Additional options appear.
8. Click **WebSphere application servers**.
9. On the right side of the console, click **server1**.
10. On the right side of the console, click **Ports** in the **Communications** area.

    The **Ports** screen appears.

11. Make note of the following port numbers (they will be needed later in the configuration process):

    `BOOTSTRAP_ADDRESS`

    `SIB_ENDPOINT_ADDRESS`

    The bootstrap address (the default setting is 2809) is used by the SIMULIA Execution Engine clients in the Connection Profile configuration (as described in *Creating the Connection Profile File*). The SIB endpoint address (the default setting is 7276) is used when configuring JMS resources.

Creating a J2C Authentication Alias for JDBC Datasources

Before you create the JDBC Providers and data sources, you must create J2C authentication aliases for the database to which you want to connect.

1. On the left side of the console, click **Security**.

   Additional options appear.

2. Click **Global security**.

   The **Global security** screen appears.

3. In the **Authentication** area on the right side of the console, expand **Java Authentication and Authorization Service**.

   Additional options appear.

4. Click **J2C authentication data**.

   The **JAAS - J2C authentication data** screen appears.

5. Click **New**.
6. In the **Alias** text box, type any alias that you want to use.

   For example, you can use `SEEOracleAuth` or `SEEDB2Auth`, depending on your database.

7. In the **User ID** text box, type the database user name that owns the SIMULIA Execution Engine database tables.
8. In the **Password** text box, type the database user password.
9. (optional) Type a description in the corresponding text box.
10. Click **OK**.

Adding JDBC Providers

Once you have specified the user that will access your database, you need to add JDBC providers for the database.

About the Driver File for Oracle

Before creating JDBC Providers for Oracle and data sources, you must make the Oracle 11gR2 JDBC driver (`ojdbc6.jar`) available on the computer where WebSphere is running.

This file is available under the `jdbc\lib` directory of your Oracle 11gR2 database instance (i.e., `<ORACLE_HOME>\jdbc\lib`). For example, your directory path should resemble the following:

Windows: `C:\oracle\product\11.2.0.1\db_1\jdbc\lib\` or `C:\oracle11gR2\ora11gR2\jdbc\lib`

Linux: `/opt/oracle/product/11.2.0/db_1/jdbc/lib/`

If Oracle is running on a different computer than WebSphere, this file needs to be copied to the computer running WebSphere. It is recommended that you copy this file to the *<websphere_install_directory>*`/AppServer/lib/` directory.

Adding JDBC Providers for Oracle or DB2

You must add JDBC Providers to WebSphere for the database. The instructions in this section apply to both Oracle and DB2 databases.

1. On the left side of the console, click **Resources**.

   Additional options appear.

2. Expand **JDBC**.
3. Click **JDBC providers**.
4. From the **All scopes** list, select **Node=**`<server_name><node_number>`**, Server=server1**.
5. Click **New**.

   The **Create a new JDBC Provider** screen appears.

6. From the **Database type** list, select **Oracle** or **DB2**.
7. In the **Provider type** list, select **Oracle JDBC Driver** or **DB2 Universal JDBC Driver Provider**.
8. From the **Implementation type** list, select **XA data source**.

9. Click **Next**.
10. Specify the path to the JDBC driver file, as follows:
    - **Oracle:** In the **Directory location** text box, type the path to the `ojdbc6.jar` file you downloaded in *About the Driver File for Oracle*. Be sure to use forward slashes (/) in your path, even on a Windows system.
    - **DB2:** In the first **Directory location for** text box, type the path to the JDBC driver in the `java/` subdirectory of your DB2 installation directory.

      Be sure to use forward slashes (/), even on a Windows system. For example, if DB2 is installed in `C:\Program Files\IBM\SQLLIB` on Windows, the path you enter should be the following:

      ```
      C:/progra~1/ibm/sqllib/java
      ```

      On Linux, your directory will resemble the following:

      ```
      /opt/IBM/SQLLIB/java
      ```

      If DB2 is not installed on the same computer as WebSphere, you need to install or copy the DB2 client on the WebSphere system and point to the files in this local, client installation. This setup is especially necessary if WebSphere and DB2 are running on different operating systems (specifically when one is running on Windows and the other is running on Linux).

11. Click **Next**.
12. Review the summary.
13. Click **Finish**.
14. For a DB2 database only, do the following:
    a. On the left side of the console, click **Environment**.
    b. Click **Variables**.
    c. Click **New**.
    d. In the **Name** text box, type `DB2UNIVERSAL_JDBC_DRIVER_PATH`.
    e. In the **Value** text box, type the path to the following subdirectory of your SIMULIA Execution Engine installation (depending on your operating system):

       `<see_install_dir>/<os_dir>/reffiles/SMAFIPserver/db/db2/java`

       where `<see_install_dir>` is the base directory of your installation.
    f. Click **OK**.

Creating Datasources

Once you have added the JDBC Providers for your database, you are ready to create the necessary data sources. You will need to create an XADataSource and a non-XADataSource.

Creating the XADataSource for Oracle or DB2

Creating the XADataSource involves specifying connection information for your database and setting some custom options. The instructions in this section apply to both Oracle and DB2 databases, except where noted.

1. If you are running the SIMULIA Execution Engine on Windows, verify that you have updated the Windows Firewall on the computer running the application server and the SIMULIA Execution Engine software.

   For more information, see *Configuring the Windows Firewall*.

2. In the **Name** column on the right side of the console, click one of the following:

   - **Oracle JDBC Driver (XA)**
   - **DB2 Universal JDBC Driver Provider (XA)**

3. In the **Additional Properties** area on the right side of the console, click **Data sources**.

   The **Data sources** screen appears.

4. Click **New**.
5. In the **Data source name** text box, type the following entry:

   ```
   Fiper XA Data Source
   ```

6. In the **JNDI name** text box, type the following entry:

   ```
   fiper/jdbc/XADataSource
   ```

7. Click **Next**.
8. For an Oracle database, do the following:
   a. Enter the following information for the **URL** setting (in the **Value** text box):

      `jdbc:oracle:thin:@`*hostname:port:databasename*

      where:

      `hostname` is the computer name where Oracle is running,

      `port` is the appropriate port number on which the Oracle database is listening, and

`databasename` is the name of the SIMULIA Execution Engine database.

For example, if the hostname is `dbcomputer`, the port number is `1521`, and the database name is `SEE`, the entry would appear as follows:

```
jdbc:oracle:thin:@dbcomputer:1521:SEE
```

b. From the **Data store helper class name** list, select the data store helper that matches the version of Oracle you are using.

9. For a DB2 database, do the following:

a. From the **Driver type** list, verify that **4** is selected.

b. In the **Database name** text box, type the appropriate database name (`SEE`).

c. In the **Server name** text box, type the name of the server that is running DB2.

d. If necessary, in the **Port number** text box, alter the port number on which DB2 is listening. If you do not know this port number, contact your local DB2 administrator.

10. Verify that **Use this data source in container managed persistence (CMP)** is selected.

11. Click **Next**.

12. From the **Authentication alias for XA recovery** list, select the J2C authentication alias you created for your database.

13. From the **Component-managed authentication alias** list, select the J2C authentication alias you created for your database.

14. From the **Mapping-configuration alias** list, select **DefaultPrincipalMapping**.

15. From the **Container-managed authentication alias** list, select the J2C authentication alias you created for your database.

16. Click **Next**.

17. Review the summary.

18. Click **Finish**.

19. On the right side of the console, click **Fiper XA Data Source**.

20. In the **Additional Properties** area on the right side of the console, click **Connection pool properties**.

21. Type the following information in the corresponding text boxes and click **OK**:

Connection timeout: `300`

Maximum connections: `100`

Minimum connections: `50`

22. In the **Additional Properties** area on the right side of the console, click **Custom properties**.

23. Click **New**.

24. Type the following information in the corresponding text boxes:

    Name: `useRRASetEquals`

    Value: `true`

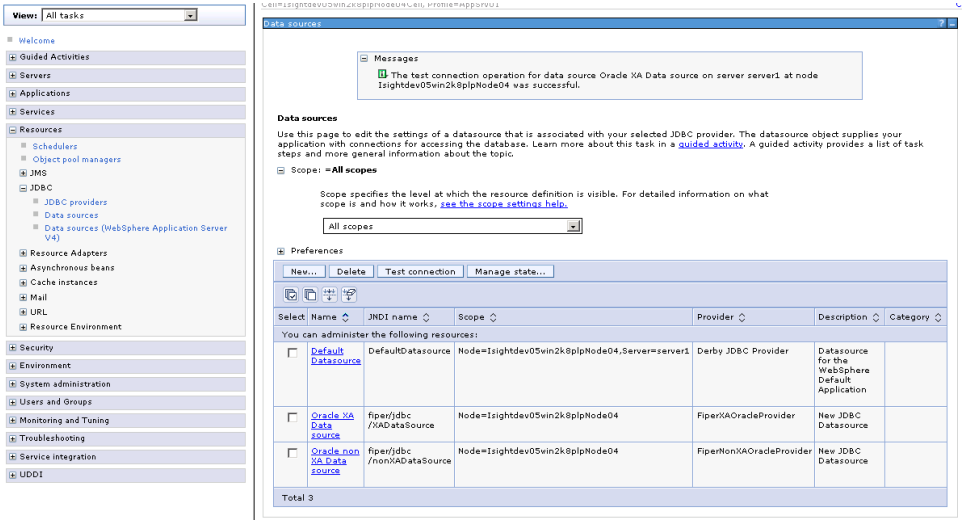
    Description: `See IBM APAR PK75897`

25. Click **OK**.
26. On the left side of the console, click **Data sources**.

    The **Data sources** screen appears.

27. Near the top of the right side of the console, click **Save** to save the configuration.
28. In the **Select** column, click the check box that corresponds to the new data source.
29. Click **Test connection**.

    A message appears, telling you that the test connection was successful.



Note: If your test connection fails, verify that all the information is entered correctly. Confirm that you have disabled the Windows Firewall on the computer running WebSphere. If the Windows Firewall is active, you will not be able to connect to the system running the Oracle database if the database is running on a separate system. For more information, see *Configuring the Windows Firewall*.

30. Verify that the connection was successful.

Creating the Non-XADataSource for Oracle or DB2

As with the XADataSource, creating the non-XADataSource involves specifying connection information for your database and setting some custom options.

1. On the left side of the console, click **JDBC providers**.
2. Verify that **Node=**`<server_name><node_number>`**, Server=server1** is selected.
3. Click **New**.

   The **Create new JDBC provider** screen appears.

4. From the **Database type** list, select **Oracle** or **DB2**.
5. In the **Provider type** list, click one of the following:

   - **Oracle JDBC Driver**
   - **DB2 Universal JDBC Driver Provider**

6. From the **Implementation type** list, select **Connection pool data source**.
7. Click **Next**.
8. In the **Directory location** text box, verify that the path to the JDBC driver file for Oracle or DB2 appears. If it is not present, add it to the text box.

   For Oracle, this is the the `ojdbc6.jar` file you downloaded in *About the Driver File for Oracle*.

   For DB2, this should be the JDBC driver in the `java/` subdirectory of your DB2 installation directory.

9. Click **Next**.
10. Review the summary.
11. Click **Finish**.
12. In the **Name** column on the right side of the console, click **Oracle JDBC Driver** or **DB2 Universal JDBC Provider**.
13. In the **Additional Properties** area on the right side of the console, click **Data sources**.
14. Click **New**.
15. In the **Data source name** text box, type the following entry:

    ```
    Fiper NonXA Data Source
    ```

16. In the **JNDI name** text box, type the following entry:

`fiper/jdbc/nonXADataSource`

17. Click **Next**.
18. For an Oracle database, do the following:
    a. Enter the following information for the **URL** setting (in the **Value** text box):

       `jdbc:oracle:thin:@`*hostname:port:databasename*

       where:

       `hostname` is the computer name where Oracle is running,

       `port` is the appropriate port number on which the Oracle database is listening, and

       `databasename` is the name of the SIMULIA Execution Engine database.

       For example, if the hostname is `dbcomputer`, the port number is `1521`, and the database name is `SEE`, the entry would appear as follows:

       `jdbc:oracle:thin:@dbcomputer:1521:SEE`

    b. From the **Data store helper class name** list, select the data store helper that matches the version of Oracle you are using.
19. For a DB2 database, do the following:
    a. From the **Driver type** list, verify that **4** is selected.
    b. In the **Database name** text box, type the appropriate database name (`SEE`).
    c. In the **Server name** text box, type the name of the server that is running DB2.
    d. If necessary, in the **Port number** text box, alter the port number on which DB2 is listening. If you do not know this port number, contact your local DB2 administrator.
20. Clear (uncheck) **Use this data source in container managed persistence (CMP)**.
21. Click **Next**.
22. From the **Component-managed authentication alias** list, select the J2C authentication alias you created for your database.
23. From the **Mapping-configuration alias** list, select **DefaultPrincipalMapping**.
24. From the **Container-managed authentication alias** list, select the J2C authentication alias you created for your database.
25. Click **Next**.
26. Review the summary.
27. Click **Finish**.
28. On the right side of the console, click **Fiper NonXA Data Source**.
29. In the **Additional Properties** area on the right side of the console, click **Connection pool properties**.

30. Type the following information in the corresponding text boxes:

    Connection timeout: `300`

    Maximum connections: `100`

    Minimum connections: `25`

31. Click **OK**.
32. In the **Additional Properties** area on the right side of the console, click **Custom properties**.
33. Click **New**.
34. Type the following information in the corresponding text boxes:

    Name: `useRRASetEquals`

    Value: `true`

    Description: `See IBM APAR PK75897`

35. Click **OK**.
36. On the left side of the console, click **Data sources**.

    The **Data sources** screen appears.

37. Near the top of the right side of the console, click **Save** link to save the configuration.
38. In the **Select** column, click the check box that corresponds to the new data source.
39. Click **Test connection**.
40. Verify that the connection was successful.

Note: If your test connection fails, verify that all the information is entered correctly. Confirm that you have disabled the Windows Firewall on the computer running WebSphere. If the Windows Firewall is active, you will not be able to connect to the system running the database if the database is running on a separate system. For more information, see *Configuring the Windows Firewall*.

Setting the fiperhome Variable and the Library Options

You need to create a WebSphere variable named `fiperhome` that points to your SIMULIA Execution Engine installation directory and to create a shared library for your configuration.

1. On the left side of the console, click **Environment**.
2. Click **WebSphere variables**.
3. From the **All scopes** list, select:

Cell=<*servername*><*nodenumber*>Cell

4. Click **New**.
5. In the **Name** text box, type `fiperhome`.
6. In the **Value** text box, type the path to the one of the following subdirectories of your SIMULIA Execution Engine installation (depending on your operating system):

   - Windows 64-bit: `<see_install_dir>\win_b64`
   - Linux 64-bit: `<see_install_dir>/linux_a64`

   where `<see_install_dir>` is the base directory of your installation. For example, if you chose the default directory during installation, the value of `fiperhome` should be set to one of the following:

   - Windows 64-bit: `C:\SIMULIA\ExecutionEngine\5.9\win_b64`
   - Linux 64-bit: `/opt/SIMULIA/ExecutionEngine/5.9/linux_a64`

7. Click **OK**.

   The variable is added to the list of defined variables.

8. On the left side of the console (under **Environment**), click **Shared libraries**.
9. From the **All scopes** list, select:

   Cell=<*servername*><*nodenumber*>Cell

10. Click **New**.
11. Type the following information in the corresponding text boxes:

    Name: `fipercommon`

    Classpath: `${fiperhome}/docs/java/SMAFIPutiljni.jar`

    Native Library Path: `${fiperhome}/code/bin`

12. Click **OK**.
13. On the left side of the console (under **Servers**/**Server Types**), click **WebSphere application servers**.
14. On the right side of the console, click **server1**.
15. In the **Server Infrastructure** area on the right side of the console, expand **Java and Process Management**.
16. Click **Class loader**.
17. Click **New**.
18. From the **Class loader order** list, verify that **Class loaded with parent class loader first** is selected.

19. Click **Apply**.
20. In the **Additional Properties** area on the right side of the console, click **Shared library references**.
21. Click **Add**.
22. From the **Library name** list, verify that **fipercommon** is selected.
23. Click **OK**.

Enabling the Startup Beans Service

You need to activate the WebSphere startup beans service.

1. On the left side of the console (under **Servers**/**Server Types**), click **WebSphere application servers**.
2. On the right side of the console, click **server1**.
3. In the **Container Settings** area on the right side of the console, expand **Container Services**.
4. Click **Startup beans service**.
5. Click **Enable service at server startup**.
6. Click **OK**.

JMS Configuration - Configuring Service Integration Bus

You need to create the Service Integration Bus and configure SIMULIA Execution Engine-specific options.

1. On the left side of the console, click **Service integration**.

   Additional options appear.

2. Click **Buses**.

   The **Buses** screen appears.

3. On the right side of the console, click **New**.
4. In the **Enter the name for your new bus** text box, type the following entry:

   ```
   Fiper Bus
   ```

5. Verify that the **Bus security** check box is cleared (unchecked).
6. Click **Next**.
7. Review the confirmation screen.
8. Click **Finish**.

9. On the right side of the console, click **Fiper Bus** in the **Name** column.
10. In the **Default messaging engine high message threshold** text box, type the following entry:

    `200000`

11. Click **Apply**.
12. In the **Topology** area on the right side of the console, click **Bus members**.
13. Click **Add**.
14. Choose the appropriate server from the **Server** list, if necessary.
15. Click **Next**.
16. Verify that **File store** is selected.
17. Click **Next**.

    The Step 1.2 options appear.

18. Perform the following for the Step 1.2 options:
    a. In the **Log size** text box, type `500`.
    b. Verify that **Default log directory path** is selected.
    c. Verify that **Same settings for permanent and temporary stores** is selected.
    d. In the **Minimum permanent store size** text box, type `500`.
    e. Click **Unlimited permanent store size**.
    f. Verify that **Default permanent store directory path** is selected.
19. Click **Next**.

    The Step 1.3 options appear.

20. Click **Next**.
21. Review the summary.
22. Click **Finish**.
23. Near the top of the right side of the console, click **Save** to save the configuration.

Creating JMS Destinations

You need to create several SIMULIA Execution Engine-specific JMS Destinations within WebSphere.

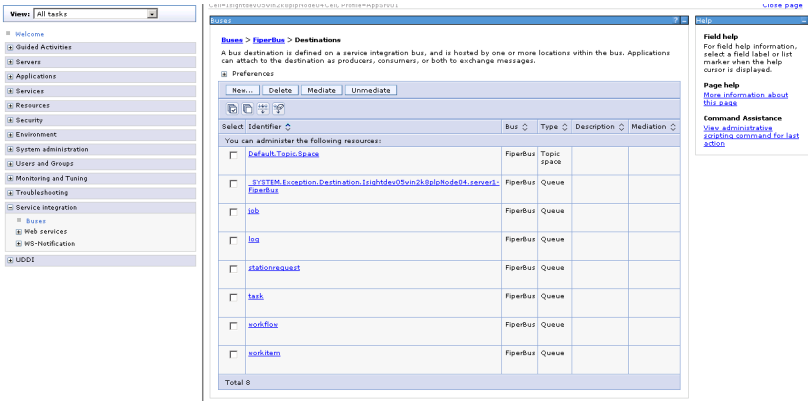
1. Near the top of the right side of the console, click **Fiper Bus**.
2. In the **Destination resources** area on the right side of the screen, click **Destinations**.

   The **Destinations** screen appears.

3. Click **New**.
4. Verify that **Queue** is selected as the destination type.
5. Click **Next**.
6. In the **Identifier** text box, type the following entry: `job`
7. Click **Next**.
8. From the **Bus member** list, verify that **Node=**`<server_name><node_number>`**:Server=server1** is selected.
9. Click **Next**.
10. Click **Finish**.
11. In the **Identifier** column of the **Destinations** table, click **job**.
12. From the **Maximum reliability** list, select **Assured persistent**. (The **Default reliability** setting does not matter and does not need to be set to any particular value.)
13. Click **OK**.
14. Repeat step 3 through step 13 for each of the following **Identifier** names (the other settings are the same for each identifier):

    - `log`
    - `stationrequest`
    - `task`
    - `workflow`
    - `workitem`

    Your list of destinations should appear similar to the following:



15. In the **Identifier** column, click **Default.Topic.Space**.
16. From the **Maximum reliability** list, select **Assured persistent**.
17. Click **OK**.
18. Save the configuration.

Creating Queues

You need to create several SIMULIA Execution Engine-specific JMS Queues within WebSphere.

1. On the left side of the console (under **Resources**), expand **JMS**.
2. Click Queues.
3. From the **All scopes** list, select **Cell=**`<server_name><node_number>`**Cell**.
4. Click **New**.

   The **Queue Wizard** appears.

5. Verify that **Default messaging provider** is selected.
6. Click **OK**.
7. Type the following information in the corresponding text boxes:

   Name: `fiper job`

   JNDI Name: `fiper/jms/job`

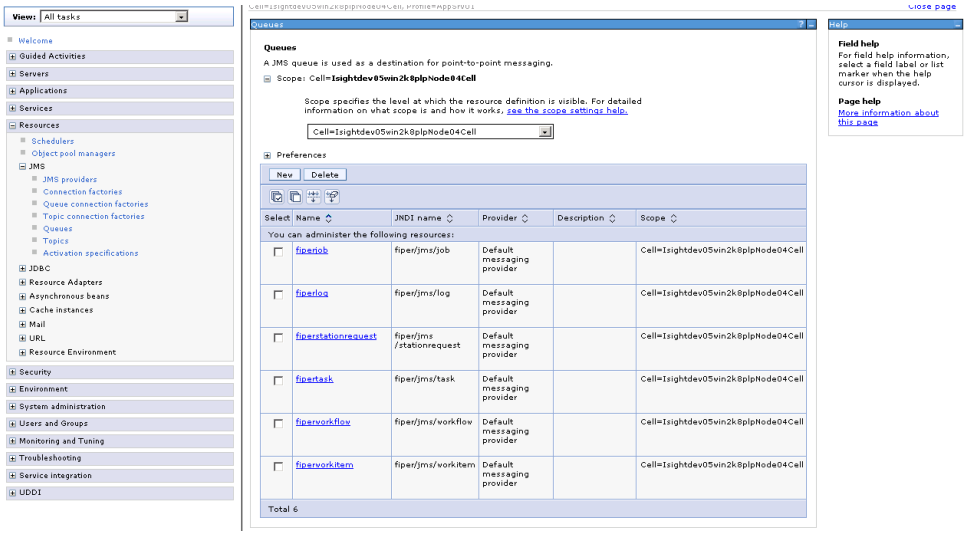
8. From the **Bus name** list, select **Fiper Bus**.
9. From the **Queue name** list, select **job**.
10. Click **OK**.
11. Repeat step 4 through step 10 for each of the following queues, replacing the information shown below in the corresponding steps:

    Create the following queues:

Name: `fiper log` **JNDI Name**: `fiper/jms/log` **Queue Name**: `log`
Name: `fiper stationrequest` **JNDI Name**: `fiper/jms/stationrequest` **Queue Name**: `stationrequest`

Name: `fiper task` **JNDI Name**: `fiper/jms/task` **Queue Name**: `task`
Name: `fiper workflow` **JNDI Name**: `fiper/jms/workflow` **Queue Name**: `workflow`
Name: `fiper workitem` **JNDI Name**: `fiper/jms/workitem` **Queue Name**: `workitem`

Your list of queues should appear similar to the following:



12. Save the configuration.

Creating Topics and the Connection Factory

You need to create a JMS Topic that is specific to your SIMULIA Execution Engine stations, and you need to define a Connection Factory for your installation.

1. Under **Resources**/**JMS** on the left side of the console, click **Topics**.
2. From the **All scopes** list, select **Cell=**`<server_name><node_number>`**Cell**.
3. Click **New**.
4. Verify that **Default messaging provider** is selected.
5. Click **OK**.
6. Type the following information in the corresponding text boxes to create a new Topic (any settings not listed below should be left at the default WebSphere value):

   Name: `fiper station`

   JNDI name: `fiper/jms/station`

   Topic name: `station`
7. From the **Bus name** list, select **Fiper Bus**.
8. From the **Topic space** list, select **Default.Topic.Space**.
9. Click **OK**.

   The topic is created.
10. Click **New**.
11. Verify that the **Default messaging provider** is selected.
12. Click **OK**.
13. Type the following information in the corresponding text boxes to create a new topic (any settings not listed below should be left at the default WebSphere value):

    Name: `job monitor`

    JNDI name: `fiper/jms/jobmonitor`

    Topic name: `jobmonitor`
14. From the **Bus** name list, select **Fiper Bus**.
15. From the **Topic** space list, select **Default.Topic.Space**.
16. Click **OK**.
17. Under **Resources**/**JMS** on the left side of the console, click Connection factories.

    Important: Be careful to select **Connection factories** and not other similar options located near it.
18. From the **All scopes** list, select **Cell=**`<server_name><node_number>`**Cell**.
19. Click **New**.

20. Verify that **Default messaging provider** is selected.
21. Click **OK**.
22. Type the following information in the corresponding text boxes to create a new Connection Factory:

    Name: `Fiper CF`

    JNDI name: `fiper/jms/connectionfactory`
23. From the **Bus name** list, select **Fiper Bus**.
24. In the **Target inbound transport chain** text box, type the following entry:

    `InboundBasicMessaging`
25. In the **Provider endpoints** text box, type the following entry, substituting the server's host name and the endpoint address noted in *Starting WebSphere and Determining Server Port Numbers*.

    For example:

    ```
    seecomputer:7276:BootstrapBasicMessaging
    ```

    Note: This port number (`7276`) is the `SIB_ENDPOINT_ADDRESS` that you noted previously.
26. From the **Nonpersistent message reliability** list, verify that **Express nonpersistent** is selected .
27. Type the following information in the corresponding text box:

    Client identifier: `fiper`
28. Select `<Node_name>.server1-FiperBus` as the **Durable Subscription** home.
29. From the **Persistent message reliability** list, select **Assured persistent**.
30. From the **Authentication alias for XA recovery** list, selectthe appropriate alias.

    Your option should appear similar to one of the following examples:

    - **acscomputerNode01/fiperOracleAuth**
    - **acscomputerNode01/fiperDB2Auth**

    Note: This setting is necessary if security will be enabled on your SIMULIA Execution Engine, but it will not impact a non-secure SIMULIA Execution Engine.

31. From the **Mapping-configuration alias** list, select **DefaultPrincipalMapping**.
32. From the **Container-managed authentication alias** list, select the appropriate alias.

Note: This setting is necessary if security will be enabled on your SIMULIA Execution Engine, but it will not impact a non-secure SIMULIA Execution Engine.

33. Click **Apply**.
34. In the **Additional Properties** area on the right side of the console, click **Connection pool properties**.
35. In the **Connections timeout** text box, verify that **180** appears.
36. Type the following information in the corresponding text boxes:

    Maximum connections: `100`

    Minimum connections: `25`

37. Click **OK**.

Creating the Activation Specifications

You need to create several Activation Specifications for your SIMULIA Execution Engine installation.

1. Under **Resources**/**JMS** on the left side of the console, click **Activation specifications**.
2. From the **All scopes** list, select **Cell=**`<server_name><node_number>`**Cell**.
3. Click **New**.
4. Verify that **Default messaging provider** is selected.
5. Click **OK**.
6. In the **Name** text box, type the following entry:

   ```
   job
   ```

7. In the **JNDI name** text box, type the following entry (this setting is case-sensitive):

   ```
   fiper/act/job
   ```

8. From the **Destination type** list, verify that **Queue** is selected.
9. In the **Destination JNDI name** text box, type the following entry:

   ```
   fiper/jms/job
   ```

   This entry represents the JMS topic name that matches the current entry. These settings are case-sensitive, just like the original JNDI name settings.

10. From the **Bus name** list, select **Fiper Bus**.
11. In the **Maximum batch size** text box, verify that **1** appears.
12. In the **Maximum concurrent MDB invocations per endpoint** text box, verify that **10** appears.
13. From the **Authentication alias** list, selectthe appropriate alias. You may need to scroll down to see this option.

    Note: This setting is necessary if security will be enabled on your SIMULIA Execution Engine, but it will not impact a non-secure SIMULIA Execution Engine.

14. Click **OK**.
15. Repeat step 3 through step 14 for each of the following specifications, replacing the information shown below in the corresponding steps (all other settings are the same as described above):

    Create the following activation specifications:

Name: `log` **JNDI name**: `fiper/act/log` **Destination JNDI name**: `fiper/jms/log` **Maximum concurrent MDB invocations per endpoint**: `10`
Name: `stationrequest` **JNDI name**: `fiper/act/stationrequest` **Destination JNDI name**: `fiper/jms/stationrequest` **Maximum concurrent MDB invocations per endpoint**: `10`
Name: `stationrequestlarge` **JNDI name**: `fiper/act/stationrequestlarge` **Destination JNDI name**: `fiper/jms/stationrequest` **Maximum concurrent MDB invocations per endpoint**: `2`
Name: `task` **JNDI name**: `fiper/act/task` **Destination JNDI name**: `fiper/jms/task` **Maximum concurrent MDB invocations per endpoint**: `1`
Name: `task2`

JNDI name: `fiper/act/task2`

Destination JNDI name: `fiper/jms/task`

Maximum concurrent MDB invocations per endpoint: `1`

Name: `task3`

JNDI name: `fiper/act/task3`

Destination JNDI name: `fiper/jms/task`

Maximum concurrent MDB invocations per endpoint: `1`

Name: `workflow`

JNDI name: `fiper/act/workflow`

Destination JNDI name: `fiper/jms/workflow`

Maximum concurrent MDB invocations per endpoint: `10`

Name: `workflowlarge`

JNDI name: `fiper/act/workflowlarge`

Destination JNDI name: `fiper/jms/workflow`

Maximum concurrent MDB invocations per endpoint: `2`

Name: `workitem`

JNDI name: `fiper/act/workitem`

Destination JNDI name: `fiper/jms/workitem`

Maximum concurrent MDB invocations per endpoint: `10`

Name: `workitemlarge`

JNDI name: `fiper/act/workitemlarge`

Destination JNDI name: `fiper/jms/workitem`

Maximum concurrent MDB invocations per endpoint: `2`

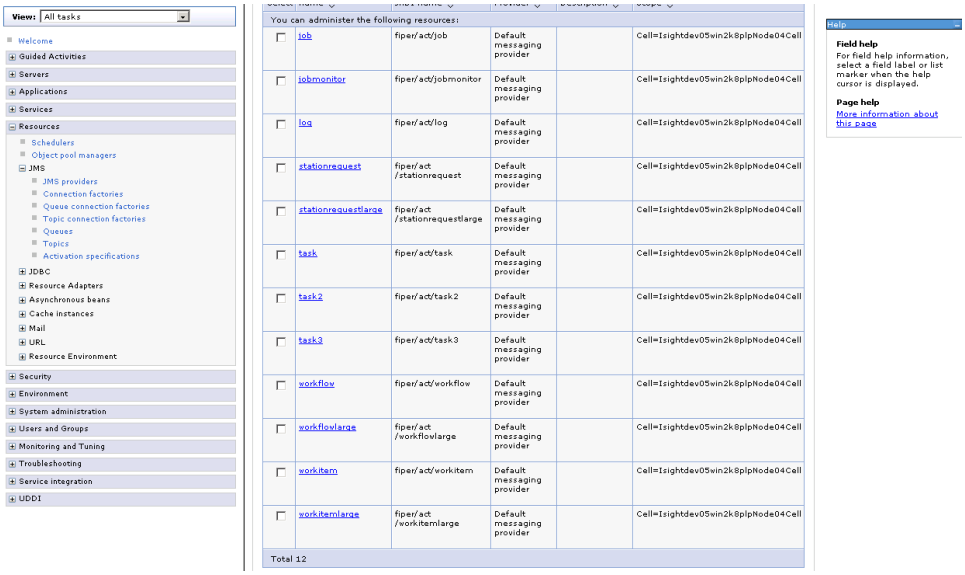
Name: `jobmonitor`

JNDI name: `fiper/act/jobmonitor`

Destination JNDI name: `fiper/jms/jobmonitor`

Maximum concurrent MDB invocations per endpoint: 10

Your list of Activation Specifications should appear similar to the following:



16. Save the configuration.

Configuring the WebSphere JVM

You must configure the JVM configuration and command line parameters for your installation.

1. Under **Servers/Server Types** on the left side of the console, click **WebSphere application servers**.
2. On the right side of the console, click **server1**.
3. In the **Server Infrastructure** area on the right side of the console, expand **Java and Process Management**.
4. Click **Process definition**.
5. In the **Additional Properties** area on the right side of the console, click **Logging and tracing**.
6. Click **JVM Logs**.
7. In the **System.out/Log File Rotation** area, verify that the **File Size** check box is selected.
8. In the **Maximum Size** text box, type `10`.
9. In the **Maximum Number of Historical Log Files** text box, type `10`.
10. In the **System.err/Log File Rotation** area, verify that the **File Size** check box is selected.
11. In the **Maximum Number of Historical Log Files** text box, type `10`.

12. Click **OK**.
13. Near the top of the right side of the console, click **Process definition**.
14. In the **Additional Properties** area of the right side of the console, click **Java Virtual Machine**.
15. Type the following information in the corresponding text boxes:

    Initial Heap Size: `256`

    (64-bit) **Maximum Heap Size**: `2800`
16. Click **Apply**.
17. In the **Additional Properties** area on the right side of the console, click **Custom properties**.
18. Click **New**.
19. In the **Name** text box, type the following entry:

    `client.encoding.override`
20. In the **Value** text box, type the following entry, then click **OK**:

    `UTF-8`
21. Click **New**.
22. In the **Name** text box, type the following entry:

    `fiper.system.parmfile`
23. In the **Value** text box, enter the following path and filename, then click **OK**:

    Windows: `<see_install_dir>\config\acs.properties`

    (or `${fiperhome}\..\config\acs.properties`)

    Linux: `<see_install_dir>/config/acs.properties`

    (or `${fiperhome}/../config/acs.properties`)
24. Click **New**.
25. In the **Name** text box, type the following entry:

    `fiper.system.configpath`
26. In the **Value** text box, enter the following path, then click **OK**:

    `<see_install_dir>/config`

    (or `${fiperhome}/../config`)

    On Windows, use a back slash “\” in the entry (instead of a forward slash “/”).

27. Click **New**.
28. In the **Name** text box, type the following entry:

    `fiper.system.installpath`
29. In the **Value** text box, enter the following path, then click **OK**:

    `${fiperhome}`
30. Click **New**.
31. In the **Name** text box, type the following entry:

    `IBM_HEAPDUMP_OUTOFMEMORY`
32. In the **Value** text box, type the following entry, then click **OK**:

    `false`
33. Click **New**.
34. In the **Name** text box, type the following entry:

    `IBM_JAVADUMP_OUTOFMEMORY`
35. In the **Value** text box, type the following entry, then click **OK**:

    `false`
36. For Linux only, complete the following substeps, then continue. For Windows, skip to the next step.
    a. Near the top of the right side of the console, click **Process definition**.
    b. In the **Additional Properties** area on the right side of the console, click **Environment Entries**.
    c. Click **New**.
    d. In the **Name** text box, enter the following for Linux:
        - `LD_LIBRARY_PATH`
    e. In the **Value** text box, enter:
       `${fiperhome}/code/bin`
    f. Click **OK**.
37. Near the top of the console in the center, click **server1**.
38. In the **Additional Properties** area on the right side of the console (near the bottom), click **Thread pools**.
39. In the **Name** column, click **Default**.
40. Type the following information in the corresponding text boxes:

Minimum Size: `25`

Maximum Size: `50`

Thread inactivity timeout: `30000`

41. Click **Allow thread allocation beyond maximum thread size**.
42. Click **OK**.
43. Near the top of the right side of the console, click **server1**.
44. On the right side of the console, expand **Container Services**.

    Additional options appear.

45. Click **ORB service**.
46. Click **Pass by reference**.
47. Click **OK**.
48. On the right side of the console, expand **Container Services**.

    Additional options appear.

49. Click **Transaction Service**.
50. In the **Total transaction lifetime timeout** text box, type `300`.
51. Click **OK**.
52. Save the configuration.

Setting the DSLS_CONFIG Environment Variable

If you are using DS licensing (not FLEXnet) for the SIMULIA Execution Engine, you must configure an environment variable to allow the application to find the DS licensing client configuration file.

The `DSLS_CONFIG` environment variable must be set to point to the path/location of the `DSLicSrv.txt` client configuration file in your installation. The `DSLicSrv.txt` file contains the server name and port number for the license server software. For more information about the `DSLicSrv.txt` file, see "Configuring Clients" in the *Dassault Systèmes License Server Installation and Configuration Guide* (`DSLS.pdf`).

1. Under **Servers**/**Server Types** on the left side of the console, click **WebSphere application servers**.
2. On the right side of the console, click **server1**.
3. In the **Server Infrastructure** area on the right side of the console, expand **Java and Process Management**.

4. Click **Process definition**.
5. In the **Additional Properties** area on the right side of the console, click **Environment Entries**.

   The breadcrumb path to this page is shown at the top of the console:

   Application servers > **server1** > **Process definition** > **Environment Entries**
6. Click **New** to add a new environment variable, with the following name and value:

   Name: DSLS_CONFIG

   Value: `<see_install_dir>/config/DSLicSrv.txt`

   (or `${fiperhome}/../config/DSLicSrv.txt`)

   This path is the default location where the SIMULIA Execution Engine installer places the `DSLicSrv.txt` file.
7. Click **OK**.
8. Save the configuration.

Deploying the SIMULIA Execution Engine EAR File

You are now ready to deploy the SIMULIA Execution Engine `.ear` file in WebSphere.

1. On the left side of the console, click **Applications**.

   Additional options appear.
2. Click **Application Types**.
3. Click **WebSphere enterprise applications**.
4. On the right side of the console, click **Install**.
5. Verify that **Local file system** is selected.
6. Click **Browse**.
7. Navigate to the location of the `fiper.ear` file.

   This file is located in the following directory:

   <SEE_install_dir>/*<os_dir>*/reffiles/SMAFIPserver/websphere/
8. Click the `fiper.ear` file.
9. Click **Open**.
10. Click **Next**.
11. Click **Detailed**.
12. Click **Next**.

The installation options appear.

13. Click **Step 3**.
14. Perform one of the following steps, based on the type of database you are using:
    - Oracle: From each of the lists in the **Current backend ID** column, select **ORACLE_V9_1**. (This selection is correct for any supported version of Oracle.)
    - DB2: **From each of the lists in the Current backend ID** column, verify that **DB2UDBNT_V8_1 is selected**. (This selection is correct for any supported version of DB2.)
15. Click the lastlink (**Summary**).
16. Click **Finish**.

    A message appears stating that the deployment of the SIMULIA Execution Engine was successful.
17. On the right side of the console, click **Save** to save the configuration.
18. In the **Select** column, click the check box to the left of the **Fiper** entry in the **Name** column.
19. Click **Start**.

    A message appears near the top of the console when the application is successfully started.
20. Perform one of the following steps, based on your desired security settings:
    - If you are *not* enabling SIMULIA Execution Engine security, click the **Logout** link at the top of the console to exit the WebSphere Integrated Solutions Console, and proceed to *Restarting WebSphere with No Security Enabled*.
    - If you are enabling SIMULIA Execution Engine security, proceed to *Enabling Security*.

Installing the WebTop and WebDashboard

The SIMULIA Execution Engine WebTop and WebDashboard are web-based interfaces that give you access to the SIMULIA Execution Engine for model execution and server administration. These features must be installed separately in WebSphere.

Several options are available for installing the SIMULIA Execution Engine WebTop and WebDashboard. These options can help to increase performance and scalability for your SIMULIA Execution Engine. For complete details on these installations, see *Configuring the WebTop or WebDashboard for the SIMULIA Execution Engine*.

Automatically Configuring the SIMULIA Execution Engine

Once you have initialized the database, you can begin configuring the SIMULIA Execution Engine within the application server.

The SIMULIA Execution Engine is built upon several basic services supplied by commercial products. Each of these services must be installed as a product itself. These services must then be configured to know about each other and to configure interaction between the services. To make it easier to configure WebSphere, SIMULIA provides scripts that automatically configure the WebSphere application server.

The WebSphere configuration scripts (`WebSphereScript.py` and `WebSphereDeploy.py`) are located in the following directory:

`<SEE_install_directory>/<operating_system>/reffiles/SMAFIPserver/websphere/deploy`

In addition, the directory contains a name-value file, `params.txt`, that you must update with the details of your installation.

Prerequisites

You must do the following before you can execute the WebSphere configuration scripts:

- Install WebSphere.
- Create a new profile in WebSphere, if necessary.
- Install, configure, and initialize the database (Oracle or DB2).
- Update and rename (if necessary) the following files:
  - `acs.properties`
  - `webtop.properties`
  - `webdashboard.properties`
- Update the `params.txt` name-value file with information about your installation (see *Params.txt Quick Reference*).

Params.txt Quick Reference

Before you execute the WebSphere configuration scripts, you must update the `params.txt` name-value file with information about your installation.

You can update the `params.txt` file in the same way that you updated the `station.properties` file (see *Configuring SIMULIA Execution Engine Station*

Properties). The table below provides a quick reference for the properties and examples of possible values. The `params.txt` file also contains information and instructions.

params.txt entry	Comment	Example Value
FIPER_HOME	This entry must include the operating system.	D:\SIMULIA\ExecutionEngine\5.x\win_b64
FIPER_CONF		D:\SIMULIA\ExecutionEngine\5.x\config
BSF_ROOT	This entry is for internal use only (leave it blank).	
DB_USER	The User ID that owns the DB2 schema. For DB2, the user must be a user already known to the system. For Oracle, provide the details for the user who created/owns the SEE database schema. For Oracle, the username/password is valid if the username/password used to create the database is the same as the username/password used to access it.	fiperacs
DB_USER_PW	The password for the User ID that owns the DB2 schema. For Oracle, provide the password for the user who created/owns the SEE database schema.	
DB_TYPE	Oracle or DB2.	
DB2_SCHEMA	Ignored for Oracle.	FIPER
DB2_HOST	Ignored for Oracle.	localhost
DB2_PORT	Ignored for Oracle.	50000
SEE_USER	WebSphere console logon.	seeadmin
SEE_USER_PW	The password for the SEE user.	
WAS_HOST	The fully qualified name of the computer running WebSphere.	computer.name.xxx.com
SERVER_NAME	An arbitrary name used in the property files. Typically, you should leave this entry blank because the `fiper.conf` folder contains the `acs.properties`, `webtop.properties`, and `webdashboard.properties` files. However, you can modify the file names to `acs-see1.properties`, etc. If you change the file names, you	

params.txt entry	Comment	Example Value
	must set the value of `SERVER_NAME` to `see1`. In addition, if you change the name for one of the files, you must also change the other two file names to have the same suffix.	
ORACLE_HOME	The local Oracle DB install or the client for the JDBC driver.	E:\Oracle\ora11g
ORACLE_SID	The Oracle instance name. The instance name is typically set when Oracle is installed.	
ORACLE_HOST	The name of the Oracle host, relative to the WAS_HOST.	ora.bar.xxx.com
ORACLE_PORT	The Oracle port number.	1521
WEBTOP_INSTALL	This entry indicates whether or not to install the `Webtop.war` file. True for yes; blank for no.	true
WEBDASH_INSTALL	This entry indicates whether or not to install the Web Dashboard. True for yes; blank for no.	true
B2B_INSTALL	Currently ignored, but the entry must be present (leave it blank).	
ACS_SECURITY	Currently ignored, but the entry must be present. Leave the value set to `1`.	1

Executing the Scripts

You can configure WebSphere automatically by executing two scripts.

Before you begin: You must have administrator privileges to execute the scripts. In addition, if you are executing the scripts on a Linux operating system, you will enter `wsadmin.sh` instead of `wsadmin.bat`. The rest of the command segments for both Windows and Linux are switches, and forward slashes are required.

1. From a command prompt, update and run the following to execute the `WebSphereScript.py` script:

```
<WebSphere installation>\AppServer\bin>wsadmin.bat -lang
jython -profileName <profile_name> -conntype SOAP -user <user
 name> -password <password> -profile

"<SEE_install_directory>/<operating_system>/reffiles/websphere/deploy/WebSphereConfigProcs.py"
```

```
-f
"<SEE_install_directory>/<operating_system>/reffiles/websphere/deploy/WebSphereScript.py"


"<SEE_install_directory>/<operating_system>/reffiles/websphere/deploy/params.txt"
```

The above single command configures your WebSphere profile as required by the SIMULIA Execution Engine.

2. From a command prompt, update and run the following to execute the `WebSphereDeploy.py` script:

```
<WebSphere installation>\AppServer\bin>wsadmin.bat -lang
jython -profileName <profile_name> -conntype SOAP -user <user
 name> -password <password> -profile

"<SEE_install_directory>/<operating_system>/reffiles/websphere/deploy/WebSphereConfigProcs.py"


-f
"<SEE_install_directory>/<operating_system>/reffiles/websphere/deploy/WebSphereDeploy.py"


"<SEE_install_directory>/<operating_system>/reffiles/websphere/deploy/params.txt"
```

The above command deploys the SIMULIA Execution Engine war/ear files to your WebSphere profile.

3. Continue with the remaining topics in *Configuring WebSphere*.

Limitations

There are some limitations to keep in mind when you use the execution scripts to configure WebSphere automatically.

The WebSphere configuration scripts do not start the WebSphere application server automatically. In addition, the scripts do not carry out the following tasks:

- Installing WebSphere.
- Creating WebSphere profiles, if necessary.
- Installing, configuring, and initializing the Oracle or the DB2 database.
- Enabling and configuring security.
- Configuring the federation (B2B) environment.

- Updating the property files in the SIMULIA Execution Engine installation configuration directory (`acs.properties`, `webtop.properties`, and `webdashboard.properties`).
- Creating the connection profile.
- Preloading the SIMULIA Execution Engine Library.

Enabling Security

You can run your WebSphere-based SIMULIA Execution Engine with security enabled or disabled. When security is not enabled, a username and password is not necessary when logging into any feature that is connecting to the SIMULIA Execution Engine.

If you do not want to set up security, proceed to *Restarting WebSphere with No Security Enabled*.

The security enabling process for the SIMULIA Execution Engine is divided into the following main steps:

- Configuring LDAP
- Setting global security options.
- Specifying users
- Enabling SIMULIA Execution Engine RunAs security

For complete instructions on how to perform these tasks, see *Specifying the WebSphere Security Settings*. Once these steps are completed, you must restart the application server as described in *Restarting WebSphere with Security Enabled*.

Note: There are other aspects related to security, such as enabling the SIMULIA Execution Engine station security feature (Run-As). Although these are discussed in *Configuring Security*, they are not necessary for setting up SIMULIA Execution Engine security.

Restarting the SIMULIA Execution Engine in WebSphere

During the configuration process, you will be required to stop and restart your SIMULIA Execution Engine server in WebSphere.

About Starting the SIMULIA Execution Engine Server

The SIMULIA Execution Engine is started and initialized when the WebSphere application server is started and when the SIMULIA Execution Engine application deployed on the server is put into the running state. Typically this process is performed automatically when the application server is started.

The application server tools, described in the following sections, should be used to start the server. The server log will show details of the SIMULIA Execution Engine startup and initialization sequence, starting with a message similar to the following, where `<server_name>` is the name of your application server:

```
SIMULIA Execution Engine (SINGLE) starting on server
"<server_name>"
```

This message is followed by a number of initialization messages. When initialization is complete, a message appears similar to the following:

```
SIMULIA Execution Engine server startup completed.
```

The method used for stopping and restarting WebSphere differs based on whether or not you have enabled security as described in *Enabling Security*. Proceed to one of the following topics, based on your security settings:

- *Restarting WebSphere with No Security Enabled*
- *Restarting WebSphere with Security Enabled*

Restarting WebSphere with No Security Enabled

When stopping and restarting a non-secure Websphere installation, it is not necessary to specify a username or password. You need to specify only the stop and start commands.

1. Perform one of the following actions to stop the WebSphere server:
   - **Windows**: Click the **Start** button, point to **All Programs**/**IBM WebSphere**/**IBM WebSphere Application Server V8.5**/**Profiles**/**AppSrv01**, and then click **Stop the server**.
   - **Linux**: Navigate to the <*websphere_install_directory*>`/AppServer/bin` directory and execute the `./stopServer.sh server1` command.

   A message appears when the server stop action is completed.
2. Perform one of the following actions to restart the WebSphere server:

- **Windows**: Click the **Start** button, point to **All Programs**/**IBM WebSphere**/**IBM WebSphere Application Server V8.5**/**Profiles**/**AppSrv01**, and then click **Start the server**.
- **Linux**: Navigate to the <*websphere_install_directory*>`/AppServer/bin` directory and execute the `./startServer.sh server1` command.

A message appears when the server start action is completed.

Restarting WebSphere with Security Enabled

Stopping and restarting Websphere when security is enabled involves typing specific commands, including the WebSphere username and password, into a Command Prompt window (Windows) or terminal window (Linux).

1. If necessary, open a Command Prompt window (Windows) or terminal window (Linux).
2. Navigate to the `<websphere_install_directory>\bin` directory.
3. Type one of the following commands on a single line to stop the WebSphere server, where `user` is the SIMULIA Execution Engine user's name and `password` is the SIMULIA Execution Engine user's password:
   - Windows: `stopserver server1 -username <user> -password <password>`
   - Linux: `./stopServer.sh server1 -username <user> -password <password>`

   Note: If you fail to supply the user name and password when stopping the server, a dialog box may appear allowing you to specify these items.

   Important: If you have been using a desktop icon or Windows Start menu option to stop the server, it must be updated to specify the user ID and password. For more information, see *Updating Windows Shortcuts for Security Authorization*.

   A message appears when the server stop action is completed.
4. After stopping WebSphere on Linux, you must log out (and preferably reboot your system) and log back in. This action is necessary so that the WebSphere server will recognize changes to the environment in `/etc/profile` made by the SIMULIA Execution Engine installer.

5. Type one of the following commands to restart the WebSphere server (it is not necessary to specify the user name and password when starting the application server):

   - Windows: `startserver server1`
   - Linux: `./startServer.sh server1`

   A message appears when the server start action is completed.

Note: An error in configuring security (such as an incorrect ID or password) can cause the server to fail to start. Because the server also runs the Administrative Console, you will not be able to use the console to fix the configuration error. In this case, edit the `security.xml` file in the `<websphere_install_directory>\AppServer\profiles\<profileName>\config\cells\<`*computer name_NodeCell*`>` directory, and fix the configuration parameters directly in that file. Once completed, you must restart the server. SIMULIA Execution Engine Security is now fully enabled.

About Stopping the SIMULIA Execution Engine Server

If it becomes necessary to stop the SIMULIA Execution Engine, you do so by stopping the WebSphere application server.

It is usually not necessary to stop running jobs before stopping the SIMULIA Execution Engine. Jobs that are running at the time of the shutdown are suspended until the SIMULIA Execution Engine is restarted. However, not all jobs survive a restart—if large file transfers are in progress, they may fail. If work items were assigned to stations and do not complete before the shutdown, they will not resume.

Individual work items in a job that are in progress on stations may continue to run uninterrupted when the server is shutdown. However, the job workflow will not progress until the SIMULIA Execution Engine is restarted.

It is important that stations that are running work items at the time of the SIMULIA Execution Engine shutdown not be terminated or themselves shutdown until after the SIMULIA Execution Engine has been restarted. Terminating a station that is running work while the SIMULIA Execution Engine is shutdown will cause the job to hang or fail. Stations that are running no work at the time of the shutdown can be terminated safely at any time.

The orderly shutdown of the SIMULIA Execution Engine is initiated automatically when the application server is stopped using the normal server administration tools (administration console or shutdown scripts).The server shutdown can take up to 20 minutes as running jobs are suspended and preparations are made for stopping the server. The server log will indicate

that the shutdown has begun with messages similar to the following, where <*server_name*> is the name of your application server:

```
SIMULIA Execution Engine stopping on server "<server_name>"
```

Additional messages are logged during the shutdown process, ending with a message similar to the following:

```
SIMULIA Execution Engine server shutdown completed.
```

It is generally not recommended to "stop" the deployed EAR file in the application server. Instead, the application server process must be shutdown. The startup procedure above can then be used to restart the SIMULIA Execution Engine.

Creating the Connection Profile and Preloading the Library

Before using the SIMULIA Execution Engine, you must configure the connection profile and preload the Isight components to the library.

Before models can be constructed or run, the SIMULIA Execution Engine library must be loaded with the basic system metamodels: components and plug-ins. All components build upon the basic system components, so they must be published in the library before any useful work can be done. Before publishing the library, you must create a connection profile file.

Creating the Connection Profile File

You may need to create a connection profile (if you have not already done so during the installation process) to connect to your SIMULIA Execution Engine.

If you have already created a connection profile for this installation, skip this section and proceed to *Publishing to the Library*.

If you have existing connection profiles from a previous release of SIMULIA Execution Engine, you cannot use these old profiles. They must be deleted and recreated.

1. Perform one of the following actions:
   - Windows: Click the **Start** button, point to **All Programs** / **SIMULIA Execution Engine x.x**, and click **Edit Logon Profile**.
   - Linux: Change directory (`cd`) to the following directory and execute the `editcpr` file (`editcpr.bat` on Windows)

     ```
     <SEE_install_dir>/<os_dir>/code/command/
     ```

The **Profile Editor** appears.

2. In the **Profile name** text box, type the name of the profile.

   This name will appear on the **Connection** list when connecting to the SIMULIA Execution Engine.

3. From the **Server type** list, select **IBM WebSphere 8**.

   Additional options appear when this option is selected.

4. In the **Server name** text box, specify the name of the computer running the SIMULIA Execution Engine.

   If you will be using LSF with your SIMULIA Execution Engine, do not specify the server using its fully qualified domain name. For example, you should enter `seecomputer`, not `seecomputer.yourcompany.com`.

5. If necessary, change the port number in the corresponding text box.

   The port number provided is the default port number for the application server selected. However, based on your network or system configuration, you may need to change this number. This port number should match the `BOOTSTRAP_ADDRESS` port number you located in *Starting WebSphere and Determining Server Port Numbers*.

6. From the **File** menu, select **Save As**.
7. Specify a name and location for the connection profile.

   All connection profiles should be stored in the top level of the SIMULIA Execution Engine installation directory.

8. Click **Save**.

   The connection profile is saved.

9. Close the **Profile Editor** dialog box.

Tip: It is recommended that you make a copy of the resulting `.cpr` file (saved as `servername.cpr`) and make it available for anyone who wants to connect to the SIMULIA Execution Engine.

Publishing to the Library

All components must be preloaded (published) in the library before an Isight gateway can connect to the SIMULIA Execution Engine to create or execute models.

If you have developed/purchased other components or plug-ins that are not part of the standard SIMULIA Execution Engine installation, you will need to publish those items separately. For example, if you are using Isight, you will need to publish the NLPQL, NSGA2, and MOST optimization plug-ins, which are distributed only with that product.

Important: When executing the following procedure on Windows, you may receive an error message if your Windows user name contains certain characters. For more information, see *Resolving Publishing Errors on Windows*.

1. If necessary, open a Command Prompt window (Windows) or terminal window (Linux).
2. Navigate to the following directory (depending on your operating system):

   ```
   <SEE_install_dir>/<os_dir>/code/command/
   ```

3. Type the following command:

   ```
   publishall
   ```

4. Log in to the SIMULIA Execution Engine.

   You may need to create a connection profile for the SIMULIA Execution Engine. For more information, see *Creating the Connection Profile File*.

   Note: If security is enabled on the SIMULIA Execution Engine, the user that logs into the SIMULIA Execution Engine to publish the components must have the administrator role in WebSphere. For more information, see *User Roles*.

5. Verify that no errors appear as the script executes.

   An entry appears for each component or plug-in that is published.

   Note: If you receive an error message similar to "`cannot open super metamodel com.engineous.component.Plugin`", see *Resolving Publishing Errors on Windows*.

   Once you are returned to the command prompt, the library is preloaded.

Understanding the acs.properties File Settings

The `acs.properties` file allows you to customize some settings for your SIMULIA Execution Engine.

The `acs.properties` file is located in the following subdirectory of your installation:

`<SEE_install_dir>/config/`

The following settings are accessible using the `acs.properties` file, and they can be altered using the text editor of your choice. Any changes to this file will not be recognized by the SIMULIA Execution Engine until the application server is restarted.

fiper.acs.name

This setting defines the logical name of the SIMULIA Execution Engine. For single server systems it is the IP host name. For clustered servers it is the cluster (cell) name.

fiper.system.esihome

This setting points to the location of your SIMULIA Execution Engine installation.

fiper.acs.isWindowsService

This setting is used if the DRM is set to `lsf` and the `fiper.security.runas.drm` setting is set to `true`. In this case you must tell the SIMULIA Execution Engine whether the application server is running as a service or not. If the DRM is set to `fiper`, this setting is not used.

fiper.system.filemgr.rootFilePath

This setting defines the directory in which the SIMULIA Execution Engine File Manager will store files used in parameter mapping and large in-model files. This directory must have the following characteristics:

- Must be large enough to hold all the files for all jobs in the SIMULIA Execution Engine database. This size can be in the 10s of GBs or much larger if your users save CAD files, mesh files, or large output reports with per-node details (such files can be 1 GB each). The amount of disk space is probably comparable to the size of the SIMULIA Execution Engine database.
- Should be on a local disk on the SIMULIA Execution Engine server host (the one running WebSphere), though a NAS device is acceptable.
- Can be written by the SIMULIA Execution Engine user ID (the one that is used to start WebSphere).
- Is not a temporary file system. This disk space must never be reclaimed automatically. This rules out `/tmp`, `/var/tmp`, and other such directories that are

cleared during a reboot or during an automatic disk space cleanup.
- Is a separate disk drive or partition, if possible, so that if the disk fills completely it will not affect the rest of the computer.
- Should be a directory uniquely for this use. You should not select an existing directory that is also used for other purposes. For example, use `/export/ExecutionEngineFilemgr` instead of `/export`.

fiper.cluster.controller.location This setting is used with the SIMULIA Execution Engine cluster configuration. It must specify the node and server name where the SIMULIA Execution Engine Controller is deployed. These must be the WebSphere node and server names, not the IP (system) names.

fiper.system.logfilter This setting and its subsettings are used for debugging purposes. They should be used only at the direction of SIMULIA technical support.

fiper.system.temp The location of your SIMULIA Execution Engine temporary directory. Directory names must be separated by forward slash characters (/). The default is the temporary directory from the environment. This setting is `/var/tmp` on Linux. This directory needs to have the following characteristics:

- Must be located in a stable part of your file system. Be sure that it is not automatically “cleaned” while the system is running. If the contents of this directory are deleted, the SIMULIA Execution Engine will not function correctly. This directory can be one that is erased when the machine is rebooted. For more information on which parts of your file system are best suited for this temporary directory, contact your local system administrator.
- Should be on a local disk on the SIMULIA Execution Engine server host (the one running WebSphere). If there is insufficient local storage space, a NAS device can be used, but this setup is not recommended.

fiper.system.drm	This setting and its subsettings allow you to configure the distributed resource management option used by the SIMULIA Execution Engine. You can choose either the built-in mode (**Fiper**) or **LSF**. For more information, see *Using Distributed Resource Management with the SIMULIA Execution Engine*.
fiper.stranded.workitem.interval	This setting defines the interval between checks for work items left stranded in the `PENDING` state due to the lack of availability of a station with the required affinities. Work items that have been in the `PENDING` state for at least half of this time will cause a job log (and system log) `WARNING` message to be generated that indicates the work item is waiting for a station that can process a work item with the work item's affinities.
fiper.system.drm.fipertimelimit	This setting defines the optional maximum time limit for components dispatched using the Fiper DRM option. This setting is optional. If a component's individual maximum time limit is set to a value larger than this value, the component will be dispatched using the LSF DRM option. This option is useful only if both the Fiper and LSF DRMs are enabled. If you specify this option without enabling both Fiper and LSR DRMs, the SIMULIA Execution Engine generates a warning message. The default value is `0`.
fiper.system.bsubpath	This setting is used only when `fiper.system.drm` is set to `lsf`. It must specify the fully qualified path and file name of the LSF **bsub** executable module. For more information on using LSF with the SIMULIA Execution Engine, see *Using Distributed Resource Management with the SIMULIA Execution Engine*.
fiper.system.lsfQueueName	This setting is used if the DRM is set to `lsf`. This value specifies the name of the LSF queue to be used for SIMULIA Execution Engine jobs.
fiper.acl.default.enabled	This setting controls whether SIMULIA Execution Engine users are allowed to set the default permissions on any published object or job. These permissions apply to users for whom no explicit permissions have been set. When set to `false` before the SIMULIA Execution

	Engine is started, any explicitly set default permissions are changed to `none`. The default setting is `true` so new users will not be accidentally locked out of their own new installations of SIMULIA Execution Engine.
fiper.ssl.keystore.file	This setting is used only when using SSL security for federated SIMULIA Execution Engine environments. It must contain the fully qualified file name of the keystore file containing X.509 security certificates.
fiper.ssl.keystore.password	This setting is used only when using SSL security for federated SIMULIA Execution Engine environments. It must contain the keystore password for the keystore file named in the `fiper.ssl.keystore.file` setting. This setting represents only the password for the keystore file itself. It is not the encryption password for any certificates stored in the file.
fiper.b2b.allowWildCards	This setting is used only for federated SIMULIA Execution Engine environments. When set to `true`, access to models in the SIMULIA Execution Engine from remote users will be allowed when wildcards (*) are used to publish shared models. This option allows, for example, all users at a given remote site to access the shared models without naming each individual user when the models are published. When set to `false`, wildcard specifications will not be used to determine access to shared models (e.g., each remote user must be specifically granted access to the shared model).
fiper.b2b.url	This setting is not used with the WebSphere application server and can be ignored.
fiper.security.runas.enabled	This setting is used with the SIMULIA Execution Engine Run-As security feature. For more information, see *Configuring Station (Run-As) Security*.
fiper.security.runas.domain	This setting is used with the SIMULIA Execution Engine Run-As security feature. For more information, see *Configuring Station (Run-As) Security*.

fiper.acs.jms.persistent

This setting specifies whether or not to use persistent JMS messaging between the station and the SIMULIA Execution Engine. The default value is `true`.

For maximum reliability of the system, this setting should be left set to `true`. A value of `false` can result in job failures due to intermittent network connectivity issues or server failures that could otherwise be handled.

There is a corresponding configuration entry in the `station.properties` file, `fiper.station.jms.persistent`, which must also be set to `true` to use persistent JMS messaging.

fiper.acs.continueRunningJobs

This setting specifies whether or not running jobs are to be continued after a planned or unplanned shutdown and restart of the SIMULIA Execution Engine. The default value is `true`.

Generally, this setting should be set to `false` only when there are running jobs that will prevent the SIMULIA Execution Engine from successfully starting up. This is not a normal situation and indicates a flaw in a component, plug-in, or other metadata that should be reported to SIMULIA or the developer of the code if it is not a SIMULIA product. Once the system has started, the setting should be changed back to `true` to be ready for the next time the SIMULIA Execution Engine is restarted.

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.

This arrangement is shown in the figure below.

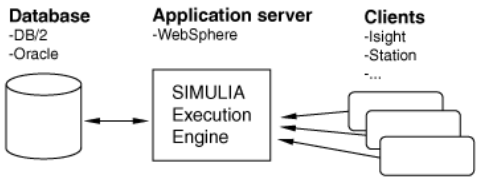


Figure 2: SIMULIA Execution Engine Architecture

The only access to the database is via the WebSphere 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.

Following the procedures described in *About Configuring WebSphere* will produce this type of environment. Any user with physical network access to the SIMULIA Execution Engine can perform any function on the SIMULIA Execution Engine or SIMULIA Execution Engine stations. Administrators should be aware that no credentials are required to access the SIMULIA Execution Engine and execute models, update the SIMULIA Execution Engine library, use the Dashboard or WebDashboard applications, or start or stop stations. In this configuration the application server makes no attempt to authenticate users who log on to the SIMULIA Execution Engine, and all users are considered to have the same user ID (“<anonymous>”).

Any user that can achieve a physical network connection with the SIMULIA Execution Engine can perform these tasks.

About Federation Security

Security configuration for the SIMULIA Execution Engine Federation feature is described in *About Federation Security* in the *SIMULIA Execution Engine Federation (B2B) Guide*.

Configuring SIMULIA Execution Engine Security

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

To activate your SIMULIA Execution Engine to use security, you need to configure WebSphere to use an LDAP server for identifying users. In addition, you must specify which users have access to the SIMULIA Execution Engine as well as their level of access (general user, administrator, or station-only user).

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 name 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 log on is allowed; otherwise, it is rejected. A client that passes this security check is said to be “authenticated,” meaning 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 or WebDashboard, the SIMULIA Execution Engine command-line client, etc. Each of these applications must provide valid user credentials to connect to the SIMULIA Execution Engine and perform any 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, so enabling this feature is a prerequisite to all other SIMULIA Execution Engine security features.

To enable this feature, the administrator configures the application server using the application server supplied tools. The application server is configured to authenticate all incoming connection requests against a particular security back-end infrastructure; usually LDAP is used, but most application servers support many other security protocols. This section describes how to perform this task for the LDAP security system, but your application server documentation should be reviewed for information on all possible options and configurations.

The instructions provided in this section assume the use of an LDAP server for authentication—specifically, Microsoft Active Directory. Other LDAP servers would be configured in a similar manner. WebSphere can also be configured to authenticate with the local computer. In this case, only users that have been added as a local user on the server system will be able to log on to the SIMULIA Execution Engine. This setup may be adequate for small test environments, but it is not suitable for production deployments. Some familiarity with LDAP is helpful to properly configure WebSphere to use LDAP.

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 once you have enabled security. 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.

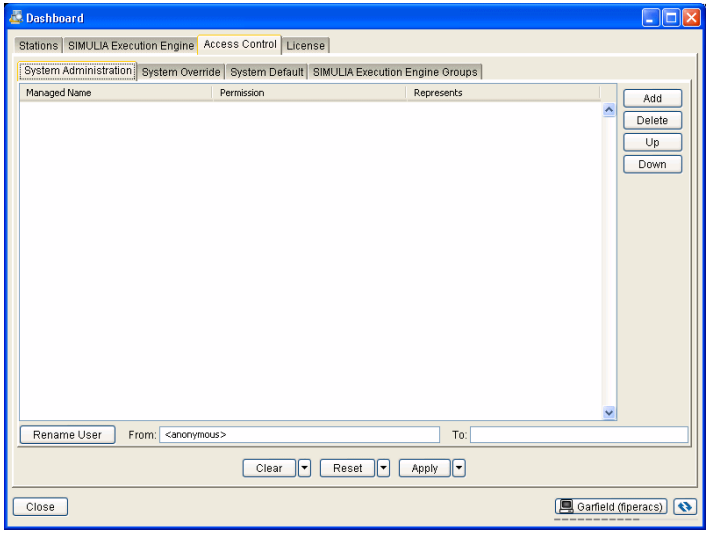


Figure 3: Access Control Tab on the SIMULIA Execution Engine Dashboard

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 (see *About Roles in SIMULIA Execution Engine*). 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 WebSphere 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.

Specifying the WebSphere Security Settings

To activate security for your SIMULIA Execution Engine, you need to update your WebSphere installation to communicate with an LDAP Server, turn on WebSphere's internal security, specify user and group information, and configure the J2EE RunAs security settings.

Specifying Your LDAP Server Settings

The first step in configuring WebSphere for secure client authentication is to configure the connection with an LDAP server. This action defines the security infrastructure against which WebSphere will validate user names and passwords.

1. Determine whether or not you need to alter the LDAP and global security options:
   - If you are enabling security for the first time in an environment that is currently open (non-secure), proceed to step 2.
   - If you are installing the SIMULIA Execution Engine for the first time, proceed to step 2.
   - If you are updating a previous installation of the SIMULIA Execution Engine on an application server instance that already has client authentication enabled, proceed to *Assigning Users to Roles*. All other settings described in *Specifying Your LDAP Server Settings* and *Enabling Global Security in WebSphere* should be unchanged and do not need to be altered.
2. Verify that the WebSphere Application Server is running and that you are logged in to the console.

For more information, see *Starting WebSphere and Determining Server Port Numbers*.

3. On the left side of the console, click **Security**.
4. Click **Global security**.

   The **Global security** screen appears.

5. From the **Available realm definitions** list, select **Standalone LDAP registry**.
6. Click **Set as current**.
7. Click **Configure**.
8. In the **Primary administrative user name** text box, type the user name (for example, `seeadmin`).
9. Verify that **Automatically generated server identity** is selected.
10. From the **Type of LDAP server** list, select the type of server to be used.

    This setting determines the type of LDAP server to be used (for example, Active Directory).

11. In the **Host** text box, type the name of the LDAP server host machine.

    The fully qualified host name is not required, unless the WebSphere host machine needs a fully qualified name to reach the LDAP server. The short host name is adequate if that's all that is needed to `ping` the LDAP server.

12. In the **Port** text box, type the port number of the LDAP server host (for example, 389).
13. In the **Base distinguished name (DN)** text box, specify the necessary information.

    This information represents the starting point in the LDAP tree from which searches should be made for users. Contact your local system administrator for the proper settings.

14. In the **Bind distinguished name (DN)** text box, specify the necessary information.

    This setting identifies a specific user in the LDAP directory that is to be used by the WebSphere server when binding with the LDAP server. This setting may be the same user as used to start the WebSphere server or some other user defined in LDAP. It does not need to be the same as the **Primary administrative user name**. It is specified as a distinguished LDAP name. Contact your local system administrator for the proper settings.

15. In the **Bind password** text box, type the password for the **Bind distinguished name** LDAP user specified in the previous step.

    The remainder of the settings can use their default values.

16. Click **Apply** to save the LDAP settings.

    You may have to scroll down to see this button.

17. Click **Test connection** at the top of the right side of the console.

A message appears if the test was successful. If WebSphere is unable to validate the LDAP settings, carefully check the spelling and case of all entries.

18. Click **OK**.

    You are returned to the **Global security** screen.

Enabling Global Security in WebSphere

After establishing the connection with an LDAP server, you need to set the WebSphere global security option to enable client authentication for the server.

1. Click **Enable administrative security**.
2. Verify that **Enable application security** is selected.
3. Clear (uncheck) **Use Java 2 security to restrict application access to local resources**.
4. From the **Available realm definitions** list, verify that **Standalone LDAP registry** is selected.
5. Click **Apply**.
6. Near the top of the console, click **Save** to save the configuration.
7. At the top of the console, click **Logout**.

   You are returned to the login screen, and you now need to stop and restart the server.

8. Perform one of the following actions to stop the WebSphere server:

   - **Windows**: Click the **Start** button, point to **All Programs**/**IBM WebSphere**/**IBM WebSphere Application Server V8.5**/**Profiles**/**AppSrv01**, and then click **Stop the server**.
   - **Linux**: Navigate to the <*websphere_install_directory*>`/AppServer/bin/` directory, and execute the `./stopServer.sh server1` command.

   Important: Once you restart the server (in the next step), stopping the server again will require you to specify the server user name and password. For more information, see *Restarting WebSphere with Security Enabled*.

   A message appears when the server stop action is completed.

9. Perform one of the following actions to start the WebSphere server:

- **Windows**: Click the **Start** button, point to **All Programs**/**IBM WebSphere**/**IBM WebSphere Application Server V8.5**/**Profiles**/**AppSrv01**, and then click **Start the server**.
- **Linux**: Navigate to the <*websphere_install_directory*>`/AppServer/bin/` directory, and execute the `./startServer.sh server1` command.

A message appears when the server start action is completed.

User Roles

There are three types of SIMULIA Execution Engine users that can be defined, and each type is identified by a security role. You assign each of your users to one or more roles.

About Roles in SIMULIA Execution Engine

Each security role to which a user belongs enables that user to use a particular set of SIMULIA Execution Engine features.

The following security roles are available:

- **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 (see *About SIMULIA Execution Engine Access Control Lists*) or shutdown 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 stations remotely using the Dashboard, WebDashboard, or the Command-Line Client; add and remove federation partners; and perform other restricted administrative functions.
- **fiperstation**. This role provides the ability to run SIMULIA Execution Engine stations. Users with this security role must also have the **fiperuser** security role and are the only users allowed to run stations (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 SIMULIA Execution Engine stations.

For instructions for assigning users or groups to these security roles, see *Assigning Users to Roles*.

Assigning Users to Roles

Once WebSphere security is activated, you can define which users (or groups of users) are allowed access to basic SIMULIA Execution Engine functions. If users are not explicitly given access via security roles, as described below, they will not be able to access the SIMULIA Execution Engine.

For more information on the security roles that are available to users and groups, see *About Roles in SIMULIA Execution Engine*.

Note: This procedure must be repeated when updating to a new release of the SIMULIA Execution Engine.

1. Verify that the WebSphere Application Server is running and that you are logged on to the console.

   For more information, see *Starting WebSphere and Determining Server Port Numbers*.
2. On the left side of the console, click **Applications**.
3. Click **Application Types**.
4. Click **WebSphere enterprise applications**.

   The **Enterprise Applications** screen appears.
5. In the **Name** column on the right side of the console, click **Fiper**.

   The **Fiper** screen appears.
6. In the **Detail Properties** area on the right side of the console, click **Security role to user/group mapping**.

   The mapping information appears.
7. Click the check box adjacent to the **fiperuser** security role.
8. Click **Map Groups**.

   Note: Individual users can be added to a security role by clicking the check box next to the role name and clicking the **Map Users** button.

   The **Map users/groups** screen appears.
9. Click **Search**.

   A list of known groups in the LDAP directory appears.

10. Select a group or multiple groups.
11. Copy the groups to the list on the right side by clicking the  button.

    You can also remove groups using the  button. Contact your local system administrator for more information on the groups that you should be using.
12. Click **OK**.

    You are returned to the **Security role to user/group** mapping screen, and the group you selected is now listed in the **Mapped groups** column.
13. Repeat step 7 through step 12 for the **fiperadmin** and **fiperstation** roles.

    If these roles will use the same group, you can define them at the same time by clicking the check box corresponding to both roles after returning to step 7.
14. Click **OK** to save your changes.

    You are returned to the **Fiper** screen.

Configuring J2EE RunAs Security

You must configure WebSphere J2EE RunAs security before using your security-enabled SIMULIA Execution Engine.

Important: This RunAs security is different from the SIMULIA Execution Engine station Run-As security, which is described in *Configuring Station (Run-As) Security*. While station Run-As security is not required, J2EE RunAs security must be configured or your SIMULIA Execution Engine will not function properly.

1. On the WebSphere console, verify that you are viewing the **Fiper** screen.

   If necessary, follow these steps to access this screen:

   a. On the left side of the console, click **Applications**.
   b. Click **Application Types**, then click **WebSphere enterprise applications**.
   c. In the **Name** column on the right side of the console, click **Fiper**.
2. In the **Detail Properties** area on the right side of the console, click **User RunAs roles**.

   The **User RunAs roles** screen appears.
3. Click the check box to the left of the **fiperuser** role.
4. Type a valid user name and password in the corresponding text boxes near the top of the screen.

Note: This user must either belong to the **fiperuser** global security role or be a member of your user authentication registry (such as LDAP) group that is mapped to the **fiperuser** global security role. For more information on configuring these settings, see *Configuring SIMULIA Execution Engine Security*.

5. Click **Apply**.

   The valid user name appears in the **User name** column.

6. Click **OK**.

   You are returned to the **Fiper** screen.

7. Save the configuration.
8. At the top of the console, click **Logout**.
9. Stop and restart the application server as described in *Restarting WebSphere with Security Enabled*.

   Security has now been enabled for your SIMULIA Execution Engine.

10. Once you restart WebSphere, verify that the SIMULIA Execution Engine application is running successfully:
    a. Access the WebSphere console.
    b. On the left side of the console, click **Applications**.
    c. Click **Application Types**.
    d. Click **WebSphere enterprise applications**.
    e. In the **Application Status** column that corresponds to the **Fiper** application, verify that the ▶ symbol appears.

Updating Windows Shortcuts for Security Authorization

You can update the WebSphere stop command on the Windows Start menu so that it still functions when security is enabled. To update the command, you need to add a user name and password to the shortcut's properties. You do not need to make these changes for the WebSphere start command.

1. Click the **Start** button, point to **All Programs**/**IBM WebSphere**/**IBM WebSphere Application Server V8.5**/**Profiles**/**AppSrv01**.
2. Right-click **Stop the server**, and select **Properties**.

   The **Properties** dialog box appears.

3. Add the following command to the end of the string in the **Target** text box, replacing the `username` and `pw` strings with the correct user name and password for starting WebSphere:

```
-username username -password pw
```

Note: Be sure to leave a space between the existing string and the additional commands added above.

4. Click **OK** to close the **Properties** dialog box.

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 names 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 shutdown.

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 (name and 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.

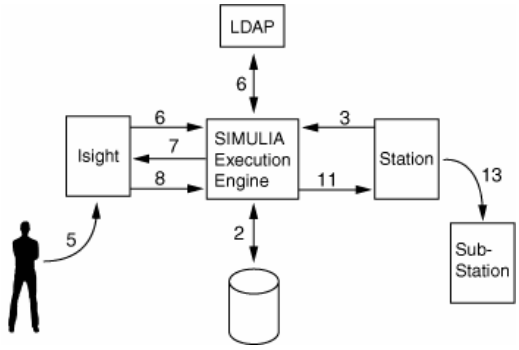


Figure 4: SIMULIA Execution Engine Encryption Process

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 name 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 name to run the SIMULIA Execution Engine station process. The restricted user name 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 name 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 name, 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:

- If the SIMULIA Execution Engine's security realm (or LDAP server) and the SIMULIA Execution Engine station's security realm are different, the extended grid credentials option must be used when a user logs on to the SIMULIA Execution Engine. This option allows the user to enter a user ID and password for the stations that is different from the credentials used to log on to the SIMULIA Execution Engine. The extended grid credentials option can be enabled using the Connection Profile Editor. For more information on using this tool, see *Creating the Connection Profile File*.
- 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.

   For more information, see *Configuring SIMULIA Execution Engine Security*.

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 WebSphere.

   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.
   - *Windows Server 2008*: Navigate to **Start** / **Control Panel** / **System and Maintenance** / **Administrative Tools** / **Local Security Policy**; and from the **Local Security Policy** dialog box, access the **User Rights Assignment** settings under the **Local Policies** option.
   - *Windows 7*: Navigate to **Start** / **Control Panel** / **System and Security** / **Administrative Tools** / **Local Security Policy**; and from the **Local Security Policy** 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 WebSphere for the associated SIMULIA Execution Engine. For more information, see *Restarting WebSphere with Security Enabled*.

    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       required     pam_stack.so service=system-auth
   account    required     pam_stack.so service=system-auth
   password   required     pam_stack.so service=system-auth
   session    required     pam_stack.so service=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. Stop and restart WebSphere for the associated SIMULIA Execution Engine.

   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*.

Configuring the WebTop or WebDashboard for the SIMULIA Execution Engine

This section describes how to set up the SIMULIA Execution Engine WebTop and WebDashboard features.

Determining Your Deployment Strategy

You can deploy the SIMULIA Execution Engine WebTop and WebDashboard in several different ways, based on your network environment and computing resources. In general, you need to decide if the web-based applications will run on the same computer as the SIMULIA Execution Engine or on a different system.

There are three basic choices:

- Install the web-based applications on a separate system from the SIMULIA Execution Engine. This is the recommended architecture.
- Install the web-based applications on the same system as the SIMULIA Execution Engine, in a different WebSphere instance (profile). It is recommended that you use a separate profile for increased scalability and performance.
- Install the web-based applications on the same system and profile as the SIMULIA Execution Engine.

  Although this configuration is possible, in a production environment it is recommended that you use a different profile (or different system) for optimal performance. This simpler configuration may be useful for prototyping or testing, however.

Configuration Steps for Different Architectures

Follow one of the three roadmap procedures below, depending on your choice of J2EE deployment architecture. Each set of steps will direct you to various procedures within this section.

Deploying on a Separate System

Follow this sequence of steps to install and run the WebTop or WebDashboard on a different computer from the one running your SIMULIA Execution Engine.

Note: This configuration is recommended for optimal performance in a production environment.

1. Install the SIMULIA Execution Engine software on the second computer (i.e., the one that will run the WebTop/WebDashboard).
2. Deploy the `webtop.war` file or `webdashboard.war` file within WebSphere. See *Deploying the Web-Based Applications*.
3. Configure various options within WebSphere. See *Configuring the Application on a Different System or Separate Profile*.
4. Check that the installation is successful and that the application is running correctly. See *Verifying the Installation*.

Deploying on the Same System in a Different Profile

Follow this sequence of steps to install and run the WebTop or WebDashboard on the same computer as the SIMULIA Execution Engine, but in a separate WebSphere profile. It is recommended that you use a separate profile for increased scalability and performance.

1. Create a separate profile for the WebTop or WebDashboard. See *Creating a New WebSphere Profile*.
2. Deploy the `webtop.war` file or `webdashboard.war` file within WebSphere. See *Deploying the Web-Based Applications*.
3. Configure various options within WebSphere. See *Configuring the Application on a Different System or Separate Profile*.
4. Check that the installation is successful and that the application is running correctly. See *Verifying the Installation*.

Deploying on the Same System in the Same Profile

Follow this sequence of steps to install and run the WebTop or WebDashboard on the same computer as the SIMULIA Execution Engine and in the same WebSphere profile.

Although this configuration is possible, in a production environment it is recommended that you use a different profile (or different system) for optimal performance. This simpler configuration may be useful for prototyping or testing, however.

1. Deploy the `webtop.war` file or `webdashboard.war` file within WebSphere. See *Deploying the Web-Based Applications*.

2. Configure various options within WebSphere. See *Configuring the Application in the Same Profile*.
3. Check that the installation is successful and that the application is running correctly. See *Verifying the Installation*.

Creating a New WebSphere Profile

If you are installing the WebTop/WebDashboard on the same computer as the SIMULIA Execution Engine, it is recommended that you use a separate profile.

You will need to determine the correct port number for your new profile before accessing the WebSphere Administrative console. You can create a new profile using the profile management tool or by using command line options.

About New Profile Port Numbers

The WebSphere Administrative console is accessed via a web browser using a URL that is specific to the system running WebSphere. You need to determine the correct port number for your new profile before accessing the console, especially if you are using a Linux system.

A sample URL for accessing the console is shown below:

```
http://acscomputer:9061/ibm/console
```

By default, the main WebSphere Administrative console uses the following port numbers: 9060 (non-secure) and 9043 (secure). These port numbers are incremented by one for any additional profiles that are added to the main WebSphere profile.

The port numbers typically used for the Administrative console of a second profile are 9061 or 9044. However, if you have multiple versions of WebSphere installed, these port numbers are incremented and should have been provided on the summary screen during the profile creation process.

You can also examine the profile's `SystemOut.log` file to determine the port numbers used. This file is located in the following directory:

```
<websphere_install_dir>\AppServer\profiles\<profile_name>\logs\
server1
```

Creating a Profile Using the Profile Management Tool

You can create a new profile using the Profile Management Tool, which is provided with a standard WebSphere installation.

1. Perform one of the following actions:
   - **Windows**: Click the **Start** button, point to **All Programs**/**IBM WebSphere**/**IBM WebSphere Application Server V8.5**/**Tools**, and click **Profile Management Tool**.
   - **UNIX/Linux**: Navigate to the <*websphere_install_dir*>`/AppServer/bin` directory, and execute the `./pmt.sh` command.

   The **Profile Management Tool** appears, and a welcome message is displayed.
2. Click **Launch Profile Management Tool**.

   A list of profiles appears.
3. Click **Create**.

   The **Profile Management Tool** wizard appears, displaying the **Environment Selection** screen.
4. Verify that **Application server** is selected.
5. Click **Next**.

   The **Profile Creation Options** screen appears.
6. Verify that **Typical profile creation** is selected.

   Note: This option provides a default name for the new profile. If you want to specify a custom name for the profile, select **Advanced profile creation**. It is recommended that you only change the profile name. Leave all the other options set to their default values, unless changes are specified in this procedure.
7. Click **Next**.

   The **Administrative Security** screen appears.
8. Clear (uncheck) **Enable administrative security**.
9. Click **Next**.

   The **Profile Creation Summary** screen appears, showing you the creation details for the new profile.
10. Review the summary, and make note of the following information:
    - **Location** (the path to the profile)
    - **Profile name**
    - **Administrative console port numbers** (both insecure and secure)

11. Click **Create**.

    The profile is created, and a message appears when the operation is successful.

12. Clear (uncheck) **Launch the First steps console**.
13. Click **Finish** to close the **Profile Management Tool** wizard.

    You are returned to the **Profile Management Tool**, and the new profile is listed along with the existing profiles.

14. Close the **Profile Management Tool**.
15. Start the application server with one of the following actions:

    - **Windows**: Click the **Start** button, point to **All Programs**/**IBM WebSphere**/**IBM WebSphere Application Server V8.5**/**Profiles**/<*new_profile_name*>, and click **Start the server**.
    - **UNIX/Linux**: Navigate to this directory:

    ```
    <websphere_install_dir>/AppServer/profiles/<new_profile_name>/bin/
    ```

    and execute the `./startServer.sh server1` command.

16. Once the server is running, perform one of the following actions:

    - **Windows**: Click the **Start** button, point to **All Programs**/**IBM WebSphere**/**IBM WebSphere Application Server V8.5**/**Profiles**/ <*new_profile_name*>, and click **Administrative console**.
    - **UNIX/Linux**: Use a browser and open the following page:

    ```
    http://localhost:new_profile_port_number/ibm/console
    ```

    For more information on determining the port number for your new WebSphere profile, see *About New Profile Port Numbers*.

17. Type a login name into the **User ID** text box.

    You can use any login name you want. However, for consistency, this procedure will use the name *seeadmin*.

    Note: When security is turned on, you will need a user name and password that are valid for the WebSphere server.

18. Click **Log in**.

Creating a Profile Using Command Line Options

When using WebSphere 8.5, you can create a new profile using command line options provided with the standard WebSphere installation.

1. Open a Command Prompt dialog box (Windows) or terminal window (UNIX/Linux).

   Windows Server 2008: You must open the Command Prompt dialog box using the **Run as administrator** option.

2. Navigate to the following directory:

   ```
   <websphere_install_dir>\bin
   ```

3. Type one of following commands (on a single line), replacing <*new_profile_name*> with the desired name for the new profile (for example, `WebTop`):

   - Windows: `manageprofiles -create -templatePath ..\profiletemplates\default -profileName <new_profile_name>`
   - UNIX/Linux: `manageprofiles.sh -create -templatePath ../profileTemplates/default -profileName <new_profile_name>`

    **Important:** The command arguments are case-sensitive on *all* operating systems.

   Your command should appear similar to the following example:

   ```
   manageprofiles -create -templatePath ..\profiletemplates\
   default -profileName WebTop
   ```

   A message appears when the profile has been successfully created.

4. Navigate to the following directory:

   ```
   <websphere_install_dir>\profiles\<new_profile>\bin
   ```

5. Type one of the following commands:

   - Windows: `startserver server1`
   - UNIX/Linux: `./startServer.sh server1`

6. Once the server is running, access a Web browser and open the following page:

   ```
   http://localhost:new_profile_port_number/ibm/console
   ```

For more information on determining the port number for your new WebSphere profile, see *About New Profile Port Numbers*.

7. Type a login name into the **User ID** text box.

   You can use any login name you want. However, for consistency, this procedure will use the name *seeadmin*.

   Note: When security is turned on, you will need a user name and password that are valid for the WebSphere server.

8. Click **Log in**.

   The full console appears.

Deploying the Web-Based Applications

You must deploy the applications that you want to use within WebSphere. This process involves installing the associated `.war` file as an enterprise application in WebSphere. The `.war` files for WebTop and WebDashboard are included in your SIMULIA Execution Engine installation.

Before you begin: Before starting this procedure, be sure that your SIMULIA Execution Engine installation meets the prerequisites below. You should also decide your deployment architecture beforehand, as described in *Determining Your Deployment Strategy*.

- You have configured the WebSphere application server to include a user called `seeadmin` as described in *About Configuring WebSphere*.
- You have installed, configured, and deployed the SIMULIA Execution Engine on WebSphere as specified in *Configuring WebSphere*.
- You have confirmed that the SIMULIA Execution Engine is running properly.
- If you are installing the web-based applications on a different computer than the one on which the SIMULIA Execution Engine is deployed, be sure that WebSphere is already installed on it. Configure this instance of WebSphere to include a user called `seeadmin`.
- If the WebTop/WebDashboard will run in a different WebSphere profile (on the same computer) than the SIMULIA Execution Engine, verify that you have created a new profile for the WebTop/WebDashboard as described in *Creating a New WebSphere Profile*.

1. Log on to the WebSphere Integrated Solutions Console with one of the following actions:

- Windows: Click the **Start** button, point to **All Programs**/**IBM WebSphere**/**IBM WebSphere Application Server V8.5**/**Profiles**/**AppSrv01**, and then click **Administrative console**.
- Linux: Open a web browser and navigate to the following page:

  `http://localhost:portnumber/ibm/console`

For more details about starting the WebSphere server or logging on, see *Starting WebSphere and Determining Server Port Numbers*.

2. Click **Applications**.

   Additional options appear.
3. Click **New Application**.
4. On the right side of the console, click **New Enterprise Application**.
5. Verify that **Local file system** is selected.
6. Click **Browse**.
7. Navigate to the location of the `.war` files.

   These files are located in the following directory:

   `<SEE_install_dir>/<os_dir>/reffiles/SMAFIPserver/websphere/`
8. Select one of the following files, depending on which application you want to deploy:
   - `webtop.war`
   - `webdashboard.war`
9. Click **Open**.
10. Click **Next**.
11. Verify that **Fast Path** is selected, and click **Next**.
12. Click **Step 3**.
13. In the **Context Root** text box, type one of the following strings (after the "/" that already appears in the text box), based on the application you want to deploy:
    - `webtop`
    - `webdashboard`

    This entry is used to define the URL for accessing the application.
14. Click **Next**.
15. Click **Finish**.

A message appears stating that the installation of the web application was successful.

16. On the right side of the console, click **Save** to save the configuration.

Configuring the Web-Based Applications

To configure the WebTop/WebDashboard application after deploying the `.war` file, you need to specify certain settings in both WebSphere and in the `.properties` file for the application.

This process differs slightly depending on whether you deployed the application on a different system than the SIMULIA Execution Engine, on the same system but in a separate WebSphere profile, or on the same system and in the same profile as the SIMULIA Execution Engine.

Configuring the Application on a Different System or Separate Profile

This section describes how to configure the WebTop/WebDashboard when it is running on a different system than the SIMULIA Execution Engine or on the same system but in a different WebSphere profile.

Setting the fiperhome Variable and the Library Options

You need to create a WebSphere variable that points to your SIMULIA Execution Engine installation directory and create a shared library for your configuration.

1. Define a WebSphere variable named `fiperhome` according to the instructions in *Setting the fiperhome Variable and the Library Options*.
2. Define a shared library (with classpath and native library path) named `fipercommon` according to the instructions in *Setting the fiperhome Variable and the Library Options*.
3. Under **Servers** / **Server Types** on the left side of the console, click **WebSphere application servers**.
4. On the right side of the console, click **server1**.
5. In the **Server Infrastructure** area on the right side of the console, expand **Java and Process Management**.
6. Click **Class loader**.
7. Click **New**.
8. From the **Class loader order** list, verify that **Class loaded with parent class loader first** is selected.
9. Click **Apply**.

10. In the **Additional Properties** area on the right side of the console, click **Shared library references**.
11. Click **Add**.
12. From the **Library name** list, select **fipercommon**.
13. Click **OK**.

Setting JVM Properties

Now you need to specify custom JVM properties for your installation. These settings vary for the WebTop and the WebDashboard.

1. Under **Servers** / **Server Types** on the left side of the console, click **WebSphere application servers**.
2. On the right side of the console, click **server1**.
3. In the **Server Infrastructure** area on the right side of the console, expand **Java and Process Management**.
4. Click **Process definition**.
5. In the **Additional Properties** area, click **Java Virtual Machine**.
6. In the **Additional Properties** area, click **Custom properties**.
7. Click **New**.
8. In the **Name** text box, type one of the following entries based on the application you are configuring:
   - `fiper.webtop.parmfile`
   - `fiper.webdashboard.parmfile`
9. In the **Value** text box, enter one of the following paths based on the application you are configuring:
   - `<see_install_dir>\config\webtop.properties`
   - `<see_install_dir>\config\webdashboard.properties`

   On Linux, use a forward slash “/” in the entry (instead of a back slash “\”).
10. Click **OK**.
11. Click **New**.
12. In the **Name** text box, type the following entry:

```
client.encoding.override
```

13. In the **Value** text box, type the following entry:

```
UTF-8
```

14. Click **OK**.
15. Perform one of the following actions, based on your installation choices:
    - If you are installing the WebTop/WebDashboard on a different machine than the SIMULIA Execution Engine, continue to *Editing the Properties File*.
    - If you are installing the WebTop/WebDashboard and the SIMULIA Execution Engine on the same physical machine but in a different WebSphere profile, perform the following steps:
        a) Click **New**.
        b) In the **Name** text box, type the following entry:
        ```
        com.ibm.websphere.orb.uniqueServerName
        ```
        c) In the **Value** text box, type `true`.
        d) Click **OK**.

Editing the Properties File

You may need to verify or edit the `webtop.properties` file or `webdashboard.properties` file for proper operation of the application.

The WebTop and WebDashboard read certain installation and SIMULIA Execution Engine connection information from a properties file. In some cases this file can be properly configured during the installation, especially if the WebTop/WebDashboard is running on the same machine as the SIMULIA Execution Engine. However, you should verify that this information is correct regardless of your deployment strategy.

1. Near the top of the right side of the WebSphere console, click **Save** to save your configuration.
2. Log out of any WebSphere Integrated Solutions Consoles that you are accessing.
3. On the system running the WebTop/WebDashboard, navigate to the following directory in your SIMULIA Execution Engine installation:
   ```
   <SEE_install_dir>/5.x/config/
   ```
4. Open one of the following files, based on the application you are configuring, in a text editor:

- `webtop.properties`
- `webdashboard.properties`

5. Locate one of the following entries in the file, based on the application you are configuring:
   - `fiper.webtop.acs.cprfile=`
   - `fiper.webdashboard.acs.cprfile=`
6. Verify that the path following the = sign points to the `.cpr` file that will be used to connect to the WebTop/WebDashboard—the `.cpr` file that matches the SIMULIA Execution Engine that the WebTop/WebDashboard will use.

   For example:

   ```
   C:/SIMULIA/ExecutionEngine/5.x/config/seehost.cpr
   ```

   Make sure that this connection profile exists on the system running the WebTop/WebDashboard. For more information, see *Creating the Connection Profile File*. Furthermore, it is recommended that you verify that the connection profile works correctly by connecting to the SIMULIA Execution Engine with any SIMULIA Execution Engine interface (for example, a SIMULIA Execution Engine station or the Dashboard).
7. Locate the following entry in the file:

   ```
   fiper.system.esihome=
   ```
8. Verify that the entry is correctly pointing to the operating system directory of your SIMULIA Execution Engine installation directory, and that the line is uncommented.

   For example:

   ```
   C:/SIMULIA/ExecutionEngine/5.x/win_b64
   ```
9. In the `webtop.properties` file only, you must specify a location for the working (temp) directory for the WebTop application. Find the following lines in this file:

   ```
   #Defines temp work directory for WebTop, must be writeable
   #fiper.system.temp=c:/temp/webtop
   ```

   Remove the `#` in the `fiper.system.temp` line to uncomment it, and replace `c:/temp/webtop` with a path that can serve as a working directory that the WebTop application can write temporary files to. Be sure that the directory/folder that you specify is not subject to automatic cleanup on Linux systems.

If the SIMULIA Execution Engine and WebTop applications are accessing the same file system, it is recommended that you configure different working (temp) directories for each application.

10. Save and close the file.

Restarting the SIMULIA Execution Engine and Application Servers

The SIMULIA Execution Engine and the web-based applications are started and initialized when the WebSphere application server is started and when the SIMULIA Execution Engine application deployed on the server is put into the running state. Typically this process is performed automatically when the application server is started.

For details about stopping and restarting WebSphere, see *Restarting the SIMULIA Execution Engine in WebSphere*.

1. Perform one of the following actions, based on your deployment strategy:

- If you are installing the WebTop/WebDashboard on a different machine than the SIMULIA Execution Engine, restart WebSphere for the WebTop/WebDashboard.
- If you are installing the WebTop/WebDashboard and the SIMULIA Execution Engine on the same physical machine but in different WebSphere profiles, restart both the SIMULIA Execution Engine and WebTop/WebDashboard WebSphere instances.

2. Perform one of the following actions, based on your SIMULIA Execution Engine security level:

- **No security**: The WebTop/WebDashboard configuration is complete. Continue to *Verifying the Installation*.
- **Security enabled**: You need to perform additional security-related actions. Continue to *Exporting the LTPA Key from the SIMULIA Execution Engine*.

Exporting the LTPA Key from the SIMULIA Execution Engine

You need to export the LTPA key from the WebSphere profile running the SIMULIA Execution Engine. This key will be used in the WebSphere profile running the WebTop/WebDashboard.

1. Access the WebSphere Integrated Solutions Console for the system running the SIMULIA Execution Engine.

2. On the left side of the console, click **Security**.

3. Click **Global security**.

4. In the **Authentication** area on the right side of the console, verify that **LTPA** is selected.
5. Click **LTPA**.
6. In the **Cross-cell single sign-on** area, type a password and confirm the password in the corresponding text boxes.

   Be sure to make note of this password. It will be used later when importing keys.
7. In the **Fully qualified key file name** text box, type a key file name.

   You can use any name you like (for example, `seecomputer.key`). Be sure to make note of this file name. It will be used later when importing keys.
8. Click **Export keys**.

   A message appears stating that the export was successful.
9. Copy the exported file to an accessible directory on the system running the WebTop/WebDashboard WebSphere instance.

   If the WebTop/WebDashboard is being installed on a different system from the SIMULIA Execution Engine, you should place this file in a shared directory that can be accessed for the WebTop/WebDashboard system.

   The file should be located in the `<websphere_install_dir>\profiles\AppSrv01` directory on the system running the SIMULIA Execution Engine.
10. Log out of the WebSphere Integrated Solutions Console for the SIMULIA Execution Engine.

Importing the LTPA Key to the Web Application

Now you need to import the LTPA Key, copied from the SIMULIA Execution Engine WebSphere profile, into the WebTop/WebDashboard profile.

1. Access the WebSphere Integrated Solutions Console for the WebTop/WebDashboard.
2. On the left side of the console, click **Security**.
3. Click **Global security**.
4. In the **Authentication** area on the right side of the console, verify that **LTPA** is selected.
5. Click **LTPA**.
6. In the **Cross-cell single sign-on** area, type the password used when you exported the key and confirm the password in the corresponding text boxes.
7. In the **Fully qualified key file name** text box, type the path and name of the key file.

For example, a local path could appear as follows:

```
<websphere_install_directory>
AppServer\profiles\<profileName>\seecomputer.key
```

If you are installing the WebTop/WebDashboard on a separate system from the SIMULIA Execution Engine, your path may appear as follows:

```
\\seecomputer\key_files\seecomputer.key
```

8. Click **Import keys**.

   A message appears stating that the import was successful.

Configuring the LDAP Connection

You need to configure the LDAP connection for the WebTop/WebDashboard to match the configuration that was defined for the SIMULIA Execution Engine.

1. Verify that you are viewing the **Global security** screen.
2. From the **Available realm definitions** list, select **Standalone LDAP registry**.
3. Click **Set as current**.
4. Click **Configure**.
5. In the **Primary administrative user name** text box, type the user name (for example, `seeadmin`).
6. Verify that **Automatically generated server identity** is selected.
7. From the **Type of LDAP server** list, select the type of server to be used.

   This setting determines the type of LDAP server to be used (for example, Active Directory).
8. In the **Host** text box, type the name of the LDAP server host.

   Important: The name of the server, as specified in this text box, must exactly match the same entry on the system running the SIMULIA Execution Engine. For example, if the SIMULIA Execution Engine system specifies a server called `ldap_server.domain.com`, the WebDashboard setting must match it, including the domain name.
9. In the **Base distinguished name (DN)** text box, specify the necessary information.

   This information represents the starting point in the LDAP tree from which searches should be made for users. Contact your local system administrator for the proper settings.

10. In the **Bind distinguished name (DN)** text box, specify the necessary information.

    This setting identifies a specific user in the LDAP directory that is to be used by the WebSphere server when binding with the LDAP server. This setting may be the same user as used to start the WebSphere server or some other user defined in LDAP. It is specified as a distinguished LDAP name. Contact your local system administrator for the proper settings.

11. In the **Bind password** text box, type the password for this user.

    This setting is the password for the SIMULIA Execution Engine user you created previously. The remainder of the settings can be left as is.

12. Click **Apply** to save the LDAP settings.

    You may have to scroll down to see this button.

13. At the top of the right side of the console, click **Test connection**.

    A message appears if the test was successful. If WebSphere is unable to validate the LDAP settings, carefully check the spelling and case of all entries.

14. Click **OK**.

    You are returned to the **Global security** screen.

Setting Global Security Options and Assigning the User Role

After establishing the connection with an LDAP server, you need to set the WebSphere global security option to enable client authentication for the applications.

In addition, you need to define which users (or groups of users) are allowed access to the web-based applications. If users are not explicitly given access via security roles, as described below, they will not be able to access the applications.

1. Click **Enable administrative security**.
2. Verify that **Enable application security** is selected.
3. Clear (uncheck) **Use Java 2 security to restrict application access to local resources**.
4. From the **Available realm definitions** list, verify that **Standalone LDAP registry** is selected.
5. Click **Apply**.
6. On the left side of the console (under **Applications** / **Application Types**), click **WebSphere enterprise applications**.
7. In the **Name** column, click one of the following links, based on the application you are configuring:

- **webtop_war**
- **webdashboard_war**

8. In the **Detail Properties** area on the right side of the console, click **Security role to user/group mapping**.

   The mapping information appears.

9. Click the check box to the left of the **fiperuser** security role.
10. Click **Map Groups**.

    Note: Individual users can be added to a security role by clicking the check box next to the role name and clicking the **Map Users** button.

    The **Map users/groups** screen appears.

11. Click **Search**.

    A list of known groups in the LDAP directory appears.

12. Select a group or multiple groups.
13. Copy the groups to the list on the right side by clicking the  button.

    You can also remove groups using the  button. Contact your local system administrator for more information on the groups that you should be using.

14. Click **OK**.

    You are returned to the **Security role to user/group mapping** screen, and the group you selected is now listed in the **Mapped groups** column.

15. For the WebDashboard only, repeat step 9 through step 14 for the **fiperadmin** role.

    Note: Groups mapped to the **fiperadmin** role have access to more WebDashboard features than those groups mapped to the **fiperuser** role. For more information, see *Using the WebDashboard*.

16. Click **OK**.

Exchanging the SSL Certificate

The final step in setting up security for your application is to exchange the SSL certificate, which involves instructing WebSphere to automatically retrieve host, port, and alias settings from the signer certificate.

1. On the left side of the console (under **Security**), click **SSL certificate and key management**.
2. In the **Related Items** area on the right side of the console, click **Key stores and certificates**.
3. In the **Name** column, click **NodeDefaultTrustStore**.
4. In the **Additional Properties** area, click **Signer certificates**.
5. Click **Retrieve from port**.
6. Type the following information in the corresponding text boxes:

   Host: The host name of the SIMULIA Execution Engine computer.

   Port: The secure port number of the SIMULIA Execution Engine WebSphere instance. For more information on determining this port number, see *Starting WebSphere and Determining Server Port Numbers*.

   Alias: The host name of the SIMULIA Execution Engine computer.
7. Click **Retrieve signer information**.

   Information appears near the bottom of the right side of the console.
8. Click **OK**.
9. Save the configuration.
10. Log out of the WebSphere Integrated Solutions Console.
11. Restart the WebSphere server that is running the WebTop/WebDashboard.

    For more information, see *Restarting WebSphere with Security Enabled*.

    The configuration is complete.

Configuring the Application in the Same Profile

This section describes how to configure the WebTop or WebDashboard when it is running on the same system as the SIMULIA Execution Engine and in the same WebSphere profile.

Note: It is recommended that you run the WebTop/WebDashboard on a separate system than the one running the SIMULIA Execution Engine.

Setting JVM Properties for the Application

You must specify custom JVM properties for your installation, which vary for the WebTop and WebDashboard.

1. On the left side of the console, click **Servers**.
2. Click **Server Types**.

   Additional options appear.

3. Click **WebSphere application servers**.
4. On the right side of the console, click **server1**.
5. In the Server Infrastructure area on the right side of the console, expand **Java and Process Management**.
6. Click **Process definition**.
7. In the **Additional Properties** area, click **Java Virtual Machine**.
8. In the **Additional Properties** area, click **Custom properties**.
9. Click **New**.
10. In the **Name** text box, type one of the following entries, based on the application you are configuring:

    - `fiper.webtop.parmfile`
    - `fiper.webdashboard.parmfile`

11. In the **Value** text box, type one of the following entries, based on the application you are configuring:

    - `<see_install_dir>\config\webtop.properties`
    - `<see_install_dir>\config\webdashboard.properties`

    On Linux, use a forward slash "/" in the entry (instead of a back slash "\").

12. Click **OK**.
13. Perform one of the following actions, based on your SIMULIA Execution Engine security level:

    - **No security**: Continue to *Editing the Properties File*.
    - **Security enabled**: Continue to *Assigning the User Role*.

Assigning the User Role

You need to define which users (or groups of users) are allowed access to the web-based applications. If users are not explicitly given access via security roles, as described below, they will not be able to access the applications.

1. Under **Applications** / **Application Types** on the left side of the console, click **WebSphere enterprise applications**.
2. In the **Name** column, click one of the following links, based on the application you are configuring:
   - **webtop_war**
   - **webdashboard_war**
3. In the **Detail Properties** area on the right side of the console, click **Security role to user/group mapping**.

   The mapping information appears.
4. Click the check box to the left of the **fiperuser** security role.
5. Click **Map Groups**.

   Note: Individual users can be added to a security role by clicking the check box next to the role name and clicking the **Map Users** button.

   The **Map users/groups** screen appears.
6. Click **Search**.

   A list of known groups in the LDAP directory appears.
7. Select a group or multiple groups.
8. Copy the groups to the list on the right side by clicking the  button.

   You can also remove groups using the  button. Contact your local system administrator for more information on the groups that you should be using.
9. Click **OK**.

   You are returned to the **Security role to user/group mapping** screen, and the group you selected is now listed in the **Mapped groups** column.
10. For the WebDashboard only, repeat step 4 through step 9 for the **fiperadmin** role.

Note: Groups mapped to the **fiperadmin** role have access to more WebDashboard features than those groups mapped to the **fiperuser** role. For more information, see *Using the WebDashboard*.

11. Click **OK** to save your changes.
12. Near the top of the right side of the WebSphere console, click **Save** to save your configuration.
13. Log out of the WebSphere Integrated Solutions Console.

Editing the Properties File

You may need to verify or edit the `webtop.properties` file or `webdashboard.properties` file for proper operation of the application.

The WebTop and WebDashboard read certain installation and SIMULIA Execution Engine connection information from a properties file. In some cases this file may be properly configured during the installation, especially if the WebTop/WebDashboard is running on the same machine as the SIMULIA Execution Engine. However, you should verify that this information is correct regardless of your deployment strategy.

1. Navigate to the following directory in your SIMULIA Execution Engine installation:

   ```
   <SEE_install_dir>/config/
   ```

2. Open one of the following files, based on the application you are configuring, in the text editor of your choice:

   - `webtop.properties`
   - `webdashboard.properties`

3. Locate one of the following entries in the file, based on the application you are configuring:

   - `fiper.webtop.acs.cprfile=`
   - `fiper.webdashboard.acs.cprfile=`

4. Verify that the path following the = sign points to the `.cpr` file that will be used to connect to the WebTop/WebDashboard (the `.cpr` file that matches the SIMULIA Execution Engine that will be used by the WebTop/WebDashboard).

   For example:

   ```
   C:\SIMULIA\ExecutionEngine\5.9\config\seehost.cpr
   ```

Important: You need to make sure this connection profile exists on the system running the WebTop/WebDashboard. For more information, see *Creating the Connection Profile File*. Furthermore, it is recommended that you verify that the connection profile works correctly by connecting to the SIMULIA Execution Engine with any SIMULIA Execution Engine interface (for example, a SIMULIA Execution Engine station or the Dashboard).

5. In the `webtop.properties` file only, you must specify a location for the working (temp) directory for the WebTop application. Find the following lines in this file:

   ```
   #Defines temp work directory for WebTop, must be writeable
   #fiper.system.temp=c:/temp/webtop
   ```

   Remove the `#` in the `fiper.system.temp` line to uncomment it, and replace `c:/temp/webtop` with a path that can serve as a working directory that the WebTop application can write temporary files to. Be sure that the directory/folder that you specify is not subject to automatic cleanup on Linux systems.

   If the SIMULIA Execution Engine and WebTop applications are accessing the same file system, it is recommended that you configure different working (temp) directories for each application.

6. Save and close the file.
7. Restart WebSphere. For more information on stopping and restarting WebSphere, see *Restarting the SIMULIA Execution Engine in WebSphere*.

Verifying the Installation

Once you have deployed and configured the WebTop/WebDashboard application, you need to verify that the application is running correctly and can be accessed. Since both the WebTop and WebDashboard run in a web browser, you access them using a URL that is specific to your SIMULIA Execution Engine environment.

About the Application Port Number and URL

The WebTop and WebDashboard are both accessed by using a URL in a Web browser. This URL varies based on the application as well as your SIMULIA Execution Engine environment. Before you verify that the application is working correctly, you need to determine your port number and application name in your URL.

Your URL will appear similar to one of the following examples, based on the application you are using:

```
http://hostname:9080/webtop
http://hostname:9080/webdashboard
```

The `webtop` and `webdashboard` portions of the URLs represent the **Context root** string provided during the deployment of the WebTop/WebDashboard (in *Deploying the Web-Based Applications*). If you specified something other than the recommended setting, you should use that customized setting in your application URL.

The port number—in these examples, `9080`—is dependent on the installation of WebSphere and could be different, especially if more than one WebSphere installation is present on the system or you are using multiple WebSphere profiles. You can confirm the port number by checking the **WC_defaulthost** port setting under **Servers** / **Application servers** / **server1** / **Ports** in the WebSphere console for the WebTop/WebDashboard (not the console for the SIMULIA Execution Engine).

Viewing the Application

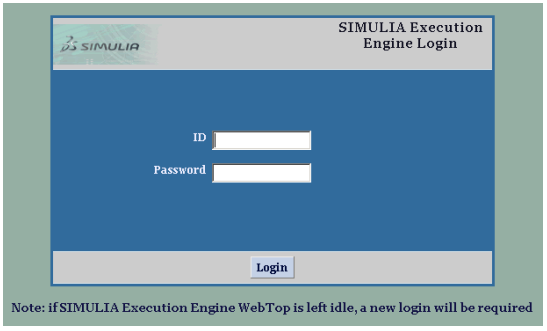
Once you have determined the correct URL for accessing your web-based application, you need to verify that it appears correctly in a web browser. If the application appears correctly, you can begin using it immediately.

1. Open a web browser, and specify the appropriate URL for the application you want to view.

   For more information on determining this URL, see *About the Application Port Number and URL*.

2. Verify that the application appears.

   The WebTop login page is shown below.



3. If desired, you can begin using the application.

   For more information about using the WebTop, see the *SIMULIA Execution Engine WebTop Guide*. For more information on using the WebDashboard, see *Using the WebDashboard*.

Using SIMULIA Execution Engine Interfaces

This section describes the interfaces that a system admistrator can use to control and monitor the SIMULIA Execution Engine and the SIMULIA Execution Engine stations. The interfaces include the station itself, the Dashboard, the WebDashboard, 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*.

System administrators should refer to *Configuring the WebTop or WebDashboard for the SIMULIA Execution Engine* for details about configuring and starting the WebTop or WebDashboard.

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.

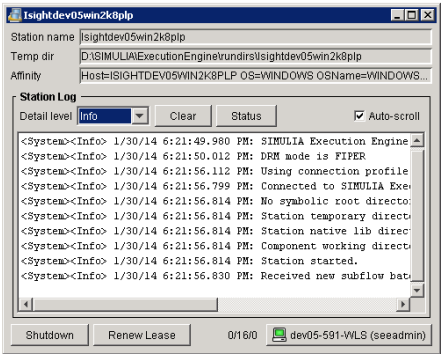
The SIMULIA Execution Engine station is a multithreaded Java application that can handle many concurrent requests. For an overview of the complete SIMULIA Execution Engine system, including stations, see *About the SIMULIA Execution Engine Environment*.

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 affinites. 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 and WebDashboard

The Dashboard and WebDashboard show 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).

For more information, see *Publishing to the Library*.

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  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.

     For complete information on this process, see *Creating the Connection Profile File*.
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\`
   - **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`

     (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 WebDashboard, click on any station name in the station list. In the new screen that appears, click **Restart**.
- 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 `stations.properties` file (and the `acs.properties` file) must include the forward slash on both Linux and Windows operating systems.

When you are configuring WebSphere, you can use the forward slash or back slash on 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.

Station Properties and Command Line Options

station.properties entry	Equivalent Command Line Argument	Full Description
fiper.station.affinity	affinity:<*affinity_name*>	*Affinities*
fiper.station.allowedusers	allowedusers:<*list_of_users*>	*Allowed Station Users*
fiper.station.concurrency	concurrency:<*number*>	*Concurrency Limit*
fiper.station.description	desc:<*text_string*> or "<*text_string*>"	*Station Description*
fiper.station.jms.persistent	*not available*	*Persistent JMS Messaging*
fiper.station.leaseinterval	leaseinterval:<*minutes*>	*Lease Interval*
fiper.station.logfile	logfile:<*log_file_name*>	*Log File Location*
fiper.station.loglevel	loglevel:[debug|info|warn|error|syserror]	*Default Logging Level*

station.properties entry	Equivalent Command Line Argument	Full Description
fiper.station.maxquiescetime	*not available*	*Shutdown Timeout Command*
fiper.station.name	name:<*station_name*>	*Station Name*
fiper.station.nogui	nogui:[true|false]	*Running a Station Without a GUI*
fiper.station.retrydelay	*not available*	*Windows Service Retry Delay*
fiper.station.runas	*not available*	*Station-Specific Run-As Behavior*
fiper.station.saveLogOnError	*not available*	*LSF Station Error Logs*
fiper.station.serviceconfigfile	serviceconfigfile:<*filename*>	*Windows Service Configuration File*
fiper.station.tempdir	tempdir:<*directory*>	*Temporary Directory*
fiper.station.workdir	*not available*	*Working Directory*
fiper.logon.prompt	logonprompt:[yes|no]	*Logon Prompt*
fiper.logon.profile	profile:<*cpr_filename*>	*Connection Profile*
fiper.logon.prop.user	user:<*username*>	*User Name*
fiper.logon.prop.pw	pw:<*password*>	*Password*
fiper.grid.ssh.configfile	*not available*	*Grid Distributed Execution*
fiper.file.*RootA*	*not available*	*Shared File System Symbolic Root Directories for File Parameters*
fiper.security.station.domain	*not available*	*Security Domain for Run-As Mode on Windows*
fiper.security.substation.cache.size	*not available*	*Substation Cache Size*
fiper.security.substation.idle.timeout	*not available*	*Idle Substation Timeout*

station.properties entry	Equivalent Command Line Argument	Full Description
not available	configfile:<*filename*> or @*filename*	*Configuration File of Command Line Arguments*
not available	-help	*Command Line Argument Help*
not available	locale:<*locale_string*>	*Locale*
not available	logonmode:[GUI|console]	*Logon Mode*
not available	service:[stop|start]	*Starting the Station as a Windows Service*
not available	unix_user:<*user_id*>	*Linux User for Extended Grid Credentials*
not available	unix_pw:<*password*>	*Linux Password for Extended Grid Credentials*
not available	win_user:<*user_id*>	*Windows User for Extended Grid Credentials*
not available	win_domain:<*domain_name*>	*Windows Domain for Extended Grid Credentials*
not available	win_pw:<*password*>	*Windows Password for Extended Grid Credentials*

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:seeadmin,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:

```
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:

```
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:

```
station tempdir:c:\simulia\Execution Engine\5.9\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\5.9\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
```

See *Using Distributed Resource Management with the SIMULIA Execution Engine* for details about DRM.

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 infrustructure 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:

```
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:

```
fiper.logon.prompt=no
fiper.logon.profile=c:/simulia/Execution Engine/5.9/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.

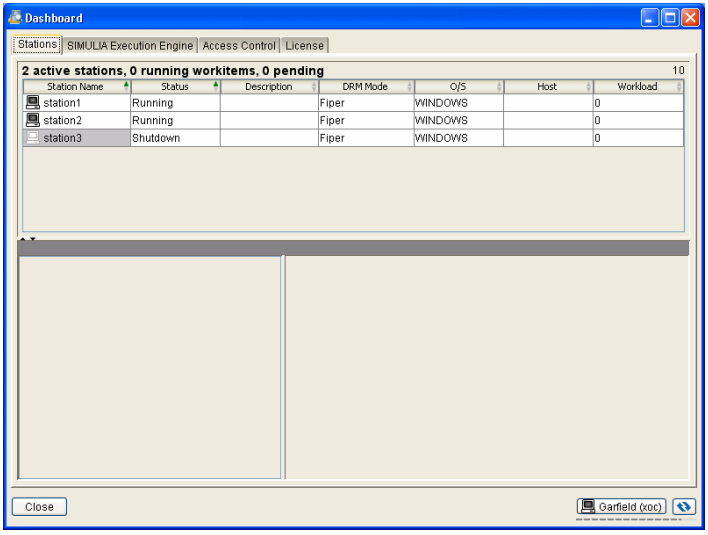
To run the Dashboard on a computer other than the one containing the SIMULIA Execution Engine installation, you must install the SIMULIA Execution Engine.

Alternatively, you can use the WebDashboard to perform the same operations as the Dashboard, without needing to install the SIMULIA Execution Engine. For more information, see *Using the WebDashboard*.

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 and WebDashboard* 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. See *Using Distributed Resource Management with the SIMULIA Execution Engine*.
- **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

```
@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:

```
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

```
-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`

Note: 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\5.9\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 `fiperadmin` 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 `fiperadmin` 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.

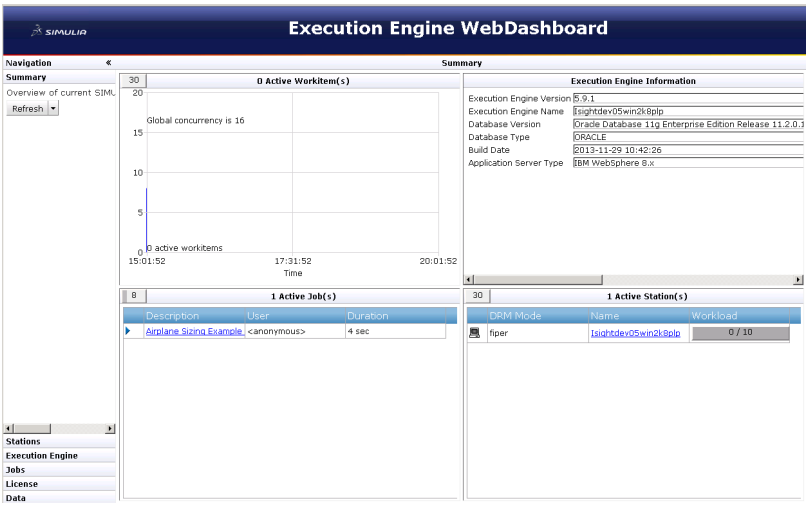
Using the WebDashboard

The WebDashboard is an application that displays the current status of the SIMULIA Execution Engine. It is similar to the Dashboard interface, except that it runs in a web browser and does not require you to install any client software.

About the WebDashboard Interface

The WebDashboard interface shows a list of running stations, the number of running jobs, the workitems on each station, and the licenses being used by the SIMULIA Execution Engine. It also allows you to search for a particular job using specific search criteria.

When you log in to the WebDashboard, a summary screen is displayed.



This screen provides an overview of the SIMULIA Execution Engine status including the following information:

- **Active Workitems graph**. Shows two workitems-related graphs on a single pair of axes:
- The blue line is a graph of the total number of workitems dispatched to all attached stations plotted against time.
- The red line is a graph of the total number of workitems that can be processed concurrently by all attached stations plotted against time. This number is calculated by adding up the concurrency settings for each of the running stations attached to the SIMULIA Execution Engine.

  This is historical data starting at the time the WebDashboard was launched (or the browser window was last refreshed), up to a maximum of five hours. It is periodically updated with the latest information from the SIMULIA Execution Engine.

  Note: This feature is available only if you have administrative privileges.

- **SIMULIA Execution Engine Information table**. Shows information such as the name of the SIMULIA Execution Engine, the type of database used, and the application server used by the SIMULIA Execution Engine (WebSphere or WebLogic). You can get more specific information as described in *Viewing Details for the SIMULIA Execution Engine*.
- **Active Jobs table**. Shows the status of the jobs currently being executed (if any), the name of the jobs currently being executed, the user executing the jobs, and how long the jobs

have been running. To see additional job information and control the execution of current jobs, see *Working with Running Jobs*.

Note: This feature is available only if you have administrative privileges.

- **Active Stations table**. Shows the stations currently connected to the SIMULIA Execution Engine (in either the active and shutdown state), the DRM mode, and the current workload for each station. You can access additional station information by clicking the link that corresponds to the station name. For more information on the options that appear when this link is clicked, as well as how to view more station information, see *Viewing Station Information*.

Note: The **Active Workitems** graph, **Active Jobs** table, and **Active Stations** table automatically refresh after a set period of time (which can be determined by viewing the progress bar in the upper left corner of each section). You can also manually refresh all sections or an individual section using the **Refresh** button on the left side of the interface.

Accessing the WebDashboard in a Browser

To view the WebDashboard, you open a particular URL in a web browser. The URL is determined by the system administrator who installed and configured the SIMULIA Execution Engine and the WebDashboard.

1. Contact your SIMULIA Execution Engine administrator and verify that the WebDashboard has been installed for the appropriate SIMULIA Execution Engine.
2. Open the following URL in a browser:

   ```
   http://hostname:9080/webdashboard
   ```

   where *hostname* is the name of the system running the WebDashboard.

Note: The port number is dependent on the installation of WebSphere and could be different, especially if more than one WebSphere is installed on the system. You can confirm the port number by checking the **WC_defaulthost** port setting under **Servers** / **Application servers** / **server1** / **Ports** in the WebSphere Integrated Solutions Console *for the WebDashboard*. See *About New Profile Port Numbers* for more information.

3. When prompted, log into the WebDashboard using a username/password combination that is valid for the corresponding SIMULIA Execution Engine.

Viewing Station Information

The WebDashboard allows you to view station details.

1. Open the WebDashboard as described in *Accessing the WebDashboard in a Browser*.
2. On the left side of the WebDashboard interface, click the **Stations** option.

   The station details appear. The following information is shown in the corresponding columns on the WebDashboard:

   - **Status**. The current availability of the station (represented by an icon) is displayed in the first column. You can also see the status in text form by placing your mouse over any icon in this column. The station can be in any of the following states: Running, Quieting, Shutdown, or Unknown. See *Station Status Reporting in the Dashboard and WebDashboard* for definitions of these conditions.
   - **DRM Mode**. The DRM mode used by the station: either Fiper or LSF. See *Using Distributed Resource Management with the SIMULIA Execution Engine*.
   - **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*.
   - **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*.
   - **Operating System**. The operating system of the computer running the station, including the operating system architecture, name, and version.
   - **Host Name**. The name of the computer where the station is running.
   - **Workload**. Uses a colored bar and a scale to show how much work the station is currently performing. The workload is displayed as the current number of workitems being processed by the station as compared to the larger of one of the following: ten (10) or the maximum number of workitems concurrently processed by any single station within recent history. For example: If Station A is processing 13 workitems, Station B is processing 3 workitems, and Station C is shut down, the workload bars would contain the following numbers:

     A: 13/13 B: /13 C: 0/13

     The number of workitems being processed by a station may be larger than the concurrency limit for that station. This scenario is true because the number of workitems

being processed includes process workitems that might be waiting for other workitems to complete and are, therefore, not included in the concurrency limit for that station.

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

3. (optional) Filter the displayed stations based on their state using the option on the left side of the screen (under the **Stations** option). You can either view all stations, only those that are active (in the Running state), or only those that are inactive (in the shutdown state).
4. (optional) Click the **Refresh** button on the left side of the interface to verify that you are viewing the most up-to-date information.

Note: The information is automatically refreshed after a set period of time (which can be determined by viewing the progress bar adjacent to the Refresh button).

5. Click a station in the list near the top of the tab to view workitems associated with the selected station.

   The **Workitems** list at the bottom of the interface is populated with the station's workitems.

6. If any workitems are listed, you can view the details of a workitem by clicking the **Workitem ID** column. You can also cancel a workitem (for example, a workitem taking too long to finish or one that appears unable to finish) by selecting the workitem row and clicking **Stop Selected Workitem**.

Restarting or Shutting Down a Station

You can restart or shut down a station from the WebDashboard.

1. Open the WebDashboard as described in *Accessing the WebDashboard in a Browser*.
2. On the left side of the WebDashboard interface, click the **Stations** option.
3. Click any SIMULIA Execution Engine station name in the list at the top of the interface to display a screen showing more station details. This screen also gives you access to the following options:

   - **Remove**. This option allows you to delete inactive (shutdown) stations that you no longer want included in the station list.
   - **Restart**. This option allows you to stop and restart a station. It is available only if the station is using the Fiper DRM mode. For more information on using this option, including its limitations, see *Restarting a SIMULIA Execution Engine Station Remotely*.

This option cannot be used to simply restart stopped stations. It is essentially a remote station reboot (stops a running station and then restarts it remotely).

- **Shutdown**. This option shuts down the selected station. It is available only if the station is using the Fiper DRM mode. Use this option with care because shutting down a station that is actively running workitems can result in lost work.

Viewing Details for the SIMULIA Execution Engine

You can use the WebDashboard to view information about the SIMULIA Execution Engine configuration as well as system properties for the computer hosting the SIMULIA Execution Engine.

Before you begin: This information is available only if you have administrative privileges.

1. On the left side of the WebDashboard interface, click the **Execution Engine** option.

   The SIMULIA Execution Engine details appear, for example server type, release number, and configuration properties.

   The top portion of this screen shows information that is obtained from the SIMULIA Execution Engine configuration. The bottom portion of this screen shows the system properties for the computer on which the SIMULIA Execution Engine is running (as of the time the SIMULIA Execution Engine was started).

2. Click the **Refresh** button on the left side of the interface to verify that you are viewing the most up-to-date information
3. Review the information. By default, the information is limited to system properties beginning with the string “fiper”. You can also click the **All server properties** check box on the left side of the interface to view even more SIMULIA Execution Engine configuration properties.

Working with Running Jobs

You can use the WebDashboard to view job information, including the current status of a job, and to control the execution of jobs that are executing on the SIMULIA Execution Engine.

Before you begin: This information is available only if you have administrative privileges.

1. On the left side of the WebDashboard interface, click the **Jobs** option.

   The Job details appear.

2. (optional) Click the **Refresh** button on the left side of the interface to verify that you are viewing the most up-to-date information.

Note: The information is automatically refreshed after a set period of time (which can be determined by viewing the progress bar adjacent to the **Refresh** button).

3. Click a job in the list near the top of the interface.

   The list at the bottom of the interface is populated with workitems related to the selected job. It does not show completed or pending workitems.

4. If any workitems are listed, you can view the details of a workitem by clicking the **Workitem ID** column. You can also cancel a workitem (for example, a workitem taking too long to finish or one that appears unable to finish) by selecting the workitem row and clicking **Stop Selected Workitem**.
5. Click any job name in the station list at the top of the interface to display a screen showing more job details.

   The job list contains the following columns:

Description	A short description of the job, as specified in the Runtime Gateway *Job Name* field.
Model Name	The name of the model selected for this job.
User	The login ID of the user that started the job.
Started	The date/time that the job was started.
Duration	The amount of time that the job has been running (as of the last refresh).

   The workitem list contains the following columns:

Workitem ID	A unique ID that identifies a workitem that is currently running as part of the selected job.
Station Name	The name of the station running the workitem.
Component Type	The type of component that this workitem represents.
Dispatched	The date/time that the workitem started running on the station.
Duration	The amount of time that the workitem has been running (as of the last refresh).

6. If desired, you can control running jobs by using the following options:
   - **Stop**. This option allows you permanently halt a running job.
   - **Pause/Resume**. This option allows you to temporarily stop and then restart a job.

Searching for Jobs

You can search through all of the jobs in your SIMULIA Execution Engine database (your job history) using specified criteria. This feature allows you to easily locate jobs that have been previously executed without having to search through the entire list of jobs.

Before you begin: This information is available only if you have administrative privileges.

1. On the left side of the WebDashboard interface, click the **Data** option.

   The **Data** options appear. These options allow you to narrow your search for jobs located in your SIMULIA Execution Engine database.

2. Specify the job information criteria to use to filter your search by using any of the following options (you can use the * wildcard in these searches):

Option	Description
User Name	Enter the name of a specific SIMULIA Execution Engine user.
Group Name	Enter the name of a Fiper Group (from the library's **ACL** tab) that contains the users whose jobs you wish to retrieve.
Model Name	Enter the name of an Isight model.
Job Name	Enter the name of an Isight job.
Job ID	Enter the ID of an Isight job.
Bigger than	Enter a number, and select the units indicating the minimum job size to retrieve.
Older than	Enter a number, and select the units indicating the minimum job age (based on job start date) to retrieve.
Run Date	Enter date information to filter the jobs based on the job start date. • Specify only a single date to use just that date for the search. • Specify only a **From** date to search for any job that started on or after that date. • Specify only a **Through** date to search for any job that started on or before that date. • Specify a date range by entering both a **From** and **Through** date.
Job Status	Select from one of the following job statuses: **Initializing**, **Started**, **Queued**, **Running**, **Done**, **Stopping**, **Paused**, **Importing**, and **Created**.
Completion Code	Select from one of the following completion codes: **OK**, **CANCELLED**, **FAILED**, or **SYSFAILED**. For more information on the codes, see *About the Job Database Interface* in the *Isight User's Guide*.  **Note:** Although you can select any combination of **Job Status** and **Completion Code**, not all combinations make sense. If a selected combination is incompatible, the search will not return any job data.

3. Click **Query** to retrieve a list of jobs that match your search criteria. The results are displayed in a scrollable table on the right side of the interface. This table includes a summary of the total number of jobs retrieved, as well as the total space used in the database, on disk, and overall.
4. Click any listed job to view details and job control options. For more information, see *Working with Running Jobs*.

   The jobs list contains the following columns:

Description	A short description of the job, as specified in the Runtime Gateway *Job Name* field.
Model Name	The name of the model selected for this job.
Model Version	The version of the model selected for this job.
User	The login ID of the user that started the job.
Submission Hostname	The hostname of the machine from which the user submitted the job. For jobs submitted via the WebTop, the name of the web server is shown.
Db Bytes	The amount of memory used by the job in the database.
Disk Bytes	The amount of memory used by the job data in folders on the disk (not including any database files).
Total Bytes	The total amount of memory used by the job, including **Db Bytes** and **Disk Bytes**.

Deleting Non-Running Jobs

You can use the WebDashboard to delete non-running jobs—those not currently being executed by the SIMULIA Execution Engine—from the SIMULIA Execution Engine database.

Before you begin: This information is available only if you have administrative privileges.

1. On the left side of the WebDashboard interface, click the **Data** option.

   The **Data** options appear.

2. Search for job information using any combination of the options on the left side of the interface. For more information, see *Searching for Jobs*.
3. Perform one of the following actions:

   - To delete individual non-running jobs, select the job that you want to delete, and click **Delete Selected** at the bottom of the interface.

- To delete all listed non-running jobs, click **Delete All Found** at the bottom of the interface.

4. Click **OK** to confirm the deletion of the specified jobs and all associated results data.

Viewing License Usage Information

You can use the WebDashboard to view 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.

1. On the left side of the WebDashboard interface, click the **License** option.

   The License options appear.

   These options allows you to view how many instance 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 returned information 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.

2. Select an item from the **License feature** list, or select **other -> enter below** and specify a license feature supported by your license server in the corresponding text box.

   Information about that license feature is displayed in the large 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 **Query** to verify that you are looking at the most up-to-date information. This option is especially useful if you leave the WebDashboard 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.

To enable most SIMULIA Execution Engine functionality, the library must also be preloaded with the basic system metamodels (components) as described in *Publishing to the Library*.

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 x.x**, 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.

     For complete information on this process, see *Creating the Connection Profile File*.

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.

Using Distributed Resource Management with the SIMULIA Execution Engine

This section describes how the SIMULIA Execution Engine performs distributed resource management (DRM) with Fiper DRM, the default enterprise load-balancing framework provided with the SIMULIA Execution Engine; with LSF DRM, the Platform LSF third-party product; and with Mixed-Mode DRM, a combination of the Fiper and LSF DRM options.

Default Fiper DRM included with the SIMULIA Execution Engine

You can use Fiper distributed resource management (DRM) to control the distribution and execution of your workload in a SIMULIA Execution Engine environment. The default DRM system, Fiper DRM, is an enterprise load-balancing framework provided with the SIMULIA Execution Engine and does not require any additional third-party software.

The Fiper DRM system distributes workload from the SIMULIA Execution Engine to the SIMULIA Execution Engine stations, which must be running and awaiting work items sent from the SIMULIA Execution Engine server. Each station is assigned a number of available slots based on the concurrency settings, with the default concurrency for each station equal to twice the number of CPUs. Each running work item is assigned to a specific slot on a specific station and, in most cases, will hold the assigned slot for the duration of the work item. Certain work items that are waiting for other work or work items not using local resources on the station (such as design drivers waiting for subflow completion or OS Command components running a grid plug-in) may temporarily relinquish a slot so that it becomes available for other work. If no slots are available on any station that matches the work item's affinity requirements during the server's dispatch process, the work item will be queued on the server until an acceptable slot becomes available.

The advantages of using the built-in Fiper DRM include very low scheduling overhead, which is particularly valuable for fast-running work items and for mixed workflows using the mixed-mode DRM (see *Mixed-Mode DRM and the SIMULIA Execution Engine*). However, Fiper DRM does not consider relative machine speed or dynamic scheduling parameters such as machine load, available memory, or available disk space when selecting a station; it considers each available slot to be equivalent. As a result, Fiper DRM does not always provide ideal

scheduling for workflows containing long-running, resource-intensive work items. For such workflows, the Platform LSF DRM option (see *LSF DRM and the SIMULIA Execution Engine*) or the Mixed-Mode DRM option (see *Mixed-Mode DRM and the SIMULIA Execution Engine*) may be appropriate.

LSF DRM and the SIMULIA Execution Engine

You can use Platform LSF distributed resource management (DRM) to control the distribution and execution of your workload in a SIMULIA Execution Engine environment.

The SIMULIA Execution Engine can be configured to use LSF, the third-party distributed resource management system, to optimize the utilization of compute resources for high-performance computing tasks. Once the system is configured, you can set LSF-specific options for individual Isight components (via the **DRM Settings** tab on the **Properties** dialog box). For more information on these component settings, see *Configuring the LSF DRM Settings* in the *Isight Component Guide*.

Enabling the LSF DRM feature can significantly enhance the scheduling capabilities of the SIMULIA Execution Engine, particularly for workflows with time-consuming, resource-intensive work items. When using the Fiper DRM option, the SIMULIA Execution Engine requires that stations be running and awaiting work items sent from the SIMULIA Execution Engine server. When using the LSF DRM option, the SIMULIA Execution Engine uses LSF to launch SIMULIA Execution Engine station processes as needed on LSF compute nodes. Each process is then connected to the SIMULIA Execution Engine server, runs a single work item, and is terminated. Each work item dispatched with the LSF DRM corresponds to a single LSF job. This configuration gives LSF direct control over the station processes that are actually doing work, both for resource management and accounting purposes, and allows the SIMULIA Execution Engine server to utilize LSF's sophisticated scheduling capabilities to select the optimal node for each piece of work.

Unlike the Fiper DRM, the LSF DRM imposes some scheduling and process-launching overhead on each SIMULIA Execution Engine work item. However, for compute-intensive, long-running work items the improved scheduling and job management that LSF DRM provides greatly outweighs this overhead. For workflows composed of significant numbers of small, short-running work items, the LSF DRM can reduce SIMULIA Execution Engine job throughput when used exclusively. Mixed-mode DRM, where both the Fiper and LSF distributed resource managers are enabled on the SIMULIA Execution Engine server, can be used to manage this scenario. For more information, see *Mixed-Mode DRM and the SIMULIA Execution Engine*.

Using LSF with the LSF Grid Plug-in

Isight and the SIMULIA Execution Engine can access the LSF system through the use of the LSF Grid plug-in. The OS Command, Simcode, and Abaqus components provide this functionality.

The LSF Grid plug-in option allows an Isight component to submit command-line codes to an LSF cluster directly from an Isight installation or a SIMULIA Execution Engine station. For more information on using the Grid plug-in with LSF or other distributed resource management systems (DRMs), see *OS Command Component*, *Abaqus Component*, and *Simcode Component* sections in the *Isight Component Guide*.

Using the LSF Grid plug-in is a distinct but often complementary scenario compared to the LSF DRM available with the SIMULIA Execution Engine. The plug-in represents a specific use case: when you need to run command line–based, compute-intensive codes on an LSF cluster. SIMULIA Execution Engine stations are typically not installed on these cluster nodes, and the LSF Grid plug-in can be used to access the back-office LSF systems. Furthermore, these back-office clusters may be using a DRM other than LSF, such as PBS/Pro or Torque. In this scenario you can use the appropriate LSF Grid plug-in to access those nodes for command-line codes only.

If LSF is installed on all of the nodes (both clusters and individual systems), you will most likely want to use the LSF DRM and limit the usage of the LSF Grid plug-in.

Using LSF Clusters

Whether you have LSF installed as a back-office computer cluster devoted to high-performance computing or you have LSF installed on every system on your network, you can take advantage of the available computing power.

If the SIMULIA Execution Engine server is installed outside of the LSF cluster, you can execute a SIMULIA Execution Engine station using the Fiper DRM option on the LSF head node (or several stations on several LSF nodes for redundancy) of a compute cluster as a gateway to the cluster. Once started, you can send compute-intensive, command line work items to these gateway stations using the affinity matching capability available in Isight. These gateway stations use the LSF Grid plug-in to submit LSF jobs to the compute cluster. This approach is limited to command line codes, such as those used by the OS Command and Abaqus components. Therefore, a more comprehensive overall scheduling capability can be achieved if LSF is available on all of the nodes and the LSF DRM option is used.

Understanding LSF Version Support and Prerequisites

You must make sure you are installing a supported version of LSF and have the necessary prerequisites.

For SIMULIA Execution Engine 5.9 only LSF version 7.0 update 6 (7.0.6) is supported.

In addition, the following prerequisites are necessary for installing LSF with the SIMULIA Execution Engine:

- **Linux**: WebSphere cannot be installed as root unless the Run-As feature is enabled. WebSphere must be installed and started by a user who has the necessary credentials to submit jobs to the LSF cluster, and this user cannot be root. If Run-As is enabled, WebSphere can run as root.
- You must install the SIMULIA Execution Engine normally and deploy on WebSphere.
- Do not start the SIMULIA Execution Engine in your application server before you have installed and configured LSF (as described in *Verifying the SIMULIA Execution Engine Configuration* and in *Configuring LSF for the SIMULIA Execution Engine*). For example, on Linux you should source the `profile.lsf` file in the command shell before starting the SIMULIA Execution Engine. The SIMULIA Execution Engine must be able to find the LSF binary files in the system executable path.
- You must install the SIMULIA Execution Engine station software on all computers that will run a SIMULIA Execution Engine station.

Limitations of LSF with the SIMULIA Execution Engine

The use of LSF distributed resource management (DRM) with the SIMULIA Execution Engine is subject to the following limitations:

- By default, SIMULIA Execution Engine work items execute with a common dedicated user ID (the ID used to start the SIMULIA Execution Engine station service). The security context of the executing code will be that of the dedicated ID, which in general is created specifically for the execution of the SIMULIA Execution Engine workload. To run SIMULIA Execution Engine work items under the submitter's security credentials, you must enable the SIMULIA Execution Engine Run-As security feature as described in *Configuring Station (Run-As) Security*.
- By default, a SIMULIA Execution Engine user cannot specify general LSF resource requirements for work items dispatched with the LSF DRM. However, standard SIMULIA Execution Engine affinities can be used. To specify more advanced LSF resource requirements for components within a model using the **Properties** dialog box, see *Configuring the LSF DRM Settings* in the *Isight Component Guide*.
- LSF preemptive scheduling and suspension of in-progress SIMULIA Execution Engine work items is not possible.

Mixed-Mode DRM and the SIMULIA Execution Engine

Depending on the type of work items you submit to the SIMULIA Execution Engine, it may be desirable to configure a single SIMULIA Execution Engine server to use a combination of the Fiper and LSF distributed resource management (DRM) options.

The LSF DRM option imposes significant overhead because of the sophisticated scheduling procedures. For resource-intensive, long-running work items, this improved scheduling is highly desirable to optimize the use of available computing resources. However, for short-running work items, this overhead may lead to reduced throughput for models with short-running components. To overcome this issue, an administrator can configure the SIMULIA Execution Engine to enable both the Fiper and LSF DRM options, a scenario known as Mixed-Mode DRM.

Mixed-Mode DRM is enabled in the `acs.properties` file. If both the `fiper.system.drm` and `fiper.system.drm.2` settings are defined in the `acs.properties` file, both DRM options are enabled. The DRM option listed first is considered the default type, which generally means this DRM option is used if the Isight model developer does not specify a different DRM type at the component level.

For example, the following settings enable both DRM types, with the Fiper DRM option being the default choice:

```
fiper.system.drm=fiper
fiper.system.drm.2=lsf
```

When Mixed-Mode DRM is enabled, the SIMULIA Execution Engine administrator can configure a maximum run time for work items that will be dispatched using the Fiper DRM option. This configuration is used to prevent model developers from overwhelming available Fiper DRM stations with long-running or resource-intensive executions. Normally, the stations using the Fiper DRM option are used for executing short-running work items. If the administrator sets the `fiper.system.drm.fipertimelimit` setting in the `acs.properties` file, any work item with a maximum run time greater than this specified limit is dispatched only with the LSF DRM option. If you do not specify a DRM option or you specify the LSF DRM option, the work item is dispatched using the LSF DRM option. If you specify the Fiper DRM option for the work item, the dispatch will fail because the administrator prohibited this scenario based on the configuration of the `fiper.system.drm.fipertimelimit` setting. This setting is meaningful only if both Fiper and LSF DRMs are enabled.

For additional information on the use of Mixed-Mode DRM and setting the DRM mode for Isight components, see *Configuring the LSF DRM Settings* in the *Isight Component Guide*.

Verifying the SIMULIA Execution Engine Configuration

You must verify the SIMULIA Execution Engine configuration settings prior to configuring and using LSF distributed resource management (DRM) and Mixed-Mode DRM. If you fail to verify these settings, you may experience problems executing LSF and the SIMULIA Execution Engine.

Check the following configuration details:

- In your connection profile verify that you did not specify the fully qualified domain name of the system running the SIMULIA Execution Engine. Use the non-FQDN hostname. For more information on this setting, see *Creating the Connection Profile File*.
- In the `<SEE_install_dir>/config/acs.properties` file, check the following:
  - Verify that you have *not* set the SIMULIA Execution Engine name property (`fiper.acs.name`). The SIMULIA Execution Engine name must be the non-FQDN hostname.
  - Verify that the `fiper.acs.isWindowsService` property is set correctly. This property informs the SIMULIA Execution Engine if the application server is running as a service.
- The user running your application server must be able to submit LSF jobs successfully to all hosts. To perform this action, the user must exist, the user name and password must be the same on all computers, and the user's password must be made known to LSF using the `lspasswd` command.
- Before starting your application server, the user account specified in the previous item must already be set up for the LSF cluster. On Windows systems user accounts are typically configured at the time LSF is installed.
- The `lspasswd` command applies to installations involving Windows only. The `lspasswd` command must be run on a Windows system, even in a mixed cluster. The `lspasswd` command is not required if there are no Windows systems. This command is an LSF utility program that each user must run once to store a Windows password in the LSF system. This step is required to run LSF work on a Windows LSF node.

Configuring LSF for the SIMULIA Execution Engine

This procedure is designed to help an experienced LSF administrator set up LSF for use with the SIMULIA Execution Engine.

In the following procedure <*SEE_install_directory*> represents the location on each system where the SIMULIA Execution Engine is installed and <*lsf_install_directory*> represents the location on each system where LSF is installed.

1. Add the following directory to the system `PATH` variable on the SIMULIA Execution Engine system, as well as on every system running a SIMULIA Execution Engine station:

   ```
   <SEE_install_dir>/<os_dir>/code/bin/
   ```

2. If you are installing on a Windows system:
   a. Copy the `elim.fiper.exe` file from the following directory:

      ```
      <SEE_install_dir>\win_b64\code\bin\
      ```

      to the `<lsf_install_dir>\7.0\etc\` directory on each system running a SIMULIA Execution Engine station.

   b. Create a file called `elim_fiper_config` in the `<lsf_install_dir>\7.0\etc\` directory. This file must contain the path to the SIMULIA Execution Engine installation. For example:

      ```
      elim.fiperbindir=C:\SIMULIA\ExecutionEngine\5.9\win_b64\code\bin
      ```

3. If you are installing on a Linux system:

   Note: This entry assumes that all the Linux systems configured to be used as transient stations should have the SIMULIA Execution Engine installed at the same root path on all systems. Another option is to install the SIMULIA Execution Engine on one Linux system and share the installation on all the Linux hosts that hosts LSF transient stations using the same mount point path as the installation path.

   a. Create a one-line shell script called `elim.fiper` in `<lsf_install_dir>/etc/` that points to the SIMULIA Execution Engine installation, and give all users execute permission on the script (mode 755). Verify that the script contains the following information (adjust your installation path, if necessary):

      ```
      #!/bin/sh
      ```

```
exec
/opt/SIMULIA/ExecutionEngine/5.9/linux_a64/code/bin/elim.fiper
 $*
```

b. Create a one line shell script in `/usr/bin/transtation/` that points to the SIMULIA Execution Engine installation, and give all users execute permission on this script (mode 755). This script must contain the following information (be sure to adjust your installation path, if necessary):

```
#!/bin/sh
exec
/opt/SIMULIA/ExecutionEngine/5.9/linux_a64/code/command/transtation
 $*
```

4. Add the following items to the `Begin Resource` section in the `lsf.shared` file:

Note: This section may not exist in your `lsf.shared` file. If it does not, you need to create it. If it is commented out, you need to uncomment it. After any modifications, it should appear as shown below:

```
Begin Resource
RESOURCENAME  TYPE    INTERVAL INCREASING  DESCRIPTION
     # Keywords
   acs         String  30       ()          (FIPER acs
name)
   fs_aff1     String  30       ()          (FIPER
affinity)
   fs_aff2     String  30       ()          (FIPER
affinity)
   fs_aff3     String  30       ()          (FIPER
affinity)
   fs_aff4     String  30       ()          (FIPER
affinity)
   fs_aff5     String  30       ()          (FIPER
affinity)
   fs_aff6     String  30       ()          (FIPER
affinity)
   fs_aff7     String  30       ()          (FIPER
affinity)
   fs_aff8     String  30       ()          (FIPER
affinity)
   fs_aff9     String  30       ()          (FIPER
affinity)
   fs_aff10    String  30       ()          (FIPER
```

```
affinity)
End Resource
```

5. Add the following to the `ResourceMap` section in the `lsf.cluster.*` file:

Note: This section may not exist in your `lsf.cluster.*` file. If it does not, you need to create it. If it is commented out, you need to uncomment it. After any modifications, it should appear as shown below:

```
Begin ResourceMap
RESOURCENAME  LOCATION
acs              [default]
fs_aff1          [default]
fs_aff2          [default]
fs_aff3          [default]
fs_aff4          [default]
fs_aff5          [default]
fs_aff6          [default]
fs_aff7          [default]
fs_aff8          [default]
fs_aff9          [default]
fs_aff10         [default]
End ResourceMap
```

6. Open the `station.properties` file with the text editor of your choice.
7. Modify the `station.properties` file as follows:
   a. Set the SIMULIA Execution Engine station temporary folder to a location accessible to and writable by all users. For example:

   ```
   fiper.station.tempdir=C:/temp/SIMULIAExecutionEngine
   ```

   Important: Do not use the `fiper.station.name` property to change the name of a SIMULIA Execution Engine station when using LSF. You must use the default station name.

   b. Set the login properties to allow the LSF station to log on to the SIMULIA Execution Engine using some generic credentials with no user interaction. For example:

   ```
   fiper.logon.profile=c:/simulia/ExecutionEngine/5.9/config/myacs.cpr
   fiper.logon.prompt=no
   fiper.logon.prop.user=<username>
   fiper.logon.prop.pw=<mycleartextpassword>
   ```

If you want to use a secured password instead of clear text, use the SIMULIA Execution Engine Command Client to generate the text of the secured password. Use the following command to encrypt the clear text password you want:

```
<SEE_install_dir>/<os_dir>/code/command/fipercmd encrypt
password:<password-cleartext>
```

This command will write to the console a text string of the form

```
$decode$>>2<<@#$%^&...^%@#&*
```

Copy the entire string, and paste it into the `station.properties` file as the password value:

```
fiper.logon.prop.pw=$decode$>>2<<@#$%^&...^%@#&*
```

The presence of the `$decode$` prefix indicates that a secured password is being provided. (If your `station.properties` file contains the `fiper.logon.prop.secured=true/false` line, delete it.)

8. Save and close the `station.properties` file.
9. On the system running the SIMULIA Execution Engine, open the `acs.properties` file with the text editor of your choice.
10. Modify the `acs.properties` file as follows:
    - Set one of the allowed `fiper.system.drm` options to `lsf`. For example:

      ```
      fiper.system.drm=lsf
      ```

      or

      ```
      fiper.system.drm.2=lsf
      ```

    - Set the `fiper.system.bsubpath` to `<lsf_install_dir>/bin/bsub.exe`. Be sure to use the full path with forward slashes (/), not backslashes (\). For example:

      ```
      fiper.system.bsubpath=C:/LSF_7.0/bin/bsub.exe
      ```

11. Save and close the file.
12. Create a connection profile for the SIMULIA Execution Engine.

    For more information, see *Creating the Connection Profile File*.

Important: The server setting in the profile must exactly match the name of the SIMULIA Execution Engine. For example, if the server name is `system`, the server setting must be `system`, not `system.domain.com`.

13. Restart the LSF cluster (from the LSF master: `lsadmin reconfig`, `badmin reconfig`, `badmin mbdrestart`).
14. If you are installing on a Linux system, create a `$HOME/.fiper.sh` file, and set up the LSF environment in this new file by adding the following line (i.e., replacing `$LSF_ENVDIR` with the path to your LSF configuration directory):

```
. $LSF_ENVDIR/profile.lsf
```

Important: You cannot use environment variables in this file setting to locate your LSF installation.

15. Start the SIMULIA Execution Engine and SIMULIA Execution Engine stations as you usually do.

Creating the seeadmin User on Windows

Before installing the SIMULIA Execution Engine on Windows, you must set up and configure a default local user account to be used with the installation. The purpose of this section is to walk you through the steps of creating this user. This local user requires special and specific settings in the Windows environment. Failure to set up this user properly will result in a failed installation.

Creating the seeadmin User

You must set up a local user account, with special and specific settings in the Windows environment, prior to installing an application server and a database.

The following procedures are written for use with Windows Server 2003. The procedures used for Windows Server 2008 are similar, although some Control Panel entries are different.

Important: Use this procedure only on Windows Server systems that are *not* domain controllers. Server software such as the SIMULIA Execution Engine should never be installed on a computer that acts as a Windows domain controller. For more information on whether your system is a domain controller, contact your local system administrator.

Before you begin: You must have Administrative privileges to add the new user to your system.

1. Right-click the **My Computer** icon on your desktop, and click **Manage**.

   The **Computer Management** dialog box appears.

2. Click **Local Users and Groups** on the left side of the dialog box.

   Important: If this option is disabled and a message appears on the right side of the dialog box stating that the computer you are using is configured as a domain controller, do not continue with this procedure. The SIMULIA Execution Engine should not be installed on a domain controller.

   User and group information appears on the right side of the dialog box.

3. Double-click the **Users** folder on the right side of the dialog box to display all current users on the system.
4. Right-click the **Users** folder on the left side of the dialog box, and select **New User...**.

   The **New User** dialog box appears.

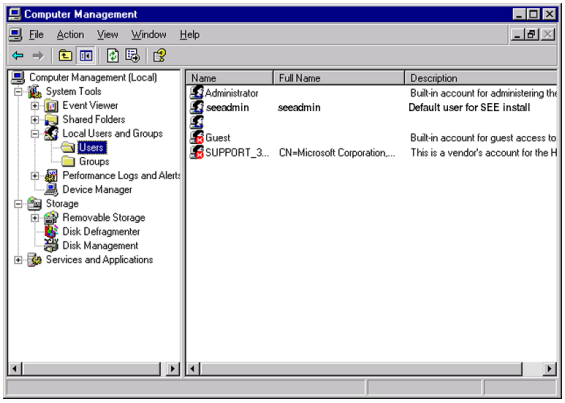
   You can choose any name you want for the user (be sure to write it down). In the instructions in this manual, the default user name `seeadmin` is used.

   If your SIMULIA Execution Engine will use a DB2 database, the user name must consist only of letters, the underscore character (_), and numbers. In addition, it must not match an SQL reserved word or start with any of the following case-sensitive prefixes: IBM, SYS, SQL, or DBM.

   Once the user name is set, it cannot be changed. The SIMULIA Execution Engine database will not run correctly if the user name is changed during or after installation.
5. Type the name of the new user in the **User name** text box; for example, `seeadmin`.
6. If desired, type a more descriptive string in the **Full Name** text box.
7. Type a description in the corresponding text box; for example, `Default user for SEE installation`.
8. Type a password for the user account in the **Password** text box, and retype the password in the **Confirm password** text box.

   The password must conform to any local security policies; otherwise, the creation of the password will fail with a generic message stating that the user cannot be created. For more information, see *Checking Password Policies*.
9. Set the password options below the **Confirm password** text box, as desired.
10. Click **Create**.
11. Click **Close**.

    You are returned to the **Computer Management** dialog box, and the new user is displayed.



12. To set the properties for the new user, right-click the new user, and select **Properties**.

    The **Properties dialog** box appears.

13. Select the **Member Of** tab.
14. Click **Add**.

    The **Select Groups** dialog box appears.

15. Click **Advanced**.
16. Click **Find Now**.

    Double-click **Administrators** from the list at the bottom of the dialog box.

    Administrator privileges will be assigned to the new user. These privileges are for SEE installation purposes only. Once installation is complete, you should convert the user to a regular (non-administrative) user.

17. Click **OK**.

    You are returned to the **Properties** dialog box.

18. Click **OK**.

    You are returned to the **Computer Management** dialog box.

19. Close the **Computer Management** dialog box.
20. If necessary, configure the new user as described in *Setting the seeadmin User Privileges*.

Setting the seeadmin User Privileges

You may need to configure the new `seeadmin` user account to function in the Windows environment.

The following guidelines should be used when determining if the `seeadmin` user has sufficient privileges:

- Verify that the `seeadmin` user is a member of the Administrators group as described in *Creating the seeadmin User*. This group assignment is necessary to administer DB2, WebSphere, and the SIMULIA Execution Engine database.
- If the `seeadmin` user installs DB2, the `seeadmin` user will automatically be granted all permissions needed to administer DB2. In this case you must log in as this `seeadmin` user to install DB2. Changing the security context to this user via the `runas` utility is not sufficient. If DB2 was installed by a different user, the `seeadmin` user must be added to the local group `DB2ADMNS`, created by the DB2 installation, to be able to create and administer the SIMULIA Execution Engine database.
- If the `seeadmin` user is going to be used to start a SIMULIA Execution Engine station when the SIMULIA Execution Engine database has Run-As security enabled, the `seeadmin` user must be granted the privilege `Replace a process level token`. For more information, see *Configuring Station (Run-As) Security*.
- If you add a user to a new group or grant it a privilege, the change does not take effect until you log out and log in again as that user. Groups assignments and privileges are checked only during the logon process.

Checking Password Policies

You should check your local password settings to make sure they conform to the Windows-based computer policy regarding the minimum length and complexity.

1. Click **Start** / **Control Panel** / **Administrative Tools** / **Local Security Policy**.

   The **Local Security Settings** dialog box appears.

2. Navigate to **Account Policies** / **Password Policy**.

   The settings for this option appear on the right side of the dialog box.

3. Check the value of the **Minimum password length** setting.

4. If the **Password must meet complexity requirements** option is enabled, the following restrictions are in place:

   - Your password must contain characters from at least three of the following four groups:
     - upper case letter
     - lower case letter
     - number
     - punctuation character
   - Your password must be a least six characters long.
   - Your password must not contain three consecutive characters from the user name or full name settings. For example, if the user name is `seeadmin`, the password could not contain `see`, `adm`, etc.

Creating an Oracle Database for the SIMULIA Execution Engine

Before the SIMULIA Execution Engine is configured, you must create a SIMULIA Execution Engine database. If you already have Oracle installed, you need to use the Database Configuration Assistant to create the database.

Creating the Database in Oracle 11gR2

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:

- Windows: **UTF8 - Unicode 3.0 UTF-8 Universal character set**
- 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 (for example, `https://hostname.yourcompany.com:1158/em`).

39. Click **Exit**.

For more information on using the Enterprise Manager, see *Initializing an Oracle Database*.

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*.

```
licusage --log logfile --logtype {dsls | flexnet} [--start
start_date] [--end end_date] [--type {export | query|]
--accessor accessors [--list_accessors] [--filter filter]
[--list_filters] [--sort sorter] [--list_sorters] [--aggregator
aggregators] [--list_aggregators] [--duration bucket_size]
[--output report_file] [--format output_format] [--list_formats]
[--custom customization_script] [--help]
```

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:

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

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 `c:\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:

- `dd-mmm-yyyy_hh:mm:ss`

  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 (*hh*) 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:

- `dd-mmm-yyyy_hh:mm:ss`

  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 (*hh*) 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:valuepairs* 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
```

Checkout Date	Username	Feature	Quantity
2012-Sep-07 12:03:40	bbaggins	IGF	10

Checkout Date	Username	Feature	Quantity
2012-Sep-07 12:03:52	gandalf	IGF	20
2012-Sep-07 12:03:53	fbaggins	IGF	20
2012-Sep-07 12:03:56	gollum	IGD	1

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
```

Bucket	Average
2012-Dec-06 12:37:53	25
2012-Sep-06 13:37:53	19
2012-Sep-06 14:37:53	8
2012-Sep-06 15:37:53	18
2012-Sep-06 16:37:53	13
2012-Sep-06 17:37:53	34

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
```

Bucket	Number of Items	Average	Maximum
2012-Sep-19 12:37:53	108	25	80
2012-Sep-19 13:37:53	98	19	80
2012-Sep-19 14:37:53	46	8	66
2012-Sep-19 15:37:53	114	18	50
2012-Sep-19 16:37:53	129	13	50
2012-Sep-19 17:37:53	74	34	66
2012-Sep-19 07:37:53	49	49	66
2012-Sep-19 08:37:53	11	52	66
2012-Sep-19 09:37:53	2	50	50

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
```

Bucket	Number of Items	Average	Maximum	Maximum
2012-Sep-19 12:37:53	108	25	80	1497
2012-Sep-19 13:37:53	98	19	80	1220
2012-Sep-19 14:37:53	46	46	66	77
2012-Sep-19 15:37:53	114	114	50	482
2012-Sep-19 16:37:53	129	129	50	449
2012-Sep-19 17:37:53	74	74	66	3552
2012-Sep-19 07:37:53	49	49	66	3272

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 WebSphere:

- **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:

  - **Windows Server 2003**: `C:\Documents and Settings\<user_name>\local settings\temp\<station_name>`
  - **Windows Server 2008**: `C:\Users\<user_name>\AppData\Local\Temp`
  - **Linux**: `/tmp/<station_name>`

- **SystemOut.log**. This WebSphere-generated file contains port information for accessing the Administrative console. It may also help when debugging WebSphere errors. It is typically referred to as the SIMULIA Execution Engine log file.

  The default location for this log file is as follows:

  `<websphere_install_dir>\AppServer\profiles\<profileName>\logs\server1\`

  This directory also contains numerous other log files that may assist you in diagnosing problems with your SIMULIA Execution Engine.

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 WebSphere*.

Configuring the Windows Firewall for WebSphere

You need to open several ports to ensure that your WebSphere-based SIMULIA Execution Engine will function correctly behind the Windows Firewall.

Important: These instructions assume that you are using WebSphere's default port settings. To ensure that your port settings match those in this procedure, contact your local WebSphere 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. Type the following information in the corresponding text boxes:

   - Name: `WAS_bootstrap_address`
   - Port number: `2809`

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. Repeat step 5 through step 7 for the following ports:

- Name: `WAS_orb_listener_address`
- Port number: `9100`
- Name: `WAS_sib_endpoint_address`
- Port number: `7276`
- Name: `WAS_webtop`
- Port number: `9080` (no SIMULIA Execution Engine security) or `9443` (SIMULIA Execution Engine security enabled)
- Name: `WAS_csiv2_ssl_serverauth_listener_address`
- Port number: `9403`

Note: This final port number exception is necessary only if SIMULIA Execution Engine security has been enabled.

9. Click **OK** to close the **Windows Firewall** dialog box.
10. 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.

Fixing Network Connection Problems

Because of a firewall issue or an IP configuration issue, you may be unable to connect SIMULIA Execution Engine stations and Isight gateways that are installed on separate computers to the SIMULIA Execution Engine WebSphere-based computer.

The following error message may appear:

```
com.engineous.sdk.pse.ServerNotAvailableException: ACS server
"HOST" is not available or is not responding. Contact the ACS
administrator.
```

To determine the cause of the error, you will need to review the full error traceback for a block of text similar to the following:

```
=== Exception Traceback ===
javax.naming.ServiceUnavailableException: A communication
failure occurred while attempting to obtain an initial context

with the provider URL: "IIOP://computername:2809".  Make sure
that any bootstrap address information in the URL is correct
and that the target name server is running.  A bootstrap
address with no port specification defaults to port 2809.
Possible causes other than an incorrect bootstrap address or
unavailable name server include the network environment and
workstation network configuration. [Root exception is
org.omg.CORBA.TRANSIENT: java.net.UnknownHostException:
companyname.com:host=computername.companyname.com,port=9100
vmcid: 0x4942f000  minor code: 3586  completed: No]
```

The key is the last section beginning with "`[Root exception`"

If the text refers to an `UnknownHostException` (as in the example above), the problem is the IP Domain Name configuration. See *IP Configuration Workaround* for more information.

If the text refers to a `ConnectException` (as in the example text shown below) and you cannot connect a station to the SIMULIA Execution Engine on the SIMULIA Execution Engine computer, the Windows firewall is enabled and must be disabled.

```
java.net.ConnectException: Connection timed out:
connect:host=computername.companyname.com,port=2809
```

IP Configuration Workaround

The cause of the `UnknownHostException` is a problem with the IP Domain Name Configuration.

If you run `ipconfig` from a Windows command prompt, the results will be similar to the following:

```
C:\home>ipconfig
Windows IP Configuration
Ethernet adapter Local Area Connection:
Connection-specific DNS Suffix  . : engineous.com
IP Address. . . . . . . . . . . . : xxx.xx.xxx.xx
Subnet Mask . . . . . . . . . . . : xxx.xxx.xxx.x
Default Gateway . . . . . . . . . : xxx.xx.xxx.x
```

The important part is the `Connection-specific DNS Suffix`, which indicates that the full name of the computer *computername* is actually `computername.engineous.com`.

The cause of the `UnknownHostException` is that the DNS Suffix is incorrect. When WebSphere is installed, it assumes its host name is the fully qualified name, and there is no way to override this setting in the installer. However, there is a workaround: you can configure the host name for the connections to WebSphere from the WebSphere console.

1. Log on to the WebSphere console.
2. Click **Servers** on the left side of the console.
3. Click **Application Servers**.
4. Click the **Server1** link on the right side of the console.
5. Click the **Ports** link in the **Communications** section on the right side of the console.

   A list of TCP/IP ports appears. In the **Host** column, the fully qualified host name appears as the entry. An asterisk (*) appears as the entry in some cases.

6. Click the **Port Name** of each row with the full host name in the **Host** column.

   A port configuration edit page appears.

7. Edit the fully qualified name to the simple name in the **Host** text box.
8. Click **OK**.
9. Repeat this procedure for all entries that have the fully qualified name in the **Host** column.
10. Verify that all host names that are not “*” are the simple name.
11. Click **Save** near the top of the console; then, click **Save**.
12. Log out of the WebSphere console.

13. Stop and restart WebSphere. For more information about stopping and restarting WebSphere, see *Restarting the SIMULIA Execution Engine in WebSphere*.

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.

This temporary directory is set during the installation of the SIMULIA Execution Engine, as described in *Installing the SIMULIA Execution Engine Server*.

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.

For more information on the settings in the `acs.properties` file, see *Understanding the acs.properties File Settings*.

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.

- WebSphere LDAP security has the password for the “Bind Distinguished Name”. If this password changes (based on your network’s LDAP settings), you must change the password on the Security page of the WebSphere console *before* changing the password on the account, and then restart WebSphere after the account’s password is updated. If you change the password on the account before changing the password in WebSphere, you will not be able to log on to the WebSphere console. For more information, contact your local WebSphere administrator.

  Symptom: If the LDAP password in WebSphere is wrong and WebSphere security is enabled, you will not be able to access the WebSphere console. Instead, you will receive a message stating that your password is incorrect. You may be able to log into an Isight interface (such as the Design Gateway) and connect to the SIMULIA Execution Engine, but the user you specified will be listed as UNAUTHENTICATED and none of the SIMULIA Execution Engine features will work correctly.

- The password of the SIMULIA Execution Engine Database owner is kept in a WebSphere J2C Authentication Alias usually named `SEEDB2Auth` or `SEEOracleAuth` (or `fiperDB2Auth` or `fiperOracleAuth`). This password must be for the user ID that is passed to the `createtables` script when creating the database. If the password for this user is changed, the SIMULIA Execution Engine will lose access to the database. Changing the password for this user will not affect access to the WebSphere console.

  To update this password, follow these steps:

  1. Access the WebSphere console as described in *Starting WebSphere and Determining Server Port Numbers*.
  2. Click **Security** on the left side of the console.
  3. Click **Secure administration, applications, and infrastructure**.
  4. Expand the **Java Authentication and Authorization Service** option in the **Authentication** area on the right side of the console.
  5. Click the **J2C authentication data** link.
  6. Click the link in the **Alias** column that corresponds to the Authentication Alias that you must update.
  7. Alter the password in the corresponding text box.
  8. Click **OK**.
  9. Click the **Save** link to save your configuration changes.
  10. Click **Resources** on the left side of the console.
  11. Expand the **JDBC** option.
  12. Click the **Data sources** link.
  13. Click the check box in the **Select** column that corresponds to the two SIMULIA Execution Engine data sources (**Fiper NonXA Data Source** and **Fiper XA Data Source**).

14. Click **Test connection**.

Two messages appear, telling you that the test connection was successful.

Symptom: If this password is wrong, the SIMULIA Execution Engine loses connection to the database. You can log on to the SIMULIA Execution Engine, but almost all operations will fail.

- The password for the SIMULIA Execution Engine administrative user (usually `fiperacs` or `seeadmin`) is stored in WebSphere in two places:

  1. J2C authentication alias `SEEDB2Auth` or `SEEOracleAuth` (or `fiperDB2Auth` or `fiperOracleAuth`). Refer to the previous entry for details on how to alter this password information.

  2. SIMULIA Execution Engine Application Run-As Role User. To fix this setting, alter the password information as described in *Configuring J2EE RunAs Security*.

  Note: You will have to restart the SIMULIA Execution Engine Application for this change to take effect, and you may have to restart WebSphere.

  Symptom: If the password for the SIMULIA Execution Engine administrative user is wrong in either the WebSphere J2C Authentication Alias or the WebSphere J2EE RunAs security, jobs fail to complete. The WebSphere `SystemOut.log` file shows errors indicating that the user is `UNAUTHENTICATED` even though all users are logged in. There may also be errors indicating that `user 'seeadmin' is not known`.

- 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.

DB2 Package Problem

If the WebSphere log shows a “could not find package” message, you can use the procedure described in this section to fix this problem.

1. Click the **Start** button, point to **All Programs** / **IBM DB2 / Command Line Tools**, and click **Command Window**.

   The **DB2 CLP** command window appears.

2. Execute the following commands:
   - **a.** `db2 terminate`
   - **b.** `db2 connect to fiper`
   - **c.** `db2 bind <sqllib_install_dir>\bnd\@db2ubind.lst grant public`
   - **d.** `db2 bind <sqllib_install_dir>\bnd\@db2cli.lst grant public`
   - **e.** `db2 terminate`
   - **f.** `db2stop force`
   - **g.** `db2start`

     Each of these commands issues a message upon successful completion.

3. Close the **DB2 CLP** command window.

Copying the WebSphere JAR Files

The WebSphere client JAR files are not distributed with the SIMULIA Execution Engine.

If your SIMULIA Execution Engine server is deployed and running on a WebShere application server, Isight and the SIMULIA Execution Engine stations cannot connect to the server until you copy a set of Websphere JAR files into each Isight and SIMULIA Execution Engine station software installation.

1. After you have installed Isight and the SIMULIA Execution Engine stations, navigate to the following directory:
   - Windows: `c:\Program Files\IBM\WebSphere\AppServer\runtimes\`
   - Linux: `/opt/IBM/WebSphere/AppServer/runtimes/`
2. Copy the following files:
   - `com.ibm.jaxrs.thinclient_8.5.0.jar`
   - `com.ibm.jaxws.thinclient_8.5.0.jar`
   - `com.ibm.ws.ebj.thinclient_8.5.0.jar`
   - `com.ibm.ws.jpa.thinclient_8.5.0.jar`
   - `com.ibm.ws.messagingClient.jar`
   - `com.ibm.ws.orb_8.5.0.jar`
   - `com.ibm.ws.sib.client.thin.jms_8.5.0.jar`
   - `com.ibm.ws.webservices.thinclient_8.5.0.jar`
   - `com.ibm.xml.thinclient_8.5.0.jar`
   - `endorsed/endorsed_apis-8.5.0.jar`
3. Do the following on each computer that Isight or SIMULIA Execution Engine stations installed:
   a. Navigate to the following directory:
      `<Isight_install_directory>/<operating_system>/reffiles/ejbclient/websphere/lib`
   b. Paste the copied files into the above directory.

Backup and Restore Procedures

You can back up your SIMULIA Execution Engine data and restore the SIMULIA Execution Engine to the point of a backup.

Backing up SIMULIA Execution Engine Data

To reliably back up your SIMULIA Execution Engine, you must back up two specific data repositories: the SIMULIA Execution Engine database and the SIMULIA Execution Engine File Manager directory.

To ensure a consistent backup, both the SIMULIA Execution Engine database and the SIMULIA Execution Engine File Manager directory should be backed up simultaneously while the SIMULIA Execution Engine is either shutdown or idle (running no jobs). These two repositories contain all the persisted data of the SIMULIA Execution Engine. Restoring them from a backup returns all the SIMULIA Execution Engine historical data to the state it had at the time of the backup. It is possible to perform a “hot backup” of these repositories. However, there is a risk that currently running jobs may not be in a consistent or useful state if the data are later restored.

The backup of these repositories does *not* back up the configuration of the SIMULIA Execution Engine, including:

- the configuration settings in the `acs.properties` file,
- the SIMULIA Execution Engine code itself,
- the application server or other middleware,
- the server deployment configuration, and
- the server configuration settings for performance, security, monitoring, etc.

Normal full-system backups of the SIMULIA Execution Engine file system should be used to back up the full state of the computer as well as all installed software.

The actual physical process of copying and archiving the repositories is outside the scope of SIMULIA Execution Engine itself. Normal backup and archiving procedures should be used as appropriate for the repository. For the SIMULIA Execution Engine database, you should use the database vendor tools for database backup and archiving. For the SIMULIA Execution Engine File Manager directory, you should use normal operating system file backup and restore tools.

The location of the repositories depends on the configuration of the SIMULIA Execution Engine.

- **Database location.** The database location is determined by the configuration of the data sources in the SIMULIA Execution Engine application deployment. All SIMULIA Execution Engine database operations are performed on the database named in those datasource configurations. In general, the configuration specifies the computer on which the database resides, and information that identifies the SIMULIA Execution Engine database on that computer.
- **SIMULIA Execution Engine File Manager location.** The SIMULIA Execution Engine File Manager directory is a directory in the file system of the computer running the SIMULIA Execution Engine application server. The computer running the application server is usually not the same as the computer on which the database resides. The name and location of this directory is set during the SIMULIA Execution Engine installation. You can find the location by examining the `acs.properties` file at the top level of the SIMULIA Execution Engine installation directory. The following property setting shows the directory name and location:

  ```
  fiper.system.filemgr.rootFilePath=
  ```

  Under the specified directory will be a number of subdirectories that are created and maintained by the SIMULIA Execution Engine. This main directory and all subdirectories and files within it must be backed up.

Restoring the SIMULIA Execution Engine

You can restore the SIMULIA Execution Engine to the point of a backup.

1. Verify that the SIMULIA Execution Engine is shutdown. It is not sufficient for it to be idle.
2. Restore the SIMULIA Execution Engine database and the SIMULIA Execution Engine File Manager repositories from the backup media. In both cases, the restored data must replace the existing data—the data must *not* be merged or combined in any way. For the SIMULIA Execution Engine File Manager, it is not sufficient to copy from the backup media to the SIMULIA Execution Engine file system. The existing directory must be deleted (or emptied) before copying from the backup media. Likewise for the SIMULIA Execution Engine database, the backup database image must completely replace the existing database. It is not sufficient to just restore the tables—all existing tables in the database must be dropped before restoring from the backup image. Most database backup tools

provide options for this task. For more information, contact your local database administrator.

3. If the backup was made with a prior release of the SIMULIA Execution Engine, any manual data migration steps that were needed when the newer releases were installed must be performed now. This action brings the restored data into the correct format for the currently installed release of the SIMULIA Execution Engine.

   Such manual migration steps are not common.

4. Restart the SIMULIA Execution Engine.

   If any jobs were running at the time of the backup, they will be marked as `Cancelled` when the SIMULIA Execution Engine starts (the SIMULIA Execution Engine log will show the list of jobs that were so affected). If the backup was from a prior release of the SIMULIA Execution Engine, some automatic data migration may be performed when the SIMULIA Execution Engine starts. Any migration that is done is noted in the SIMULIA Execution Engine logs, but no manual actions are necessary for this process to occur.

5. If the backup was from a prior release, the SIMULIA Execution Engine Library, which is stored in the database, will have back-level versions of all the components. You need to execute the `publishall` command to re-publish all the SIMULIA Execution Engine components from the current release into the library. This step must be performed before any new job is submitted. If there are any custom components required, those must also be re-published to fully populate the SIMULIA Execution Engine Library.

   For more information, see *Publishing to the Library*.

   Otherwise, the SIMULIA Execution Engine is now ready for use.

Index

C

Configuring WebSphere for FIPER
    Defining Services 54, 125, 127
Connection profile file
    Creating 89
Creating the Database
    Oracle 11g 211

J

Java 7 6

L

license utility 215

P

publishall command 91

R

report utility 215

S

SIMULIA Execution Engine stations 2, 145

W

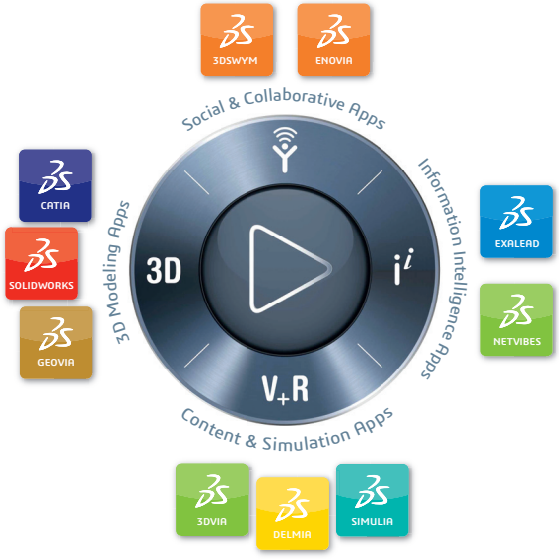
WebSphere
    Automated Prerequisites 81
    Automated Scripts 83
    params.txt:Automatically Configuring 81
    Scripts:Limitations 84

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' **3D**EXPERIENCE 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 **3D**EXPERIENCE® Platform powers our brand applications, serving 12 industries, and provides a rich portfolio of industry solution experiences.

Dassault Systèmes, the **3D**EXPERIENCE 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 170,000 customers of all sizes in all industries in more than 140 countries. For more information, visit **www.3ds.com**.



DASSAULT SYSTEMES | The **3D**EXPERIENCE Company

Europe/Middle East/Africa
Dassault Systèmes
10, rue Marcel Dassault
CS 40501
78946 Vélizy-Villacoublay Cedex
France

Asia-Pacific
Dassault Systèmes
Pier City Shibaura Bldg 10F
3-18-1 Kaigan, Minato-Ku
Tokyo 108-002
Japan

Americas
Dassault Systèmes
175 Wyman Street
Waltham, Massachusetts
02451-1223
USA