



SIMULIA Execution Engine 5.7

Installation and Configuration Guide

WebLogic and Oracle



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.

© Dassault Systèmes, 2012.

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 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 the Isight and the SIMULIA Execution Engine 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.

Locations

SIMULIA welcomes any suggestions for improvements to Isight software, the support program, or documentation. We welcome 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>.

Headquarters

SIMULIA Worldwide Headquarters	Rising Sun Mills, 166 Valley Street, Providence, RI 02909-2499, Tel: +1 401 276 4400, Fax: +1 401 276 4408, simulia.support@3ds.com , http://www.3ds.com/products/simulia
SIMULIA European Headquarters	Stationsplein 8-K, 6221 BT Maastricht, The Netherlands, Tel: +31 43 7999 084, Fax: +31 43 7999 306, simulia.europe.info@3ds.com

Dassault Systèmes' Centers of Simulation Excellence

United States	Fremont, CA, Tel: +1 510 794 5891, simulia.west.support@3ds.com
	West Lafayette, IN, Tel: +1 765 497 1373, simulia.central.support@3ds.com
	Northville, MI 48167, Tel: +1 248 349 4669, simulia.greatlakes.info@3ds.com
	Woodbury, MN 55125, Tel: +1 612 424 9044, simulia.central.support@3ds.com
	Woodbury, MN 55125, Tel: +1 612 424 9044, simulia.central.support@3ds.com
	Mason, OH 45040, Tel: +1 513 275 1430, simulia.central.support@3ds.com
	Warwick, RI 02886-0201, Tel: +1 401 739 3637, simulia.east.support@3ds.com
	Lewisville, TX 75067, Tel: +1 972 221 6500, simulia.south.info@3ds.com
Australia	Richmond VIC, Tel: +61 3 9421 2900, simulia.au.support@3ds.com
Austria	Vienna, Tel: +43 1 22 707 200, simulia.at.info@3ds.com
Benelux	Maarsse, The Netherlands, Tel: +31 346 585 710, simulia.benelux.support@3ds.com
Canada	Toronto, ON, Tel: +1 416 402 2219, simulia.greatlakes.info@3ds.com
China	Beijing, P. R. China, Tel: +8610 6536 2288, simulia.cn.support@3ds.com
	Shanghai, P. R. China, Tel: +8621 3856 8000, simulia.cn.support@3ds.com
Finland	Espoo, Tel: +358 40 902 2973, simulia.nordic.info@3ds.com
France	Velizy Villacoublay Cedex, Tel: +33 1 61 62 72 72, simulia.fr.support@3ds.com
Germany	Aachen, Tel: +49 241 474 01 0, simulia.de.info@3ds.com
	Munich, Tel: +49 89 543 48 77 0, simulia.de.info@3ds.com
India	Chennai, Tamil Nadu, Tel: +91 44 43443000, simulia.in.info@3ds.com
Italy	Lainate MI, Tel: +39 02 3343061, simulia.ity.info@3ds.com
Japan	Tokyo, Tel: +81 3 5442 6302, simulia.jp.support@3ds.com

	Osaka, Tel: +81 6 7730 2703, simulia.jp.support@3ds.com
Korea	Mapo-Gu, Seoul, Tel: +82 2 785 6707/8, simulia.kr.info@3ds.com
Latin America	Puerto Madero, Buenos Aires, Tel: +54 11 4312 8700, Horacio.Burbridge@3ds.com
Scandinavia	Stockholm, Sweden, Tel: +46 8 68430450, simulia.nordic.info@3ds.com
United Kingdom	Warrington, Tel: +44 1925 830900, simulia.uk.info@3ds.com

Authorized Support Center

Italy	Exemplar srl, Turin, Tel: +39 011 564 51 17, Luca.Fattore@examplarsolutions.it
-------	---

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 at the top level of your Isight installation directory.
 - 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.....	6
About the SIMULIA Execution Engine Environment.....	6
Basic Installation Steps for the SIMULIA Execution Engine	7
About Licensing and Your License File.....	8
Installing as an Administrator (Windows) or Non-Root User (UNIX/Linux).....	8
Assumed Default Windows Settings.....	9
Determining the Start Menu Setting.....	9
Determining the Control Panel Setting.....	10
Determining Folder and File Extension Options.....	11
Opening a Command Prompt Dialog Box on Windows.....	11
What's New?.....	13
Prerequisites.....	14
Software Requirements.....	14
Operating System.....	14
Database and Java EE Application Server.....	15
Web Browser and PDF Viewer Software.....	15
Hardware Requirements.....	15
Updating the Windows Firewall.....	15
Disabling Real-Time Virus Scanning.....	16
Migrating to SIMULIA Execution Engine 5.7	16
Installing the SIMULIA Execution Engine Software on Windows.....	17
Installing the SIMULIA Execution Engine	17
Uninstalling the SIMULIA Execution Engine.....	19
Closing the SIMULIA Execution Engine Applications.....	19
Removing the SIMULIA Execution Engine Software.....	20
Deleting Temporary Directories and Files.....	20
Installing a SIMULIA Execution Engine Station on Windows.....	22
Installing the Station Software.....	22
Installing a SIMULIA Execution Engine Station as a Service Manually.....	24
Installing a SIMULIA Execution Engine Station as a Service.....	24
Starting a SIMULIA Execution Engine Station as a Service.....	24
Stopping a SIMULIA Execution Engine Station Service.....	25
Uninstalling a SIMULIA Execution Engine Station Service.....	25
Uninstalling a SIMULIA Execution Engine Station.....	26
Stopping the Station.....	26
Removing the SIMULIA Execution Engine Station Software.....	27
Deleting Temporary Directories and Files.....	27
Installing the SIMULIA Execution Engine Software on UNIX and Linux.....	29
Installing the SIMULIA Execution Engine Software.....	29
Installing the SIMULIA Execution Engine Non-Interactively.....	31
Uninstalling the SIMULIA Execution Engine and License Server.....	32
Uninstalling the SIMULIA Execution Engine and License Server.....	32
Stopping the SIMULIA Execution Engine	32

Removing the SIMULIA Execution Engine Software.....	32
Deleting Temporary Directories and Files.....	33
Installing a SIMULIA Execution Engine Station on UNIX and Linux.....	35
Before You Begin.....	35
Shared Network Install.....	35
Supported Platforms and Patch Information.....	35
Root Privileges.....	36
Installing the Station Software.....	36
Enabling the SIMULIA Execution Engine Station Security Feature (Run-As).....	39
Installing a SIMULIA Execution Engine Station as a Service Manually.....	39
Uninstalling a SIMULIA Execution Engine Station.....	40
Stopping the SIMULIA Execution Engine Station.....	40
Removing the SIMULIA Execution Engine Station Software.....	40
Deleting Temporary Files and Other Files.....	41
Initializing the SIMULIA Execution Engine Database.....	43
Initializing an Oracle Database.....	43
About Oracle URLs and Port Numbers.....	43
Creating Tablespaces and Defining User Information.....	43
Updating the Database.....	46
Creating the Database Tables.....	47
Configuration.....	48
Configuring Your FLEXnet License to Work with a Windows Firewall.....	48
Setting Station Execution Permissions for the Excel and Word Components.....	48
Setting Word and Excel Privileges for Stations Installed as a Service.....	49
Configuring WebLogic.....	51
About Configuring WebLogic.....	51
About the Deployment Configuration and Options.....	52
Configuring the SIMULIA Execution Engine	55
Creating the WebLogic Domain Using the SIMULIA Execution Engine Administration Server Template.....	55
Editing the Environment File.....	57
Starting the Administration Server.....	59
Configuring Client Authentication.....	60
Defining a Managed Server.....	67
Creating WebLogic Domains for Remote Managed Servers.....	70
Starting the Managed Servers.....	71
Deploying the SIMULIA Execution Engine EAR File.....	72
Deploying the WebTop, WebDashboard, or Federation Applications.....	74
Configuring the WebTop and WebDashboard Applications.....	75
Restarting the SIMULIA Execution Engine in WebLogic.....	77
About Starting the SIMULIA Execution Engine Server.....	77
Stopping the SIMULIA Execution Engine Server.....	78
Configuring the Connection Profile and Preloading the Library.....	79
Configuring the Connection Profile.....	79
Publishing to the Library.....	81
Understanding the acs.properties File Settings.....	82
Configuring Security.....	85
About SIMULIA Execution Engine Security.....	85

Configuring SIMULIA Execution Engine Security.....	86
About Client Authentication.....	86
About SIMULIA Execution Engine Access Control Lists.....	86
Configuring Station (Run-As) Security.....	88
About Station Run-As Security.....	88
About User Credential Encryption.....	89
About Securing the SIMULIA Execution Engine Station File System.....	90
About Run-As Security Limitations.....	91
Configuring the Run-As Feature.....	92
Using SIMULIA Execution Engine Interfaces.....	96
Using the SIMULIA Execution Engine Station.....	96
About the SIMULIA Execution Engine Station Interface.....	96
About Log Message Detail Levels.....	97
About Station Affinities.....	98
Station Status Reporting in the Dashboard and WebDashboard.....	99
About Running Multiple Stations on a Single Host Computer.....	99
Starting a SIMULIA Execution Engine Station.....	99
Shutting Down a SIMULIA Execution Engine Station.....	101
Restarting a SIMULIA Execution Engine Station Remotely.....	101
Configuring SIMULIA Execution Engine Station Properties.....	102
Using the Dashboard.....	115
About the Dashboard Interface.....	116
Starting the Dashboard.....	117
Viewing Connection Information.....	120
Viewing Station Information.....	121
Controlling Station Workitems.....	121
Shutting Down, Restarting, or Deleting a Station.....	121
Managing Access Control for SIMULIA Execution Engine Users.....	122
Managing Access Control – System Administration.....	122
Viewing License Usage Information.....	123
Using the WebDashboard.....	123
About the WebDashboard Interface.....	123
Accessing the WebDashboard in a Browser.....	125
Viewing Station Information.....	125
Restarting or Shutting Down a Station.....	126
Viewing Details for the SIMULIA Execution Engine	126
Working with Running Jobs.....	127
Searching for Jobs.....	128
Deleting Non-Running Jobs.....	129
Viewing License Usage Information.....	129
Using the Command Line Client.....	130
Starting the Command Line Client.....	130
Using Distributed Resource Management with the SIMULIA Execution Engine	132
.....	132
Default Fiper DRM included with the SIMULIA Execution Engine	132
LSF DRM and the SIMULIA Execution Engine	132
Mixed-Mode DRM and the SIMULIA Execution Engine	134
Verifying the SIMULIA Execution Engine Configuration.....	135

Configuring LSF for the SIMULIA Execution Engine	136
Creating and Configuring a Local User on Windows.....	139
Creating the New User.....	139
Setting the New User Privileges.....	141
Checking Password Policies.....	141
User Security Identity Configuration	142
Creating an Oracle Database for the SIMULIA Execution Engine	143
Creating the Database in Oracle 10g or 11g.....	143
Basic Troubleshooting.....	146
User Login Names Containing Punctuation.....	146
Log Files for the SIMULIA Execution Engine	146
Configuring the Windows Firewall.....	147
Configuring the Windows Firewall for WebLogic	147
Updating Internet Explorer Settings for the WebLogic Administration Console.....	148
Adding the Console Address to the Trusted Sites List.....	148
Installing Java for Internet Explorer.....	149
Resolving Publishing Errors on Windows.....	149
UNIX/Linux-based SIMULIA Execution Engine Stops Functioning Correctly.....	149
Changing Your SIMULIA Execution Engine Passwords.....	150
WebLogic Client JAR File.....	151
Backup and Restore Procedures.....	152
Backing up SIMULIA Execution Engine Data.....	152
Restoring the SIMULIA Execution Engine	153

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.

A visual layout of the SIMULIA Execution Engine environment is shown in the following figure.

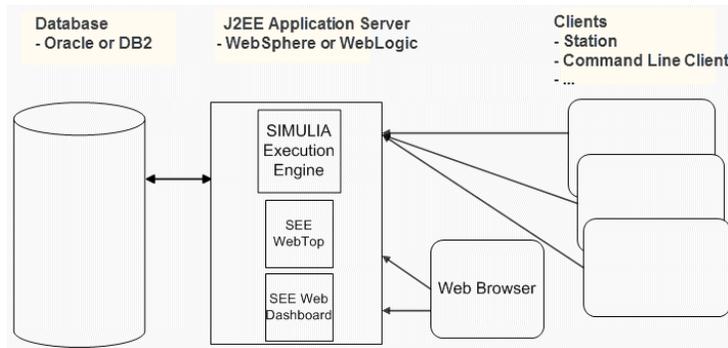


Figure 1: SIMULIA Execution Engine Architecture

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 UNIX and Linux](#).

1. Install the SIMULIA Execution Engine software.

For details, see [Installing the SIMULIA Execution Engine Software on Windows](#) and [Installing the SIMULIA Execution Engine Software on UNIX and Linux](#).

2. Install and upgrade (if necessary) an application server and database combination to run the SIMULIA Execution Engine on. For more information on the supported versions of the third-party software, see [Database and Java EE Application Server](#).

3. Configure the SIMULIA Execution Engine.

For details, see [Configuration](#), [About Configuring WebLogic](#), and [Configuring Security](#).

4. Start the SIMULIA Execution Engine in the application server.

About Licensing and Your License File

During the installation of the SIMULIA Execution Engine, you will be asked to supply information about your license file or license server.

For detailed instructions for installing and configuring a license server, see *Installing or Configuring Licensing* in the Isight Installation Guide.

You can use either FLEXnet licensing or Dassault Systèmes (DS) licensing with the SIMULIA Execution Engine, depending on which type of license key file you have. The SIMULIA Execution Engine installer lets you choose which type of license server software to install. If you are not sure which type of license file you have, ask your DS SIMULIA sales representative.

If you have already received a license file for the SIMULIA Execution Engine, be sure that you know the location of the license file on your (local) computer or the information for the computer acting as a license server. You will have to specify this information during the installation process.

You must have your license key file before installing the DS or FLEXnet license server software, and you must install and start the license server program before installing the SIMULIA Execution Engine.

All licenses for an Isight/SIMULIA Execution Engine job must reside on the same server. For example, a job that contains a DOE component and a Taguchi component must obtain licenses for both components from a single server. The job cannot use a DOE license from one server and a Taguchi license from another server.

Dassault Systèmes Licensing

The Dassault Systèmes license server software package includes the *DS License Server Installation and Configuration Guide* (DSLS .pdf), which contains instructions for configuring the license server and administering licenses. Refer to this guide for all Dassault Systèmes licensing operations. The DSLS .pdf file is installed in the /Docs/ subdirectory of your SIMULIA Execution Engine installation; for example, under C:\SIMULIA\ExecutionEngine\5.7\Docs\.

After the DS license server installation finishes, you must use the DS License Administration Tool to enroll the SIMULIA Execution Engine licenses for individual users; see “Enrolling Product Licenses” in the *DS License Server Installation and Configuration Guide*.



Note: Virtual machines, such as VMware, are not supported for DS licensing. It is not possible to install or run the Dassault Systèmes license server on a virtual machine.

FLEXnet Licensing

FLEXnet licensing for the SIMULIA Execution Engine uses FLEXnet Version 11.6.1, and SIMULIA Execution Engine 5.7 requires lmgrd Version 11.6.1 or higher. If you integrate your SIMULIA Execution Engine license features with other products using a FLEXnet combined license file, ensure that your version of lmgrd meets this requirement.

For more information, refer to the *FLEXnet License Administration Guide* Version 11.6.1, which is available for download from the **Licensing** section at <http://www.3ds.com/products/simulia/support/>.

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

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

Windows

You must use an Administrator user account on Windows to install the SIMULIA Execution Engine because the installer must install Microsoft Visual C++ runtime libraries in the `C:\WINDOWS\winSxS\` directory. Microsoft's User Account Control (UAC) feature enforces this requirement on Windows 7, Windows Vista, and Windows 2008 Server. On Windows XP and Windows 2003 Server, the SIMULIA Execution Engine installer will exit with an error message if you attempt to run it as a non-administrator user. This requirement applies to both 32-bit Windows and 64-bit Windows.

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

UNIX/Linux

Root (superuser) privileges are not required to install the SIMULIA Execution Engine on UNIX or 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/ExecutionEngine`.

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/ExecutionEngine` and makes it writable, a non-root user can also install in `/opt`.

The FLEXnet license server software should be installed similarly, by a non-root user.

The DS license server, however, does require root privileges to install. In addition, you cannot use the `su` command to become root and run the DS licensing installer—this leaves the environment in an incorrect state. Instead, you must log out and log back in as root, or become root using the command "`su -`" (su space dash) to force the environment to be reset as though root had logged in.

After you install the SIMULIA Execution Engine and the license server software, the following (optional) steps will require root privileges:

- Configuring the FLEXnet license server to run as a service; i.e., an application that automatically starts (from `/etc/rc/`) when the computer reboots.
- Configuring a SIMULIA Execution Engine station to run as a service.
- Setting permissions on the program `SMAFITPplaunch` before you can start a station that uses the Run-As security mode. For more information, see [Enabling the SIMULIA Execution Engine Station Security Feature \(Run-As\)](#).

Assumed Default Windows Settings

Since the procedures in this manual assume that you are using the default Windows settings for certain components in your operating system, it is important that you determine which settings you are currently using. In particular, if you are using non-default settings for the Start menu, Control Panel, and folder and file extension options, you may have difficulty following the installation procedures.



Important: It is recommended that you change the settings as described in the following procedures. Doing so will help you avoid any possible confusion when installing the SIMULIA Execution Engine. After installing the software, you can revert back to your old settings.

Determining the Start Menu Setting

To easily install the SIMULIA Execution Engine, it is important to make sure that you are using the default Start menu settings.

The Windows Start Menu is used to access certain configuration features following the installation of the SIMULIA Execution Engine. It is also used when uninstalling the product.

Note: This section does not apply to the Windows 7 or Windows 2008 operating systems. These operating systems do not have multiple settings as described in the following procedure.

1. Right-click the **Start** button, and select **Properties**.

The **Properties** dialog box appears.

2. Verify that the **Start Menu** tab is selected.

Two menu options are listed on this dialog box: **Start menu** and **Classic Start menu**.

3. If necessary, change the current setting to be **Start menu**, which is the default and the recommended setting for installing the SIMULIA Execution Engine.
4. Click **OK**.

Determining the Control Panel Setting

To easily install the SIMULIA Execution Engine, it is important to make sure that you are using the default Windows Control Panel settings.

The Windows Control Panel is used to access certain configuration features following the installation of the SIMULIA Execution Engine. It is also used when uninstalling the product.

Note: This section does not apply to the Windows 2000 and Windows Server 2003 operating systems. These operating systems do not have multiple settings as described in the following procedure.

1. Click the **Start** button, and click **Control Panel**.

This step assumes that you are using the default Windows **Start Menu** option as described in [Determining the Start Menu Setting](#).

The **Control Panel** dialog box appears.

2. Examine the view setting for your **Control Panel**:

- **Windows XP**. Examine the view setting in the upper left corner of the dialog box. One of the following two options will be listed (based on your operating system):
 - **Switch to Category View**. If this option is displayed, the **Control Panel** is currently in **Classic View**. For simplicity, it is recommended that you click this option to switch the **Control Panel** to **Category View**.
 - **Switch to Classic View**. If this option is displayed, the **Control Panel** is configured correctly to correspond with the installation procedures in this manual. No changes are necessary prior to installing the SIMULIA Execution Engine.
- **Windows Vista/Server 2008**. Examine the view setting in the upper left corner of the dialog box. One of the following two options will be listed (based on your operating system):
 - **Control Panel Home**. If this option is displayed in bold text with a small dot to its left, the **Control Panel** is configured correctly to correspond with the installation procedures in this manual. No changes are necessary prior to installing the SIMULIA Execution Engine.
 - **Classic View**. If this option is displayed in bold text with a small dot to its left, the **Control Panel** is currently in **Classic View**. For simplicity, it is recommended that you click the **Control Panel Home** option.

-
- **Windows 7.** Verify that **Category** is selected from the **View by** list in the upper right corner of the dialog box.
3. Close the **Control Panel** dialog box by clicking **X** or selecting **Close** from the **File** menu.

Determining Folder and File Extension Options

To easily install the SIMULIA Execution Engine, it is important to make sure that you are using the recommended Windows folder and file extension options.

There are times, such as when you are removing the SIMULIA Execution Engine software, when you must delete certain temporary directories that Windows hides by default. To delete these directories, you need to ensure your system settings are such that the directories are visible in the file structure.

In addition, since the procedures in this manual refer to file extensions (.exe, .bat, etc.), it is recommended that you ensure your system settings are such that these extensions are displayed.

1. Access the **Control Panel** using one of the following methods, based on your operating system:
 - **Windows XP/Server 2003/Windows Vista/Server 2008/7:** Click **Start**, and click **Control Panel**.
 - **Windows 2000:** Click **Start**, point to **Settings**, and click **Control Panel**.

This step assumes that you are using the default Windows **Start Menu** option as described in [Determining the Start Menu Setting](#).

The **Control Panel** dialog box appears.

2. Access the **Folder Options** dialog box using one of the following methods, based on your operating system:
 - **Windows XP:** Click **Appearance and Themes**, and click **Folder Options**.
 - **Windows 2000/ Windows Server 2003:** Double-click **Folder Options**.
 - **Windows Vista/Server 2008/7:** Click **Appearance and Personalization**, and click **Folder Options**.
3. Click the **View** tab, and click **Show hidden files and folders**.
4. Clear (uncheck) **Hide extensions for known file types**.
5. Click **OK**.

The hidden directories and file extensions will now appear when browsing through your system's directory structure.

Opening a Command Prompt Dialog Box on Windows

At times during the installation of the SIMULIA Execution Engine, you will be asked to open a Windows **Command Prompt** dialog box. This section describes how to open the **Command Prompt** dialog box.

On Windows XP, Windows Vista, Windows Server 2008, and Windows 7, you can access the **Command Prompt** dialog box through the **Start** menu. Click **Start**, point to **All Programs / Accessories**, and click the **Command Prompt** option.

Alternatively, you can follow one of the procedures below:

Windows XP / Windows Server 2003

1. Click the **Start** button, and click the **Run...** option.

The **Run** dialog box appears.

2. Type cmd in the **Open** text box, and click **OK**.

A **Command Prompt** dialog box appears.

Windows Vista / Windows Server 2008

1. Click the **Start** button.
2. Type cmd in the **Start Search** text box, and click **OK**.

A **Command Prompt** dialog box appears.

Windows 7

1. Click the **Start** button.
2. Type cmd in the **Search programs and files** text box, and click **OK**.

A **Command Prompt** dialog box appears.

What's New?

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

Enhanced Functionality

Controlling Station Workitems

New commands allow you to cancel a running workitem in the SIMULIA Execution Engine Dashboard or SIMULIA Execution Engine WebDashboard. This feature lets you find and cancel a workitem that takes too long to finish or appears to be unable to complete.

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 **System Requirements** section at <http://www.3ds.com/products/simulia/support/>.

X Windows Library Required for Linux

The SIMULIA Execution Engine can only be run on 64-bit Linux platforms, but the 32-bit version of the X Windows library `libXext.so.6` is required for installation.

The SIMULIA Execution Engine installer will not run correctly unless the 32-bit version of the X Windows library `libXext.so.6` has been installed in your 64-bit Linux operating system. This library is usually located in the file `/usr/lib/libXext.so.6`.

If this library file is missing from your operating system, the optional **libXext** (32-bit) package must be installed. The required RPM file is `libXext-1.1-3.el6.i686.rpm`, which is located in the `Packages/` directory of the RHEL 6 installation disk.

Solaris Settings

The following Solaris-specific settings and changes are necessary to ensure that the SIMULIA Execution Engine functions properly.

- Java Run-time Environment Requirements

You must install Solaris patches as described on the following web site:

<http://www.oracle.com/technetwork/systems/patches/solaris/index-jsp-139697.html>

- Swap Space Settings

You must verify that the system has swap space equal to three or four times the main memory. For more information on determining your system's current specification, contact your local system administrator.

The large amount of swap space is needed because the SIMULIA Execution Engine station can grow to more than 500 MB. Additional swap space equal to the process size is allocated for a short time every time the program creates a sub-process to execute an external program. This space is never used but must be available. Insufficient swap space will cause SIMULIA Execution Engine jobs to fail sporadically with messages about “insufficient disk space.” Other versions of UNIX/Linux use a different scheme for allocating swap space to processes and can run the SIMULIA Execution Engine with swap space equal to twice main memory.

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 **System Requirements** section at <http://www.3ds.com/products/simulia/support/>.

The SIMULIA Execution Engine software and WebLogic must be installed on the same computer for the SIMULIA Execution Engine to function properly. The recommended hardware configuration is to install the WebLogic application server and the Oracle 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.

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 WebLogic and Oracle. For a list of supported web browsers and versions, see the **System Requirements** section at <http://www.3ds.com/products/simulia/support/>.
- 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 SIMULIA Execution Engine 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 5.7

You do not have to uninstall SIMULIA Execution Engine 5.6 or a 5.6 station before you install SIMULIA Execution Engine 5.7 or a 5.7 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 Software on Windows

This section describes how to install the SIMULIA Execution Engine on supported Microsoft Windows platforms. It also discusses license configuration tasks that you may need to perform to enable your execution environment.

Installing the SIMULIA Execution Engine

The SIMULIA Execution Engine installation wizard, which is included on your SIMULIA Execution Engine DVD, guides you through the process of installing the SIMULIA Execution Engine on your system.

This wizard is used for both standard SIMULIA Execution Engine installations and when installing only a SIMULIA Execution Engine Station. For details on installing a SIMULIA Execution Engine Station, see [Installing a SIMULIA Execution Engine Station on Windows](#).

1. Log in as the Administrator that will install the SIMULIA Execution Engine.

You must be an Administrator user to install the SIMULIA Execution Engine; see [Installing as an Administrator \(Windows\) or Non-Root User \(UNIX/Linux\)](#).

2. Insert the SIMULIA Execution Engine DVD.
3. If you are installing on Windows 7, Windows Vista, or Windows Server 2008, click **Allow** to confirm the execution of the installation program.

Note: If the installer does not start automatically, execute the `setupwin.exe` file located on the top level of the SIMULIA Execution Engine DVD. This installation program is used for both the 32-bit and 64-bit versions of the software. Do not double-click the `setup.jar` file. Double-clicking the `setup.jar` file starts the installer using a local copy of Java, but the installation will later fail.

The installer starts, and the splash screen appears. The installation wizard will guide you through the installation process.

4. Click **Next**.

The installation directory screen appears.

5. Perform one of the following actions:
 - To accept the default directory, click **Next**. Your default directory may vary based on your system settings and permissions.
 - If you want to install the SIMULIA Execution Engine in a different location, click **Browse**, choose a destination folder, and click **Open**.

Click **Next**.

6. Select the product you want to install:
 - **SIMULIA Execution Engine**. This option installs everything needed to run the SIMULIA Execution Engine server and clients.
 - **Station**. This option installs everything needed to run a SIMULIA Execution Engine Station with the SIMULIA Execution Engine. You can install it with the rest of the SIMULIA Execution Engine software, or you can install it separately. If you are installing only a station, and none of the other available options, see [Installing the Station Software](#) for more concise instructions.

-
- **Dassault Systemes License Server.** This option installs the software needed to run a Dassault Systèmes license server. See [About Licensing and Your License File](#) for more details about licensing.
 - **FLEXnet License Server.** This option installs the software needed to run a FLEXnet license server.

Click **Next**.

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

- **FLEXnet License Server**
- **Dassault Systemes License Server**
- **Skip license setup 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 name and (optionally) port number. This information is used to contact the license server and create a client license file that references the server. If you have installed a redundant license server triad, enter the host name and port for all three machines. The port number is found in your license file—the default is to leave this setting empty (the license manager software selects the port number).

For details about the choice of licensing scheme, see [About Licensing and Your License File](#). For complete information about configuring the license server, refer to the appropriate documentation:

- *FLEXnet License Administration Guide* Version 11.6.1, which is available for download from the **Licensing** section at <http://www.3ds.com/products/simulia/support/>.
- *DS License Server Installation and Configuration Guide* (DSLIS .pdf), which is available in the `Doc/` subdirectory of your SIMULIA Execution Engine DVD.

Click **Next**.

8. Enter the following required information, which allows the installation wizard to build a connection profile (.cpr file) for interfaces that will connect to the SIMULIA Execution Engine. If no server name is provided, the installation wizard will not create a .cpr file.

- **Server Name .** The host name of the computer running the SIMULIA Execution Engine. The name of the current host is used if you are installing the SIMULIA Execution Engine.

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.** Select **WebLogic Server 10** from the list.
- **Port Number.** This setting is determined automatically using the default port for the server type. If necessary, you can change this value.

Click **Next**.

9. Click **Next**.

The installation progress screen appears, and the software is installed. Once the installation is complete, an installation summary appears.

10. Review the summary, and click **Next**.

11. Click **Next**.

Information about the .ear files for the SIMULIA Execution Engine appears.

12. Review the information, and click **Finish**.

The installation is complete.

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.

Typically, the uninstallation process removes the license server.

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.



Important: If you have two copies of the same releases of the SIMULIA Execution Engine installed (for example, the initial release and a maintenance release that followed), you must directly access the uninstaller of the release you want to remove. It is recommended that you not use the Windows Control Panel as described in the following procedure. The uninstaller is located in the `_uninst` subdirectory of the main SIMULIA Execution Engine installation directory for the release that you want to remove. Execute the `uninstall.exe` file in this subdirectory.

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 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.
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 using the uninstall wizard, which is similar to the installation wizard used to originally install the software. The wizard will guide you through the entire software removal process.

1. Click **Start**, point to **Control Panel**, and click **Add or Remove Programs**.

The **Add/Remove Programs** dialog box appears.

2. From the list of programs, select **SIMULIA Execution Engine x.x**, and click **Change/Remove**.

The **Welcome** dialog box appears.

3. Click **Next**.

A summary of the uninstallation appears.

4. Click **Next**.

Note: If you are asked about individual files during the uninstall process, click the **Yes to All** option to remove all affected files.

The software is removed. You are informed when the removal is complete.

5. Click **Finish**.

Note: If a message appears informing you that the SIMULIA Execution Engine 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.

6. Close the **Add/Remove Programs** dialog box.

7. You now need to remove any temporary directories or files that the SIMULIA Execution Engine created as described in [Deleting Temporary Directories and Files](#).

Deleting Temporary Directories and Files

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\5.7`, navigate to the `C:\SIMULIA\Execution Engine` directory.
2. Delete the `5.7` directory and all of its contents.
3. Navigate to one of the following directories, based on your operating system, where `<user_name>` is the name of the user who installed and uninstalled the SIMULIA Execution Engine:
 - Windows Server 2003: `C:\Documents and Settings\<user_name>`
 - Windows Server 2008: `C:\Users\<user_name>\`
4. Delete the following items:
 - the `fiper` directory
 - `fiper.preferences` file
5. Navigate to one of the following directories, based on your operating system, where `<user_name>` is the name of the user who installed and uninstalled the SIMULIA Execution Engine:

-
- Windows Server 2003: C:\Documents and Settings*<user_name>*\Local Settings\Temp
 - Windows Server 2008: 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 was specified during the SIMULIA Execution Engine installation.
 8. Delete the entire SIMULIA Execution Engine file manager directory.
 9. Navigate to the SIMULIA Execution Engine temporary directory. This directory location was specified during the SIMULIA Execution Engine installation. By default, this directory is C:\Documents and Settings*<user_name>*\Local Settings\Temp, where *<user_name>* is the name of the user who installed and uninstalled the SIMULIA Execution Engine.
 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. This directory location was specified during the SIMULIA Execution Engine installation.
 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.

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

The SIMULIA Execution Engine installation wizard, which is included on your SIMULIA Execution Engine DVD, 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; see [Installing as an Administrator \(Windows\) or Non-Root User \(UNIX/Linux\)](#).

2. Insert the SIMULIA Execution Engine DVD.
3. If you are installing on Windows 7, Windows Vista, or Windows Server 2008, click **Allow** to confirm the execution of the installation program.

Note: If the installer does not start automatically, execute the `setupwin.exe` file located on the top level of the SIMULIA Execution Engine DVD. This installation program is used for both the 32-bit and 64-bit versions of the software. Do not double-click the `setup.jar` file. Double-clicking the `setup.jar` file starts the installer using a local copy of Java, but the installation will later fail.

The installer starts, and the splash screen appears. The installation wizard will guide you through the installation process.

4. Click **Next**.

The installation directory screen appears.

5. Perform one of the following actions:

- To accept the default directory, click **Next**. Your default directory may vary based on your system settings and permissions.
- If you want to install the SIMULIA Execution Engine in a different location, click **Browse**, choose a destination folder, and click **Open**.

Click **Next**.

6. Select the software you want to install:

- **Station**. This option installs everything needed to run a SIMULIA Execution Engine station that will connect to the SIMULIA Execution Engine server.

Click **Next**.

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

- **FLEXnet License Server**
- **Dassault Systemes License Server**
- **Skip license setup 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 name and (optionally) port number. This information is used to contact the license server and create a client license file that references the server. If you have installed a redundant license server triad, enter the host name and port for all three machines. The port number is found in your license file—the default is to leave this setting empty (the license manager software selects the port number).

Click **Next**.

8. Enter the following required information, which allows the installation wizard to build a connection profile (.cpr file) for connecting the station to the SIMULIA Execution Engine server. If no server name is provided, the installation wizard will not create a .cpr file.

- **Server Name.** The host name of the computer running the SIMULIA Execution Engine server.

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.** Select **WebLogic Server 10** from the list.
- **Port Number.** This setting is determined automatically using the default port for the server type. If necessary, you can change this value.

9. Click **Next**.

The installation progress screen appears, and the software is installed. Once the installation is complete, an installation summary appears.

10. After the installer completes, you should 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](#).

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 Manually

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 as described in [Opening a Command Prompt Dialog Box on Windows](#).
3. Type the following command (for a default SIMULIA Execution Engine station installation):

- For Windows 32-bit:

```
C:\SIMULIA\ExecutionEngine\5.7\intel_a\code\command\installstation.bat
```

- For Windows 64-bit:

```
C:\SIMULIA\ExecutionEngine\5.7\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 XP:
 - a) Click the **Start** button, and click **Control Panel**.
 - b) Click the **Performance and Maintenance** link.

-
- c) Click the **Administrative Tools** link, and double-click the **Services** icon.
 - If your station is running on Windows Server 2003:
 - a) Click the **Start** button, and point to **Control Panel/Administrative Tools**.
 - b) Click **Services**.
 - If your station is running on Windows Vista:
 - a) Click the **Start** button, and click **Control Panel**.
 - b) Click **System and Maintenance**.
 - c) Click **Administrative Tools**, and double-click **Services**.
 - d) Click **Continue**.
 - 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 `SIMULIA Execution Engine 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 `SIMULIA Execution Engine 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 as described in [Opening a Command Prompt Dialog Box on Windows](#).
 3. Type the following command (for a default SIMULIA Execution Engine station installation):

- For Windows 32-bit:

```
C:\SIMULIA\ExecutionEngine\5.7\intel_a\code\command\uninstallstation
```

- For Windows 64-bit:

```
C:\SIMULIA\ExecutionEngine\5.7\win_b64\code\command\uninstallstation
```

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 **SIMULIA Execution Engine 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.

If you have two releases of SIMULIA Execution Engine 5.7 installed (for example, the initial release and a patch release that followed), you should directly access the uninstaller of the release you want to remove. It is recommended that you not use the Windows Control Panel as described in the following procedure. The uninstaller is located in the `_uninst` subdirectory of the main SIMULIA Execution Engine directory for the release you want to remove. Execute the `uninstall.exe` file.

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. Click **Start**, and click **Control Panel**.

The **Control Panel** appears.

2. Access the list of installed programs using one of the following options, based on your operating system:
 - **Windows XP:** Click **Add or Remove Programs**.
 - **Windows Server 2003:** Click **Start**, point to **Control Panel**, and click **Add or Remove Programs**.
 - **Windows Vista, Windows Server 2008, and Windows 7:** Under the **Programs** option, click **Uninstall a program**.

A list of installed programs appears.

3. Perform one of the following actions, based on your operating system:
 - **Windows XP, Windows Server 2003:** From the list of programs, select **SIMULIA Execution Engine x.x**, and click **Change/Remove**.
 - **Windows Vista, Windows Server 2008, and Windows 7:** From the list of programs, select **SIMULIA Execution Engine x.x**, click **Uninstall/Change**, and click **Continue** to verify the action.

The **Welcome** dialog box appears.

4. Click **Next**.

A summary of the uninstallation appears.

5. Click **Next**.

Note: If you are asked about individual files during the uninstall process, click the **Yes to All** option to remove all affected files.

The software is removed. You are informed when the removal is complete.

6. Click **Finish**.

The SIMULIA Execution Engine station has been successfully deleted.

7. If necessary, close the list of installed files.

8. Close the **Control Panel**.

9. You now need to remove any temporary directories or files that the SIMULIA Execution Engine station created as described in [Deleting Temporary Directories and Files](#).

Deleting Temporary Directories and Files

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\5.7`, navigate to the `C:\SIMULIA\Execution Engine` directory.
2. Delete the `5.7` directory and all of its contents.

-
3. Navigate to one of the following directories, based on your operating system, where `<user_name>` is the name of the user who installed and uninstalled the SIMULIA Execution Engine station:
 - Windows XP/Server 2003: `C:\Documents and Settings\<username>`
 - Windows Vista/Server 2008/7: `C:\Users\<username>`
 4. Delete the following items:
 - the `fiper.preferences` file
 - the `fiper` directory
 5. Navigate to one of the following directories, based on your operating system, where `<user_name>` is the name of the user who installed and uninstalled the SIMULIA Execution Engine station:
 - Windows XP/Server 2003: `C:\Documents and Settings\<username>\Local Settings\Temp`
 - Windows Vista/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 the SIMULIA Execution Engine Software on UNIX and Linux

This section describes how to install and configure the SIMULIA Execution Engine software on UNIX and Linux platforms. It includes installing the SIMULIA Execution Engine station software with the SIMULIA Execution Engine software. If you want to install only the SIMULIA Execution Engine station software, see [Installing a SIMULIA Execution Engine Station on UNIX and Linux](#). In particular, if you are using non-default settings for the Start menu, Control Panel, and folder and file extension options, you may have difficulty following the installation procedures.

Installing the SIMULIA Execution Engine Software

You do not need to uninstall SIMULIA Execution Engine 5.6 to install SIMULIA Execution Engine 5.7, but you must stop the SIMULIA Execution Engine and any SIMULIA Execution Engine stations before installing the upgrade.

Root (superuser) privileges are not required to install the SIMULIA Execution Engine on UNIX or 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 UNIX requires an X-Windows display, either local or remote over a network. If the `DISPLAY` environment variable is not set, the installer will not execute.

1. Log in as the user that will install the SIMULIA Execution Engine.

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 \(UNIX/Linux\)](#) 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. Load the SIMULIA Execution Engine DVD and mount it, if necessary.

On some UNIX/Linux systems, the SIMULIA Execution Engine DVD will mount automatically when it is inserted into the DVD drive, and a file browser window will appear. In this case you can double-click the appropriate `setupXXXX` script file for your operating system (see Step 4).

3. From the terminal window (shell), change to the directory where the SIMULIA Execution Engine DVD is mounted. For example, type:

```
cd /media/dvd
```

4. Execute the installation script by typing one of the following commands, based on your operating system:

- `./AIX/setupaix`
- `./Linux/setuplinux`
- `./SUN/setup_solaris`



Note: The `setupaix` file can be used only when configuring a WebSphere-based SIMULIA Execution Engine or when installing a SIMULIA Execution Engine station, but not for a WebLogic application server.

The SIMULIA Execution Engine installation wizard starts, showing the **Welcome** screen. This wizard will guide you through the rest of the installation.

5. Click **Next**.

The legal notice agreement screen appears. Read the notice, and click **Next**.

6. In the next screen, perform one of the following actions:

- To accept the default installation directory, click **Next**. Your default directory may vary based on your system settings and permissions.
- If you want to place the SIMULIA Execution Engine in a different location, click **Browse**, choose a destination folder, and click **Open**.

7. Select **SIMULIA Execution Engine** as the product to install. Click **Next**.

8. Select the type of license server software you will use with the SIMULIA Execution Engine, or skip the licensing selection for now:

- **FLEXnet License Server**
- **Dassault Systemes License Server**
- **Skip license setup 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 name and (optionally) port number. This information is used to contact the license server and create a `license.dat` file that references the server. If you have installed a redundant license server triad, enter the host name and port for all three machines. The port number is found in your license file—the default is to leave this setting empty (the license manager software selects the port number).



Note: The FLEXnet license server is not available and cannot be installed or run on AIX or Solaris.

For details about the choice of licensing scheme, see [About Licensing and Your License File](#). For complete information about configuring the license server, refer to the appropriate documentation:

- *FLEXnet License Administration Guide* Version 11.6.1, which is available for download from the **Licensing** section at <http://www.3ds.com/products/simulia/support/>.
- *DS License Server Installation and Configuration Guide* (`DSL_S .pdf`), which is available in the `Doc/` subdirectory of your SIMULIA Execution Engine DVD.

Click **Next**.

9. Enter the following required information, which allows the installation wizard to build a connection profile (`.cpr` file) for interfaces that will connect to the SIMULIA Execution Engine. If no server name is provided, the installation wizard will not create a `.cpr` file.

- **Server Name**. The host name of the computer running the SIMULIA Execution Engine. The name of the current host is used if you are installing the SIMULIA Execution Engine.

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**. Select **WebLogic Server 10** from the list.
- **Port Number**. This setting is determined automatically using the default port for the server type. If necessary, you can change this value.

Click **Next**.

10. Click Next.

The installation progress screen appears, and the software is installed. Once the installation is complete, an installation summary appears.

11. Review the summary, and click Next.

12. Click Next.

Information about the .ear files for the SIMULIA Execution Engine appears.

13. Review the information, and click Finish.

The installation is complete.

Installing the SIMULIA Execution Engine Non-Interactively

You can install the SIMULIA Execution Engine with minimal interaction, instead of specifying the numerous options available using the installation script. When installing with pre-determined user responses, you run the installation program once (in the usual way) to create a response file that stores your installation settings. You can then use this file to automate the installation, without user interaction, when the install program is run again on this computer or some other computer. The install options file created on UNIX will not work on Windows.

1. Open a terminal window.
2. Enter the following command (where `<dvdrom_mount_point>` is specific to your system, and `platform` specifies the type of UNIX operating system you are using):

```
<dvdrom_mount_point>/setupplatform -options-record  
<absolute_path_to_response_file>/<response_file>
```

Note: This command should be entered on a single line at the terminal prompt. You must use an absolute path to where you want the response file written. This command will not work with a relative path.

3. In the installation interface that appears, follow the steps for the installation procedure.

The response file is generated automatically.

4. Enter the following command (where `<dvdrom_mount_point>` is specific to your system, and `platform` specifies the type of UNIX operating system you are using) to run the installation without interaction:

```
<dvdrom_mount_point>/setupplatform -silent -options  
<absolute_path_to_response_file>/<response_file>
```

Note: This command should be entered on a single line at the terminal prompt.

An absolute path *must* be used when recording or using a response file because the setup program changes the working directory, which invalidates a relative path.

If you will be installing the software on many systems, it is usually more convenient to copy the entire installation DVD to a shared file system and then run the install from there. This setup avoids having to mount the DVD on each installation system.

Uninstalling the SIMULIA Execution Engine and License Server

This section describes how to uninstall the SIMULIA Execution Engine on UNIX and Linux.

Uninstalling the SIMULIA Execution Engine and License Server

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 (see [Stopping the SIMULIA Execution Engine](#)).
2. If you are running a license server on the same computer as the SIMULIA Execution Engine, stop the license server process.
3. Removing the SIMULIA Execution Engine software (see [Removing the SIMULIA Execution Engine Software](#)).
4. Removing the temporary directories and files created by the SIMULIA Execution Engine (see [Deleting Temporary Directories and Files](#)).

Stopping the SIMULIA Execution Engine

You must stop the SIMULIA Execution Engine and the Isight gateways.

1. Verify that you are logged in as the same user that installed the SIMULIA Execution Engine.
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 can now safely remove the SIMULIA Execution Engine software.

1. Navigate to the following directory:
`<SEE_install_directory>/_uninst`

-
2. Execute the command `./uninstaloperating_system`

The Welcome dialog box appears.

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

3. Click **Next**.

A summary of the uninstallation appears.

4. Click **Next**.

Note: If you are asked about individual files during the uninstall process, click the **Yes to All** option to remove all affected files.

The software is removed. You are informed when the removal is complete.

5. Click **Next**.

A message appears, informing you that you must log out and log back into the system to reset the environment variables.

6. Click **Finish**.

The uninstaller is closed.

7. Log out of your system, and log back into your system.

8. You now need to remove any temporary directories or files that the SIMULIA Execution Engine created as described in [Deleting Temporary Directories and Files](#).

Deleting Temporary Directories and Files

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/5.7`, navigate to the `/opt/SIMULIA/Execution Engine` directory.
3. Delete the 5.7 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 UNIX and Linux

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

Before You Begin

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

Supported Platforms and Patch Information

For complete details on the supported platforms for SIMULIA Execution Engine 5.7, refer to the following Web site:

<http://www.3ds.com/products/simulia/support/>

Necessary Settings for AIX Systems

The following sections describe AIX-specific settings and changes that are necessary to ensure that the SIMULIA Execution Engine stations function properly.

AIX Operating System Requirements

To ensure that execution on the AIX operating system is successful, you need to verify that the following patch level has been installed, based on your version of AIX:

- AIX v6.1: minimum AIX level 6100-GOLD

FORTRAN Library Requirements

If you are installing your SIMULIA Execution Engine station on an AIX-based system, you need to verify that you have the FORTRAN run-time file (`libxlf90.a`) installed in the `/usr/lib/xlf` directory. This file is an optional add-on to the AIX operating system and may need to be installed manually.

Installations of the SIMULIA Execution Engine on the AIX platform require the following libraries:

- `xlf90` 8.1.0.0 XL Fortran Runtime Environment
- `xlf90.aix50` 8.1.0.0 XL Fortran Runtime Environment AIX 5.0 Libraries

For more information on determining if you have these libraries installed, or instructions on how to install the libraries, refer to your local system administrator.

Solaris System Settings

The following sections describe Solaris-specific settings and changes that are necessary to ensure that SIMULIA Execution Engine stations function properly.

Java Run-time Environment Requirements

You must install Solaris patches as described on the following web site:

<http://www.oracle.com/technetwork/systems/patches/solaris/index-jsp-139697.html>

Swap Space Settings

If you are installing the SIMULIA Execution Engine on a Solaris-based system, you must verify that the system has swap space equal to three or four times the main memory. For more information on determining your system's current specification, contact your local system administrator.

The large amount of swap space is needed because the SIMULIA Execution Engine station can grow to more than 500 MB. Additional swap space equal to the process size is allocated for a short time every time the program creates a sub-process to execute an external program. This space is never used but must be available. Insufficient swap space will cause the SIMULIA Execution Engine jobs to fail sporadically with messages about “insufficient disk space.” Other versions of UNIX/Linux use a different scheme for allocating swap space to processes and can run the SIMULIA Execution Engine with swap space equal to twice main memory.

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 `plaunch` 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 `plaunch` 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 UNIX/Linux](#).

Installing the Station Software

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

1. Log in as the user that will install the SIMULIA Execution Engine.

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 \(UNIX/Linux\)](#) 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. Load the SIMULIA Execution Engine DVD and mount it, if necessary.

On some UNIX/Linux systems, the SIMULIA Execution Engine DVD will mount automatically when it is inserted into the DVD drive, and a file browser window will appear. In this case, you can double-click the appropriate `setupXXXX` script file for your operating system (see Step 4).

3. From the terminal window (shell), change to the directory where the SIMULIA Execution Engine DVD is mounted. For example, type:

```
cd /media/dvd
```

4. Execute the installation script by typing one of the following commands, based on your operating system:

- `./AIX/setupaix`
- `./Linux/setuplinux`
- `./SUN/setupSolaris`

The SIMULIA Execution Engine installation wizard starts, showing the **Welcome** screen. This wizard will guide you through the rest of the installation.

5. Click **Next**.

The legal notice agreement screen appears. Read the notice, and click **Next**.

6. In the next screen, perform one of the following actions:

- To accept the default installation directory, click **Next**. Your default directory may vary based on your system settings and permissions.
- If you want to place the SIMULIA Execution Engine station in a different location, click **Browse**, choose a destination folder, and click **Open**.

7. Select **Station** as the product to install. Click **Next**.

8. Select the type of license server software you will use with the SIMULIA Execution Engine, or skip the licensing selection for now:

- **FLEXnet License Server**
- **Dassault Systemes License Server**
- **Skip license setup 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 name and (optionally) port number. This information is used to contact the license server and create a `license.dat` file that references the server. If you have installed a redundant license server triad, enter the host name and port for all three machines. The port number is found in your license file—the default is to leave this setting empty (the license manager software selects the port number).



Note: The FLEXnet license server is not available and cannot be installed or run on AIX or Solaris.

For details about the choice of licensing scheme, see [About Licensing and Your License File](#). For complete information about configuring the license server, refer to the appropriate documentation:

- *FLEXnet License Administration Guide* Version 11.6.1, which is available for download from the **Licensing** section at <http://www.3ds.com/products/simulia/support/>.

-
- *DS License Server Installation and Configuration Guide* (DLSLS .pdf), which is available in the Doc/ subdirectory of your SIMULIA Execution Engine DVD.

Click **Next**.

9. Enter the following required information, which allows the installation wizard to build a connection profile (.cpr file) for the station to connect to the SIMULIA Execution Engine. If no server name is provided, the installation wizard will not create a .cpr file.

- **Server Name.** The host name of the computer running the SIMULIA Execution Engine. The name of the current host is used if you are installing the SIMULIA Execution Engine.

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.** Select **WebLogic Server 10** from the list.
- **Port Number.** This setting is determined automatically using the default port for the server type. If necessary, you can change this value.

Click **Next**.

10. Click **Next**.

The installation progress screen appears, and the software is installed. Once the installation is complete, an installation summary appears.

11. Review the information, and click **Finish**.

The installation is complete.

12. After the installer completes, you should 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](#).

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 system's temporary directory (usually `/var/tmp`). 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 station 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.
 - Preferably on a local disk on the station host system. If there is insufficient local storage space, a NAS device can be used, but this setup is not recommended.

The actual station temporary directory is created as a sub-directory of the specified directory. The sub-directory name is the name of the computer running the station. It is important that this directory be writable by the user running the station. If the station is started as a service, the permissions on this directory will be incorrect for running the station manually. This directory should be given full-access to all users (permissions `rwxxrwxxrwxx`) under most circumstances.

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 or UNIX, 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
solaris_a64    for Solaris
aix_a64        for AIX
```

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 UNIX/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 UNIX or 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/5.7/linux_a64/code/command/station.service
install
```

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

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

- To uninstall the station, run the following command (as a root user):

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

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

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

- You can stop a station as a service with the following command:

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

- If you want the station to run as a non-root user, edit the `station.service.template` file before installing the station as a service and change the following line:

```
STATION_USER=
```

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

Uninstalling a SIMULIA Execution Engine Station

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

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

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

Stopping the SIMULIA Execution Engine Station

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

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

Important: 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 safely remove the SIMULIA Execution Engine station software.

1. Navigate to the following directory:

```
<SEE_install_dir>
/_uninst
```

2. Execute the command `./uninstall<operating_system> .`

The Welcome dialog box appears.

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. Click **Next**.

A summary of the uninstallation appears.

4. Click **Next**.

Note: If you are asked about individual files during the uninstall process, click the **Yes to All** option to remove all affected files.

The software is removed. You are informed when the removal is complete.

5. Click **Next**.

A message appears, informing you that you must log out and log back into the system to reset the environment variables.

6. Click **Finish**.

The uninstaller is closed.

7. Log out of your system, and log back into your system.

8. 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/5.7`, navigate to the `/opt/SIMULIA/Execution Engine` directory.
3. Delete the `5.7` directory and all of its contents.
4. Navigate to the `$HOME` directory for the user that installed the SIMULIA Execution Engine.
5. Delete the following items (if they are present):
 - `.fiper.preferences` file (notice the leading “.” in the file name)
 - `dashboard.log` file
 - `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 database for use with the SIMULIA Execution Engine.

Initializing an Oracle Database

Before the SIMULIA Execution Engine is configured, an Oracle database's 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` to control the configuration of the SIMULIA Execution Engine.

These instructions apply to both Oracle 10g and 11g, with some differences noted.

About Oracle URLs and Port Numbers

If you are accessing the Oracle web-based interface on UNIX/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 11g, 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, open the `emConfig.log` file, and skip to the last line of the file. The full URL for the database, including the port number, is shown.

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 - OraDb10g_home1** or **OraDb11g_home1**, and click **Database Control - FIPER**. (In this example, **FIPER** is the name of the database you are accessing.)

-
- UNIX/Linux: Use a browser to open the following page:

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

In Oracle 11g, use `https` instead of `http` in this URL. For more information on determining the port number for your database, see [About Oracle URLs and Port Numbers](#).

4. (Oracle 10g only) Verify that you are about to access the correct database.
The screen that appears should have the correct database name next to the **Login to Database** string.
5. Type `SYS` as the user name in the corresponding text box, and enter the password you specified when you created the database.
6. From the **Connect As** list, select **SYSDBA**.
7. Click **Login**.
8. (Oracle 11g only) Verify that you have accessed the correct database.
The screen that appears after you log in shows the database name next to the **Instance Name** string (in the upper left corner of the screen).
9. At the top of the console, access the appropriate **Administration** or **Server** information.
 - **Oracle 10g:** Click **Administration**.
 - **Oracle 11g:** Click **Server**.
10. In the **Storage** list, click **Tablespaces**.
The **Tablespaces** page opens.
11. On the right side of the console, click **Create**.
The **Create Tablespace** page appears.
12. In the **Name** text box, type the following:
`F1PERTS1`
13. Click **Set as default permanent tablespace**.
14. In the **Datafiles** area, click **Add**.
The **Add Datafile** page appears.
15. In the **File Name** text box, type the following entry:
`F1PERDF1.DBF`
16. In the **File Size** text box, type `2048`, and verify that **MB** is selected from the corresponding list.
17. Click **Automatically extend datafile when full (AUTOEXTEND)**.
18. In the **Increment** text box, type `100`, and select **MB** from the corresponding list.
19. In the **Maximum File Size** area, verify that **Unlimited** is selected.
20. Click **Continue**.
The **Create Tablespace** page appears.
21. Click **OK**.
A message appears indicating that the object was created successfully.
22. At the top of the left side of the console, click **Database Instance**.
The **Administration** page (Oracle 10g) or **Server** page (Oracle 11g) appears.

23. Access the **Users** page:

- **Oracle 10g:** In the **Users & Privileges** section, click **Users**.
- **Oracle 11g:** In the **Security** list, click **Users**.

The **Users** page appears.

24. Click **Create**.

The **Create User** page appears.

25. In the **Name** text box, type `fiperacs`.

26. From the **Profile** list, verify that **DEFAULT** is selected.

27. In both the **Enter Password** and **Confirm Password** text boxes, type the password for the `seeadmin` user (usually `seeadmin`).

28. Click the  icon adjacent to the **Default Tablespace** text box.

The **Search and Select** window appears.

29. Verify that **FIPERTS1** is selected.

30. Click **Select**.

You are returned to the **Create User** page.

31. Click **Roles**.

32. On the right side of the console, click **Edit List**.

The **Modify Roles** screen appears.

33. From the **Available Roles** list, select **Resource**.

34. Click **Move** to move the role to the **Selected Roles** list.

35. Click **OK**.

You are returned to the **Create User** page.

36. Click **Object Privileges**.

37. From the **Select Object Type** list, select **View**, and click **Add**.

38. Click the  icon adjacent to the **Select View Objects** text box.

The **Select View Objects** window appears.

39. From the **Schema** list, select **SYS**.

40. In the **Search View Name** text box, type the following entry:

```
DBA_PENDING_TRANSACTIONS
```

41. Click **Go**.

The object is now listed at the bottom of the window.

42. Click **DBA_PENDING_TRANSACTIONS**, and click **Select**.

The **Add View Objects Privileges** page appears.

43. In the **Available Privileges** list, click **Select**.

44. Click **Move** to move the item to the **Selected Privileges** list.

45. Click **OK**.

You are returned to the **Create User** page.

46. Click **OK**.

A message appears indicating the object was created successfully.

47. In the top right corner of the console, click **Logout** to exit the Enterprise Manager.

48. 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 UNIX/Linux).
2. If you are updating a database that is running on a UNIX/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. (Oracle 10g) Execute the following commands by typing each command individually and pressing the **Enter** key:

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

Note: The `fiperacs` user name in these commands 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. (Oracle 11g) 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 fiperacs;
```

Note: The `fiperacs` 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.

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

The database update is complete.

7. 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 UNIX/Linux).
2. Navigate to the following directory:

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

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

- `/intel_a/` for Windows 32-bit
 - `/win_b64/` for Windows 64-bit
 - `/linux_a64/` for Linux 64-bit
 - `/solaris_a64/` for Oracle Sun Solaris 64-bit
 - `/aix_a64/` for AIX 64-bit
3. Type one of the following commands, based on your operating system (where `user_name` is the name of the user 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`
 - UNIX/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 UNIX/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.

Configuration

This section describes additional steps that may be necessary to configure the or a station. It also describes changes that may be needed to your license file based on your network environment.

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 XP/Server 2003: Click the **Start** button, and click **Run**.
 - Windows Vista/Server 2008/Windows 7: Click, the **Start** button.
2. Perform one of the following steps, based on your operating system:
 - Windows XP/Server 2003: In the **Open** text box, type `dcomcnfg`, and click **OK**.
 - Windows Vista/Server 2008/Windows 7: 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 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 , 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 .

1. Verify that the 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 **SIMULIA Execution Engine 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.

Configuring WebLogic

This section describes how to set up and start the SIMULIA Execution Engine using the WebLogic Application Server and an Oracle database.

About Configuring WebLogic

This section describes the configuration of WebLogic 11g (10.3.2.0) with Oracle 10g (10.2.0.1.0) or Oracle 11g (11.1.0.6.0) to run the SIMULIA Execution Engine

The configuration procedures in this section assume that you created a user called `seeadmin` to control the configuration of the SIMULIA Execution Engine and that you installed the SIMULIA Execution Engine in the default installation directory. For more information, see [Installing the SIMULIA Execution Engine Software on Windows](#) or [Installing the SIMULIA Execution Engine Software on UNIX and Linux](#).

About the seeadmin User

To configure the SIMULIA Execution Engine, you need to have a user called `seeadmin`, and this user must possess certain characteristics within your environment.

This user must be known within the target security realm of the user authentication registry. These security settings must be implemented or the SIMULIA Execution Engine deployment will fail. The remaining procedures in this section assume that you created the `seeadmin` user and have preconfigured the necessary security settings. If you create a user with a different name or use an existing user with a different name, with the proper security settings, keep this in mind when user information is presented.

For information on creating this user (on Windows systems) and for more details on the necessary preconfigured security settings, see [Creating and Configuring a Local User on Windows](#).

For more information on creating this user in a UNIX/Linux environment, contact your system administrator.

Deploying the SIMULIA Execution Engine in a Single-Server Standalone Configuration

You can deploy the SIMULIA Execution Engine application in a single-server, standalone configuration in which only the WebLogic Administration Server is used. For this option, you perform the following steps:

1. Create the WebLogic domain using the administration server template provided in your SIMULIA Execution Engine installation. See [Creating the WebLogic Domain Using the SIMULIA Execution Engine Administration Server Template](#).
2. Edit the WebLogic environment file, as described in [Editing the Environment File](#).
3. Start the WebLogic Administration Server and access the WebLogic Administration Console through a web browser. See [Starting the Administration Server](#).
4. Configure client authentication for the SIMULIA Execution Engine within WebLogic, using an LDAP server. See [Configuring Client Authentication](#).
5. Deploy the Java EAR file, as described in [Deploying the SIMULIA Execution Engine EAR File](#).
6. Deploy any additional SIMULIA Execution Engine applications desired, such as the WebTop, WebDashboard, or Federation application. See [Deploying the WebTop, WebDashboard, or Federation Applications](#).
7. Configure the WebTop, WebDashboard, or Federation applications, as needed. See [Configuring the WebTop and WebDashboard Applications](#).
8. Stop and restart WebLogic as described in [Stopping the SIMULIA Execution Engine Server](#), to force WebLogic and the SIMULIA Execution Engine to recognize the configuration changes.

-
9. Configure the connection profile for the SIMULIA Execution Engine, as described in [Configuring the Connection Profile](#).
 10. Preload the basic system metamodels for Isight and the SIMULIA Execution Engine, as described in [Publishing to the Library](#).

Deploying the SIMULIA Execution Engine Using Multiple Servers

Alternatively, you can deploy the SIMULIA Execution Engine application and its web client UIs in multiple WebLogic servers, using WebLogic Managed Servers for each application. For this option, you perform the following steps:

1. Create the WebLogic domain using the administration server template provided in your SIMULIA Execution Engine installation. See [Creating the WebLogic Domain Using the SIMULIA Execution Engine Administration Server Template](#).
2. Edit the WebLogic environment file, as described in [Editing the Environment File](#).
3. Start the WebLogic administration server and access the WebLogic Administration Console through a web browser. See [Starting the Administration Server](#).
4. Configure client authentication for the SIMULIA Execution Engine within WebLogic, using an LDAP server. See [Configuring Client Authentication](#).
5. Define a managed server for SIMULIA Execution Engine within the WebLogic Administration Server. See [Creating a Managed Server Using the ServerTemplate](#).
6. For a multiple-server domain, you must create a WebLogic domain on the computers that will host the managed servers. See [Creating WebLogic Domains for Remote Managed Servers](#).
7. Start the managed server for the SIMULIA Execution Engine, as described in [Starting the Managed Servers](#).
8. Deploy the Java EAR file, as described in [Deploying the SIMULIA Execution Engine EAR File](#).
9. Deploy any additional SIMULIA Execution Engine applications desired, such as the WebTop, WebDashboard, or Federation application. See [Deploying the WebTop, WebDashboard, or Federation Applications](#).
10. Configure the WebTop, WebDashboard, or Federation applications, as needed. See [Configuring the WebTop and WebDashboard Applications](#).
11. Stop and restart WebLogic as described in [Stopping the SIMULIA Execution Engine Server](#), to force WebLogic and the SIMULIA Execution Engine to recognize the configuration changes.
12. Configure the connection profile for the SIMULIA Execution Engine, as described in [Configuring the Connection Profile](#).
13. Preload the basic system metamodels for Isight and the SIMULIA Execution Engine, as described in [Publishing to the Library](#).

Additional security options are available with your SIMULIA Execution Engine, including Run-As security and Access Control Lists (ACLs). For more information on these options, see [Configuring Security](#).

About the Deployment Configuration and Options

Before you begin the configuration process for your WebLogic SIMULIA Execution Engine, you need to understand the options available for your configuration, including your server arrangement and the templates included with your SIMULIA Execution Engine installation. Your exact configuration will be based on your environment.

About Domain Composition

A domain is the basic unit of administration for the WebLogic Server and its runtime environment. It consists of one or more WebLogic Server instances, computers, components, and logically related resources and services that are managed, collectively, as one unit, which consists of one Administration Server and optional Managed Servers.

The SIMULIA Execution Engine domain is a WebLogic domain that must, at least, include a server instance that is configured as the Administration Server with the following server domain configuration types:

- Single-Server** This is a standalone server configuration consisting of a single Administration Server that also hosts the SIMULIA Execution Engine enterprise and web applications.
- Multiple-Server** This type of domain configuration consists of an Administration Server, performing management operations, with Managed Servers hosting the SIMULIA Execution Engine enterprise and web applications. Managed Servers can be targeted for deployment as one of the following.
- **Vertical Targeting.** This is a single high-end server machine that allows easy addition of system resources. The Administration Server and Managed Server instances run on this machine.
 - **Horizontal Targeting.** The Administration Server and each Managed Server instance run on a separate server machine, which may offer better load balancing.

You can also deploy your managed servers using a combination of these two options, depending on the deployment requirements.

About Configuration Templates

To simplify the process of configuring your WebLogic server, the SIMULIA Execution Engine installation provides two configuration templates.

Domain Templates are used to create Administration and Managed Server domains, and the Server Template is embedded within the Administration Server domain template and is used to create Managed Server instances.

Domain Templates

The WebLogic Configuration Wizard is used to create and configure a new SIMULIA Execution Engine domain for stand-alone servers. It is also used to create the Administration Servers when using Managed Servers. The Configuration Wizard runs independently of the WebLogic Server, and it is recommended that you not use it when a WebLogic Server instance is running. Using configuration templates, the Configuration Wizard guides you through the process of creating the SIMULIA Execution Engine domain for the target environment. The term configuration template refers to a `.jar` (Java Archive) file that contains the required files and scripts to create a new SIMULIA Execution Engine domain.

The SIMULIA Execution Engine installation is delivered with two predefined domain templates:

- AdminServer Template** This template defines the full set of required resources, components, services, and security options. It should be used to create and configure a single-server or multiple-server WebLogic Server domain consisting of an Administration Server performing management operations.
- ManagedServer Template** This template defines a subset of required resources for a Managed Server on a remote machine. It is used to create and configure the WebLogic Server domain for Managed Servers hosting the SIMULIA Execution Engine enterprise and web applications.

Server Template

The Administration Console of the Administration Server is used to create and configure a Managed Server. The Administration Server that is created using the AdminServer template contains a server named ServerTemplate, which is preconfigured with the required server properties for a server instance running as a Managed Server in the SIMULIA Execution Engine domain. This server template is cloned whenever a new Managed Server needs to be created and should never be deleted.

Deployment Considerations

Four deployment units (SIMULIA Execution Engine, Federation (B2B), WebTop, and WebDashboard) are included with the SIMULIA Execution Engine installation. These units can be deployed to a single-server (standalone with a single Administration Server) or to multiple servers.

Important: The SIMULIA Execution Engine and the Federation (B2B) enterprise applications must be deployed to the same server instance, either on the Administration Server or on a Managed Server.

For a multiple-server domain deployment, it is highly recommended that the Administration Server be used only for administration purposes.

The following figure illustrates the various ways in which the SIMULIA Execution Engine applications can be organized and deployed within WebLogic.

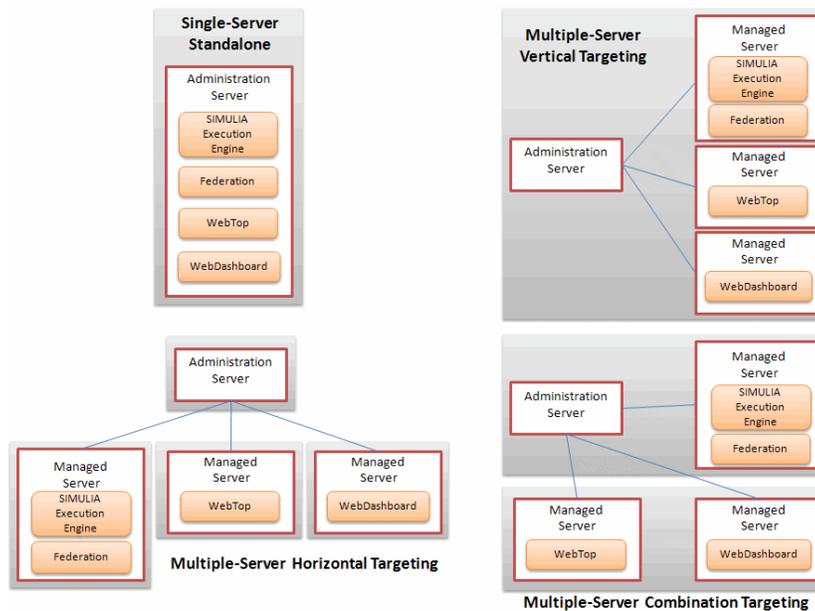


Figure 2: WebLogic Deployment Options

Guidelines for Configuring the WebLogic Server

Before configuring the WebLogic Server, make sure you have decided on the deployment targets for your production environment including the following details:

- You have identified the target domain configuration type: Single-Server or Multiple-Server.
- For a multiple-server domain, you have identified the deployment targeting strategy for each Managed Server: Horizontal or Vertical.

When you create a domain, you also create an Administration Server, which runs the Administration Console and may distribute configuration changes to other servers in the domain.

When configuring the WebLogic Server for SIMULIA Execution Engine applications, you must perform the following tasks:

1. Create a WebLogic domain in which the SIMULIA Execution Engine applications are executed.

For a single-server or a multiple-server domain with Managed Servers, create an Administration Server domain using the procedure described in [Creating the WebLogic Domain Using the SIMULIA Execution Engine Administration Server Template](#).

For a multiple-server domain, create a Managed Server domain for each host with the horizontal targeting strategy using the procedure described in [Creating WebLogic Domains for Remote Managed Servers](#).

2. If you are using a multiple-server domain, create and configure Managed Servers for the domain.

In a typical production environment, you create one or more Managed Servers in the domain to host Enterprise and Web applications and use the Administration Server only to configure and monitor the Managed Servers. A pre-configured server template with the name `ServerTemplate` is included with each SIMULIA Execution Engine Administration Server. This template must be used to create and configure all SIMULIA Execution Engine Managed Servers. Repeat the procedure described in [Creating a Managed Server Using the ServerTemplate](#) for each Managed Server in your domain.

Important: Do not delete or deploy any application to the **ServerTemplate** server instance.

3. Configure the SIMULIA Execution Engine domain per its target runtime environment. Prior to starting any of the configured servers, you must set the required SIMULIA Execution Engine environment variables for its target runtime. Follow the procedure described in [Editing the Environment File](#).

Configuring the SIMULIA Execution Engine

The SIMULIA Execution Engine configuration process includes creating a WebLogic domain using the templates included with your installation, editing your configuration for your local security settings, and deploying the EAR file to WebLogic.

Depending on your configuration, you may also need to create managed servers and configure the SIMULIA Execution Engine Web-based applications: the WebTop and WebDashboard.

Creating the WebLogic Domain Using the SIMULIA Execution Engine Administration Server Template

You must create a WebLogic domain for your SIMULIA Execution Engine environment if you are creating a new SIMULIA Execution Engine or if you are migrating from SIMULIA Execution Engine 5.6 to 5.7.

1. Verify that you have WebLogic installed on the system that will host the Administration Server.
2. Verify that you have initialized your database as described in [Initializing the SIMULIA Execution Engine Database](#).
3. Perform one of the following actions, based on your operating system:
 - **Windows:** Click **Start**, point to **All Programs / Oracle WebLogic / WebLogic Server 11gR1 / Tools**, and click **Configuration Wizard**.
 - **UNIX/Linux:** Execute the `./config.sh` command in the directory `<weblogic_install>/wlserver_10.3/common/bin/`.

The **Fusion Middleware Configuration Wizard** dialog box appears.

4. Verify that **Create a new WebLogic domain** is selected.
5. Click **Next**.

The **Select Domain Source** screen appears.

6. Click **Base this domain on an existing template**.

-
- Click **Browse**, and navigate to the following directory:

```
<SEE_install_dir>/<os_dir>/reffiles/SMAFIPserver/weblogic/deploy/
```

- Select the **AdminServer.jar** file, and click **OK**.

You are returned to the **Configuration Wizard**.

- Click **Next**.

- (optional) Change the domain name and location in the corresponding text box, if desired. It is recommended that you not change these default settings, if possible.

- Click **Next**.

The **Configure Administrator User Name and Password** screen appears.

- If desired, change the username or password.

You are provided with default username and password settings (*seeadmin* and *seeadmin4*, respectively). This information is used to start the server (in Production Mode) and log in to the WebLogic Administration Console (the browser-based interface).

- Click **Next**.

The **Configure Server Start Mode and JDK** screen appears.

- Click **Production Mode**.

- In the **Available JDKs** area, verify that **Sun SDK** is selected. If it is not selected, click it to select it.



Note: If you are creating your domain on a 64-bit system, no JDK options are provided. You must install the appropriate JDK manually. For more information, contact your local system administrator.

- Click **Next**.

The **Configure JDBC Data Sources** screen appears.

- Click the check box adjacent to the **Fiper XA Data Source** option.

- In the text boxes at the top of the wizard, change the following settings for your database:

- DBMS/Service:** The name of the database (for example, *fiper*)
- Host Name:** The name of the computer where the database is running.
- Port:** If necessary, alter this number to match the port for your database.

- Clear (uncheck) **Fiper XA Data Source**.

- If you have a nonXA data source, click the check box adjacent to the **Fiper nonXA Data Source** option and change the following settings for your database:

- DBMS/Service:** The name of the nonXA database (for example, *fiper*)
- Host Name:** The name of the computer where the nonXA database is running.
- Port:** If necessary, alter this number to match the port for your nonXA database.

- Clear (uncheck) **Fiper nonXA Data Source**.

- Click **Next**.

The data source connections are tested, and the status is displayed. Verify that both connections were successful.



Important: Do not proceed if either of the test connections fail. Both tests must be successful for the domain installation to complete properly. For more information about connecting your database, contact your local database administrator or system administrator.

23. Click **Next**.

The **Select Optional Configuration** screen appears.

24. Click **Next**.

The **Configuration Summary** appears.

25. Click **Create**.

The status of the operation is displayed. Verify that the domain was successfully created.

26. Click **Done** when the operation is complete.

Editing the Environment File

The WebLogic environment file must be edited prior to executing the SIMULIA Execution Engine. This process varies based on the operating system on which WebLogic is running.

Editing the Environment File on Windows

You need to edit the WebLogic `setFiperEnv.cmd` file to match your environment.

1. Navigate to the following directory:

```
<weblogic_install_dir>\user_projects\domains\SEE\
```

Note: This directory location assumes that your domain name is `SEE`, which is the default domain name specified by the WebLogic template. If you changed the domain name (under [Creating the WebLogic Domain Using the SIMULIA Execution Engine Administration Server Template](#)), navigate to the appropriate domain directory.

2. Open the `setFiperEnv.cmd` file with the text editor of your choice.

3. Locate the following entry in the file:

```
set FIPER_HOME=
```

4. Verify that the correct SIMULIA Execution Engine installation path is defined.

If necessary, alter the path to point to your installation location. Be sure to use forward slashes (`/`) in your path.

5. Locate the section labeled:

```
Set Fiper/WebLogic Server platform-specific environment variables
```

This section contains information for all supported SIMULIA Execution Engine platforms that correspond to the installed WebLogic Server JVM (32-bit or 64-bit). All of the information is commented out.

6. Uncomment the lines for your installed WebLogic Server by removing the `@REM` characters at the front of the line.

Each installed WebLogic Server has two lines that must be uncommented. For example, if you are running a 64-bit WebLogic Server on a 64-bit Windows platform, the uncommented lines appear as shown below:

```
@REM 64-bit Windows WebLogic Server
set PLATFORM=win64
set USER_MEM_ARGS=-XX:MaxPermSize=256m -Xms512m -Xmx2048m
```

If you are running a 32-bit WebLogic Server on a 64-bit Windows platform, the uncommented lines appear as shown below:

```
@REM 32-bit Windows WebLogic Server
set PLATFORM=win32
set USER_MEM_ARGS=-XX:MaxPermSize=128m -Xms512m -Xmx1024m
```

7. Save and close the file.

Editing the Environment File on Solaris and Linux

You need to edit the WebLogic `setFiperEnv.sh` file to match your environment.

1. Navigate to the following location:

```
<weblogic_install_dir>/user_projects/domains/SEE/
```

Note: This directory location assumes that your domain name is `SEE`, which is the default domain name specified by the WebLogic template. If you changed the domain name (under [Creating the WebLogic Domain Using the SIMULIA Execution Engine Administration Server Template](#)), navigate to the appropriate domain directory.

2. Open the `setFiperEnv.sh` file with the text editor of your choice.
3. Locate the following entry in the file:

```
FIPER_HOME=
```

4. Verify that the correct SIMULIA Execution Engine installation path is defined. If necessary, alter the path to point to your installation location.
5. Locate the section labeled:

```
Set Fiper/WebLogic Server platform-specific environment
variables
```

This section contains information for all supported SIMULIA Execution Engine platforms that correspond to the installed WebLogic Server JVM (32-bit or 64-bit). All of the information is commented out.

Note: Only the 32-bit Solaris platform is supported.

6. Uncomment the lines for your installed WebLogic Server (Solaris or Linux) by removing the `#` character at the front of the line.

Each installed WebLogic Server has two lines that must be uncommented. For example, if you are running a 64-bit WebLogic Server on a 64-bit Linux platform, the uncommented lines appear as shown below:

```
# 64-bit Linux WebLogic Server
PLATFORM="Linux_x64"
USER_MEM_ARGS="-XX:MaxPermSize=256m -Xms512m -Xmx2048m"
```

If you are running a 32-bit WebLogic Server on a 64-bit Linux platform, the uncommented lines appear as shown below:

```
# 32-bit Linux WebLogic Server
PLATFORM="Linux_7.3_x86"
USER_MEM_ARGS="-XX:MaxPermSize=128m -Xms512m -Xmx1024m"
```

7. Save and close the file.

Starting the Administration Server

After creating the WebLogic domain and editing the WebLogic environment file, you need to start the WebLogic server and access the WebLogic Administration Console through a web browser.

1. Perform one of the following actions, based on your operating system:
 - **Windows:** Open a Command Prompt window, and execute the `startWebLogic.cmd` file in the `<weblogic_install_dir>\user_projects\domains\SEE\` directory.
 - **UNIX/Linux:** Open a terminal window, and execute the `./startWebLogic.sh` command in the `<weblogic_install_dir>/user_projects/domains/SEE/` directory.

The server begins to start, and several messages are displayed. You are prompted to specify a username and password.

2. Type the username and password for starting the WebLogic server.

This information was specified during the domain installation (under [Creating the WebLogic Domain Using the SIMULIA Execution Engine Administration Server Template](#)).

Additional server messages are displayed. After a few initial messages, no information appears in this dialog box. Instead, all the startup information is written to the `AdminServer.log` file, which is located in the following directory:

```
<weblogic_install_dir>/user_projects/domains/<domain_name>/
servers/AdminServer/log/
```

When the server successfully completes its startup process, it writes the following message to the `AdminServer.log` file:

```
<Notice> <WebLogicServer> <hostname> <AdminServer> <000360>
<Server started in RUNNING mode>
```

3. Open a Web browser, and type the following address in the browser:

```
http://<computer-name>:7001/console
```

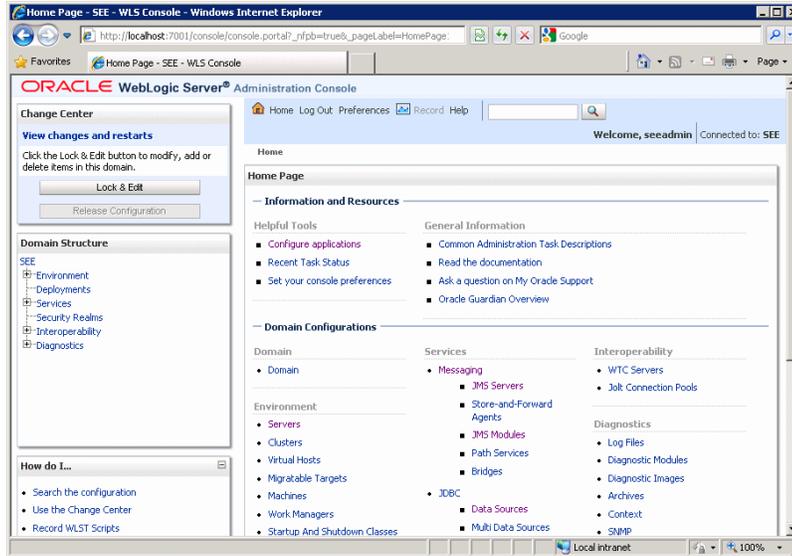


Note: The `<computer-name>` can either be the computer's name or the IP address on which the WebLogic Server is running.

The **Welcome** screen appears.

4. Enter your username and password in the corresponding text boxes, and click **Log In**.

The **Home** screen appears.



If you are using Internet Explorer 7 and you are unable to get past the login screen, despite entering a user name and password, you may need to update your browser's security settings. For more information, see [Updating Internet Explorer Settings for the WebLogic Administration Console](#).

If you are unable to view the Java-based navigation area on the left side of the console, you may need to install Java on your system. For more information, see [Installing Java for Internet Explorer](#).

Configuring Client Authentication

Security for a WebLogic-based SIMULIA Execution Engine is handled using client authentication. You can either connect to an LDAP Server for this authentication, if your environment already uses an LDAP server, or you can use WebLogic's embedded LDAP.

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 ID and password).

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

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

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

Important: The WebLogic configuration template automatically configures the default security realm, *myrealm*, with the following global security roles: *fiperuser*, *fiperstation*, and *fiperadmin*. For more information, see [About Security Roles](#).

About Security Roles

There are three types of SIMULIA Execution Engine user roles provided, each of which is identified by a security role. Each security role to which a user belongs enables that user to use a particular set of SIMULIA Execution Engine features.

The SIMULIA Execution Engine security roles are as follows:

- **fiperuser**. This role enables basic access to the SIMULIA Execution Engine system. All users (including administrators) must be associated with this security role. Users in this role can run the Isight Design Gateway and all other client programs except for SIMULIA Execution Engine stations. When using the Dashboard, users with this access level cannot view or change Access Control List configurations (see [About SIMULIA Execution Engine Access Control Lists](#)) or shutdown SIMULIA Execution Engine stations.
- **fiperadmin**. This role enables access to SIMULIA Execution Engine administrative features. Users with this security role must also have the **fiperuser** security role. SIMULIA Execution Engine administrators can use all features of the Dashboard, shutdown SIMULIA Execution Engine stations remotely using the Dashboard or the Command-Line Client, add and remove Federation (B2B) partners, and perform other restricted administrative functions.
- **fiperstation**. This role provides the ability to run a SIMULIA Execution Engine station. Users with this security role must also have the **fiperuser** security role and are the only users allowed to run a SIMULIA Execution Engine station (which can perform work on behalf of all users of the SIMULIA Execution Engine). In production deployments, this role will be given only to restricted user IDs created solely for the purpose of running the SIMULIA Execution Engine stations.

Using an LDAP Server for Client Authentication

To enable client authentication from LDAP, you configure the application server using the tools supplied in WebLogic. The server is configured to authenticate all incoming connection requests against a particular security back-end infrastructure—usually LDAP, but most application servers support many other security protocols. This topic describes how to perform this task for the LDAP security system, but your application server documentation should be consulted for all possible options and configurations.



Note: 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 [Mapping Security Roles with an LDAP Server](#). All other settings described in the next two sections ([Configuring the LDAP Connection](#) and [Testing the Connection to Your LDAP Server](#)) should be unchanged and do not need to be repeated.

The instructions in the following section assume the use of an LDAP server for authentication (specifically, Microsoft Active Directory; other LDAP servers would be configured in a similar manner). WebLogic 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 onto 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 WebLogic to use LDAP.

Configuring the LDAP Connection

The first step in configuring WebLogic for secure client authentication is to configure the LDAP connection, which defines the security infrastructure against which WebLogic will validate user IDs and passwords.

By default, WebLogic uses an Embedded LDAP Server for authenticating users. Optionally, you can configure most commercial LDAP servers for client authentication with the WebLogic server.

Note: If you do not want to use LDAP Server security for the SIMULIA Execution Engine, you still need to define users and roles in the Embedded LDAP Server supplied by the WebLogic Server to apply security policies to the SIMULIA Execution Engine. See [Using the WebLogic Embedded LDAP for Client Authentication](#) for instructions.

The following instructions assume the use of an LDAP server for authentication (specifically, Microsoft Active Directory Server). Other LDAP servers can be configured in a similar manner. For more information, please check your WebLogic Server documentation.

WebLogic Server's Embedded LDAP authentication may be adequate for small test environments, but it is not suitable for production deployments. Some familiarity with LDAP is helpful to properly configure the WebLogic Server to use LDAP.

Note: Using an LDAP Browser may be helpful while configuring an LDAP Connection with the WebLogic Server.

1. Click **Lock & Edit** in the upper left corner of the WebLogic Administration Console.
2. Click **Security Realms** on the left side of the console.

The **Summary of Security Realms** screen appears.

3. Click **myrealm** in the **Name** column on the right side of the console.
4. Click the **Providers** tab.

The **Authentication Providers** table appears.

5. Click **New**.

The **Create a New Authentication Provider** screen appears.

6. Type `ActiveDirectoryAuthenticator` in the Name text box.
7. Select **ActiveDirectoryAuthenticator** from the **Type** list.
8. Click **OK**.
9. Click the **ActiveDirectoryAuthenticator** link in the **Name** column.
10. Select SUFFICIENT from the **Control Flag** list.
11. Click **Save**.
12. Click the **Provider Specific** subtab.
13. Type the host name of your Active Directory Server in the Host text box.
14. Type the port number in the Port text box. The default port for an LDAP Server is 389.
15. Type the principal name of the user in the **Principal** text box. For example, the principal name of the user that started the WebLogic Server may be `SEE Admin`. The actual user ID may be different from the principal name defined in the LDAP server.
16. Type the password in the **Credential** text box, and retype the password in the **Confirm Credential** text box.
17. Type the base distinguished name of the user in the User Base DN text box. This setting is the value that classifies users for searching and location, including the user that starts the WebLogic Server.
Contact your local system administrator for the proper settings.
18. In the User From Name Filter text box, replace **cn** with `sAMAccountName`. For example, the entry should look similar to the following:

```
(&(sAMAccountName=%u)(objectclass=user))
```

19. Type `sAMAccountName` in the **User Object Class** text box replacing the existing text.
20. Type the base distinguished name of the group in the Group Base DN text box. This is the value under which all the groups are searched and located, including the group of the user that starts the WebLogic Server.

Contact your local system administrator for the proper settings.

21. Clear (uncheck) the **Cache Enabled** check box.
22. Click **Save**.
23. Click the **Providers** link at the top of the console.
24. Click the **DefaultAuthenticator** link in the **Name** column.

The **DefaultAuthenticator** screen appears.

25. Select SUFFICIENT from the **Control Flag** list.
26. Click **Save**.
27. Click **Activate Changes** in the upper left corner of the console.

A message appears stating that all of your changes have been activated. Now you need to stop and restart the WebLogic Server.

28. Click **SEE** at the top of the left side of the console.

Note: This link will have a different name if you did not use the provided domain name setting during the domain creation as described in *Creating the WebLogic Domain Using the SIMULIA Execution Engine Administration Server Template*.

29. Click the **Control** tab on the right side of the console.

A list of servers appears.

30. Click **AdminServer(admin)**.
31. Click **Shutdown**, and select **When work completes** from the list that appears.
32. Click **Yes** to verify the shutdown.

The server is stopped.

33. Perform one of the following actions, based on your operating system:

- **Windows:** Open a Command Prompt window, and execute the `startWebLogic.cmd` file in the `<weblogic_install_dir>/user_projects/domains/fiper` directory.
- **UNIX/Linux:** Open a terminal window, and execute the `./startWebLogic.sh` command in the `<weblogic_install_dir>/user_projects/domains/fiper` directory.

34. Type the username and password for starting the WebLogic server.
35. Refresh your Web browser.
36. Log back into the WebLogic Administration Console.

Testing the Connection to Your LDAP Server

To test your LDAP settings, you need to verify that you can see LDAP information in your WebLogic Administration Console.

1. Click **Security Realms** on the left side of the console.
The **Summary of Security Realms** screen appears.
2. Click **myrealm** in the **Name** column on the right side of the console.
3. Click the **Users and Groups** tab.
4. Click the **Groups** subtab.

You should be able to view group information defined on your LDAP server.

Mapping Security Roles with an LDAP Server

In this portion of the security configuration, the administrator defines what users (or groups of users) are allowed access to basic SIMULIA Execution Engine application functions and to other functions. Users without the correct access level will not be able to log onto the SIMULIA Execution Engine.

Configuring SIMULIA Execution Engine security with an LDAP server involves mapping the predefined security roles to actual principals (users or groups) in the LDAP server. The installation template assumes that you have defined the following user groups in your LDAP server—**fiperusers**, **fiperstations**, **fiperadmins**—and that each group contains the appropriate users.

If these user group mappings are already defined on your LDAP server, you do not need to manually configure the mappings. Proceed to [Finalizing the ActiveDirectoryAuthenticator Settings](#). If you are using different, custom user groups, as defined by your LDAP server, follow the procedure below to complete your user group mappings.

Before you begin: If you want to set up security using the embedded WebLogic LDAP server, you do not need to complete the procedure in this section. Proceed to [Using the WebLogic Embedded LDAP for Client Authentication](#).

1. Click **Lock & Edit** on the left side of the WebLogic Administration Console.
2. Click the **Roles and Policies** tab on the right side of the console.
3. In the **Name** column of the Roles table on the right side of the console, expand the **Global Roles** option.
4. Click the **Roles** link.
5. In the **Global Roles** table on the right side of the console, click the **fiperuser** link.

The **Edit Global Role** screen appears.

6. Click **Add Conditions**.
7. Verify that **Group** is selected from the **Predicate List** list.
8. Click **Next**.
9. In the **Group Argument Name** text box, type the appropriate group name.
10. Click **Add**.

The specified group is added to the text box at the bottom of the screen.

11. Click **Finish**.

An entry specifying your group name should now appear in the list on the **Edit Global Role** screen.

12. Click **Save**.
13. Click the **Global Roles** link at the top of the console.

The **Global Roles** table is displayed.

14. Repeat step 5 through step 13 for the **fiperadmin** and **fiperstation** roles, mapping them to the appropriate groups.

SIMULIA Execution Engine users now must supply a user ID and password to connect to the SIMULIA Execution Engine. The SIMULIA Execution Engine will authenticate the user credentials against the LDAP server.

15. Proceed to [Finalizing the ActiveDirectoryAuthenticator Settings](#) when all three roles have been mapped.

Finalizing the ActiveDirectoryAuthenticator Settings

Now that you have determined that your LDAP settings are correct, you need to set your **ActiveDirectoryAuthenticator** settings to their final configuration.

-
1. Click **Security Realms** on the left side of the console.

The **Summary of Security Realms** screen appears.

2. Click **myrealm** in the **Name** column on the right side of the console.
3. Click the **Providers** tab on the right side of the console.
4. Click the **ActiveDirectoryAuthenticator** link in the **Name** column.
5. Verify that **SUFFICIENT** is selected from the **Control Flag** list.



Note: These settings represent general security guidelines. For more information on the security settings required for your environment, contact your local system administrator.

6. Click **Save**.
7. Click the **Providers** link at the top of the console.
8. Click the **DefaultAuthenticator** link on the right side of the console.

The **DefaultAuthenticator** screen appears.

9. From the **Control Flag** list, select **OPTIONAL**.
10. Click **Save**.
11. Click **Activate Changes** in the upper left corner of the console.

A message appears stating that all of your changes have been activated. Now you need to stop and restart the server.

12. Click **SEE** at the top of the left side of the console.
13. Click the **Control** tab on the right side of the console.

A list of servers appears.

14. Click **AdminServer(admin)**.
15. Click **Shutdown**, and select **When work completes** from the list that appears.
16. Click **Yes** to verify the shutdown.

The Administration Server is stopped, as well as any managed servers that you specified.

17. Perform one of the following actions, based on your operating system:

- **Windows:** Open a Command Prompt window, and execute the `startWebLogic.cmd` file in the `<weblogic_install_dir>/user_projects/domains/fiper` directory.
- **UNIX/Linux:** Open a terminal window, and execute the `./startWebLogic.sh` command in the `<weblogic_install_dir>/user_projects/domains/fiper` directory.

18. Type the username and password for starting the WebLogic server.

The client authentication feature of your SIMULIA Execution Engine is now functional.



Note: Additional security options are available with your SIMULIA Execution Engine, including Run-As security and Access Control Lists (ACLs). Unlike client authentication, these security options are not necessary for running your SIMULIA Execution Engine. For more information on these options, see [Configuring Security](#).

19. Proceed to one of the following sections, based on your SIMULIA Execution Engine configuration:

- Multiple-server configuration. Proceed to [Creating a Managed Server Using the ServerTemplate](#).
- Single-server configuration: Proceed to [Deploying the SIMULIA Execution Engine EAR File](#).

Using the WebLogic Embedded LDAP for Client Authentication

You can implement client authentication using the LDAP function embedded within WebLogic.

Important: If you set up security using an external LDAP server, you do not need to complete the procedures in this section. Proceed to [Creating a Managed Server Using the ServerTemplate](#).

The following instructions describe steps to create basic users and groups for the SIMULIA Execution Engine to function properly. You may want to substitute the users and groups with your choices of users and groups.

1. Click the **Users and Groups** tab (under **Security Realms / myrealm**).

2. Click the **Users** subtab.

The **Users** table appears.

3. Click **New**.

The **Create a New User** screen appears.

4. Type a username in the Name text box (for example, `fiperuser1`).

5. Type `Default Fiper User` in the Description text box.

6. Type a password in the **Password** text box, and reenter the password in the **Confirm Password** text box.

7. Click **OK**.

The user is created and appears in the **Users** table.

8. Click the link that corresponds to the new user in the **Name** column of the **Users** table.

The **Settings for `user_name`** screen appears.

9. Click the **Groups** tab.

10. Select the groups that this user will be a member of (**fiperusers**, **fiperstations**, or **fiperadmins**). See [About Security Roles](#).

11. Click the  button to move the selected groups to the **Chosen** list.

12. Click **Save**.

13. Click the **Users and Groups** link near the top of the console.

The **Users** table appears.

14. Repeat step 3 through step 13 for each user you want to create.

The client authentication feature of your SIMULIA Execution Engine is now functional.



Note: Additional security options are available with your SIMULIA Execution Engine, including Run-As security and Access Control Lists (ACLs). Unlike client authentication, these security options are not necessary for running your SIMULIA Execution Engine. For more information on these options, see [Configuring Security](#).

15. Proceed to one of the following sections, based on your SIMULIA Execution Engine configuration:

- Multiple-server configuration. Proceed to [Creating a Managed Server Using the ServerTemplate](#).
- Single-server configuration: Proceed to [Deploying the SIMULIA Execution Engine EAR File](#).

About the WebLogic Embedded LDAP

You may want to apply security policies with the WebLogic Embedded LDAP Server to the SIMULIA Execution Engine if you do not have a standard LDAP authentication mechanism in your organization. Defining security with the WebLogic Embedded LDAP Server involves creating groups and users and mapping the roles to the users or groups.

It is better to create a group first if you are going to add users to that group. The WebLogic configuration template automatically configures the Default Authenticator (i.e., Embedded LDAP) of the *myrealm* security realm, with the following properties/attributes:

- Groups:
 - **fiperusers**
 - **fiperstations**
 - **fiperadmins**

For more information on these groups, see [Using the WebLogic Embedded LDAP for Client Authentication](#).

- Group membership (role conditions) in global security roles:
 - The **fiperusers** group is a member of the **fiperuser** global role.
 - The **fiperstations** group is a member of the **fiperstation** global role.
 - The **fiperadmins** group is a member of the **fiperadmin** global role.
- Users:
 - The user name is `fiperacs`.
 - The password is `fiperacs`.
 - The group membership is **fiperusers**, **fiperstations**, and **fiperadmins**.

Note: This default group membership enables the `fiperacs` user to perform all tasks. The **fiperusers** group membership is mandatory (for more information, see [User Security Identity Configuration](#)), but the other memberships can be removed, which limits the `fiperacs` user rights (for example, the user cannot start a station if the **fiperstations** membership is removed).

Defining a Managed Server

This section explains how to define servers within the Administration Server that will host your SIMULIA Execution Engine applications.

Creating a Managed Server Using the ServerTemplate

After you have installed your WebLogic domain and started the Administration Server, you need to define servers within the Administration Server that will host your SIMULIA Execution Engine applications.

Regardless of whether you are going to run your applications on the same host as the Administration Server (vertical deployment strategy), on different hosts (horizontal deployment strategy), or a combination of the two, you first need to create the Managed Servers in the Administration Server.



Important: Do not create a separate managed server for the Federation (B2B) feature. This feature is deployed to the same managed server created for the SIMULIA Execution Engine deployment.

-
1. On the left side of the WebLogic Administration Console, expand the **Environment** option.
 2. Click **Servers**.
 3. Click **ServerTemplate** on the right side of the console.
 4. Click **Clone**.
The **Clone a Server** screen appears.
 5. Perform the following actions to define the new managed server:
 - a) Type the name of the server in the **Server Name** text box. Each server within a domain must have a name that is unique for all configuration objects in the domain, and the name cannot be the same as the WebLogic server domain name.
 - b) If you want to limit the valid addresses for a server instance, type an IP address or DNS name in the **Server Listen Address** text box. Otherwise, URLs to the server can specify any of the host computer's IP addresses, any DNS name that maps to one of the IP addresses, or the localhost string.
 - c) Type the port number from which you want to access the server instance in the **Listen Port** text box. If you run multiple server instances on a single computer (vertical deployment), each server must use its own listen port. If possible, it is recommended that you use port 7101 for the first managed server, 7201 for the second managed server, etc.
 6. Review the configuration options you have chosen and click **OK**.
The server appears in the list of servers on the right side of the console.
 7. Click the server link in the **Name** column.
The settings for the server appear.
 8. Change the port number listed in the **SSL Listen Port** text box. It is recommended that you set this port to the setting in the **Listen Port** text box, but increment it by one. For example, if you are using port 7201 for the **Listen Port**, set the **SSL Listen Port** to 7202.
 9. Click **Save**.
 10. (optional) Return to step 3 to create additional managed servers.
 11. Perform one of the following actions, based on your deployment strategy:
 - Horizontal deployment: Proceed to [Setting the Staging Mode for Horizontally Deployed Managed Servers](#). You need to set the Staging mode for any horizontally deployed managed servers (even if you are using a mix of vertical and horizontal managed servers).
 - Vertical deployment: Proceed to [Setting Additional Server Targets](#).

Setting the Staging Mode for Horizontally Deployed Managed Servers

You need to set the staging mode for any horizontally deployed managed servers (even if you are using a mix of vertical and horizontal managed servers).

Note: These steps are not necessary if the managed server is being hosted on the same system as the Administration Server (vertical deployment).

1. Verify that you are viewing the Servers table on the right side of the WebLogic console (under **Environment / Servers**).
2. Click the name of the managed server instance (in the **Name** column) that you want to configure.
3. Click the **Deployment** subtab (under the main **Configuration** tab).
4. Verify that stage is selected from the **Staging Mode** list.
5. In the **Staging Directory Name** text box, specify the path where you would like to store staged deployment files. The path is relative to the root directory of the selected server. A default directory is provided for vertical

deployments, but for a horizontal deployment you must verify that the directory is correct on the system running the managed server.

6. Verify that the directory specified in step 5 exists.
7. If you are configuring the staging mode for the Administration Server, specify the Upload Directory Name setting, relative to the server's root directory. This is the directory where the Administration Server stores uploaded files for deployment to servers and clusters in the domain. Typically, this setting will point to the directory that contains the `.ear` or `.war` file that will be hosted by the managed server. These EAR files are located in the `server` and `web` subdirectories in the following directory:

```
<SEE_install_dir>/<os_dir>/reffiles/SMAFIPserver/weblogic/
```

8. Click **Save**.
9. Repeat step 1 through step 8 for all necessary servers.

Setting Additional Server Targets

If you are deploying your SIMULIA Execution Engine to a managed server (either vertical or horizontal), you must specify the managed server as a target for your WebLogic JMS Server, JMS Module, and your two database data sources.

1. Expand the **Messaging** option (under **Services**) on the left side of the console.
2. Click **JMS Servers**.

The **Summary of JMS Servers** screen appears.

3. Click the **FiperJMSServer** link in the **Name** column on the right side of the console.

The **Settings for FiperJMSServer** screen appears.

4. Click the **Targets** tab.
5. Select the managed server that will host the SIMULIA Execution Engine from the **Target** list.
6. Click **Save**.
7. Click **JMS Modules** (under **Services/Messaging**) on the left side of the console.

The **JMS Modules** screen appears.

8. Click the **Fiper-SystemModule** link in the **Name** column on the right side of the console.

The **Settings for Fiper-SystemModule** screen appears.

9. Click the **Targets** tab.

A list of servers appears.

10. Click the check box that corresponds to the managed server that will host the SIMULIA Execution Engine.
11. Clear (uncheck) the **AdminServer** check box. Only the managed server that will host the SIMULIA Execution Engine should be selected.
12. Click **Save**.

13. Expand the **JDBC** option (under **Services** on the left side of the console).

14. Click **Data Sources**.

The **Summary of JDBC Data Sources** screen appears.

15. Click the **Fiper nonXA Data Source** link in the **Name** column on the right side of the console.

The **Settings for Fiper nonXA Data Source** screen appears.

16. Click the **Targets** tab.

A list of servers appears.

17. Click the check box that corresponds to the managed server that will host the SIMULIA Execution Engine.
18. Clear (uncheck) the **AdminServer** text box. Only the managed server that will host the SIMULIA Execution Engine should be selected.
19. Click **Save**.
20. Click **Data Sources** on the left side of the console (under **JDBC**).
21. Repeat step 15 through step 19 for the **Fiper XA Data Source** entry.
22. Click **Activate Changes** in the upper left corner of the console.
23. Proceed to one of the following sections, based on your deployment:
 - If you are deploying the applications to different hosts (horizontal deployment), proceed to [Creating WebLogic Domains for Remote Managed Servers](#).
 - If you are deploying the applications to the same host as the Administration Server (vertical deployment), proceed to [Starting the Managed Servers](#).

Creating WebLogic Domains for Remote Managed Servers

For a multiple-server domain (horizontal deployment), you must create a WebLogic domain on the systems that will host the managed servers. This process uses a template and is similar to the domain creation performed for the WebLogic Administration Server.

1. Verify that you have WebLogic installed on the system that will host the managed server.
2. Verify that you have defined the managed servers in the Administration Server as described in [Creating a Managed Server Using the ServerTemplate](#).
3. Verify that the SIMULIA Execution Engine is installed on the system that will host the managed server.
4. Start the configuration wizard on the system that will host the managed server:
 - On Windows, click **Start**, point to **All Programs / Oracle WebLogic / WebLogic Server 11gR1 / Tools**, and click **Configuration Wizard**.
 - On UNIX/Linux, execute the `./config.sh` command in the `<weblogic_install_dir>/wlserver_10.3/common/bin/` directory.

The **Fusion Middleware Configuration Wizard** dialog box appears.

5. Verify that **Create a new WebLogic domain** is selected.
6. Click **Next**.

The **Select Domain Source** screen appears.

7. Click **Base this domain on an existing template**.
8. Click **Browse**, and navigate to the following directory:

```
<SEE_install_dir>/<os_dir>/reffiles/SMAFIPserver/weblogic/deploy/
```

9. Select the **ManagedServer.jar** file, and click **OK**.

You are returned to the Configuration Wizard.

10. Click **Next**.
11. If desired, change the domain name and location in the corresponding text box. It is recommended that you not change these default settings.

12. Click **Next**.

The **Configure Administrator User Name and Password** screen appears.

You are provided with default username and password settings (*seeadmin* and *seeadmin4*, respectively).

13. If desired, change the username or password. This information is used to start the server (in Production Mode) and to log into the WebLogic Administration Console (the browser-based interface).

14. Click **Next**.

The **Configure Server Start Mode and JDK** screen appears.

15. Click **Production Mode**.

16. Verify that the **Sun SDK** option is selected in the **Available JDKs** area. If it is not selected, click it to select it.

17. Click **Next**.

The **Select Optional Configuration** screen appears.

18. Click **Next**.

The **Configuration Summary** screen appears.

19. If desired, change the domain name and location in the corresponding text box. It is recommended that you not change these default settings and that you use the same domain name as used when creating the Administration Server.

20. Click **Create**.

The status of the operation is displayed.

21. Verify that the domain was created successfully.

22. Click **Done** when the operation is complete.

23. Edit the domain environment file as described in [Editing the Environment File](#).

Starting the Managed Servers

Once you have created the necessary Managed Servers—either on the same host as the Administration Server or on a different host in an installed WebLogic domain—you need to start them.

1. Verify that you are logged into the system that will host the Managed Server (either the same system that is hosting the Administration Server for vertical deployment or a different system for horizontal deployment).
2. Open a Command Prompt window (terminal window on UNIX/Linux), and navigate to the following directory:

```
<weblogic_install_dir>\user_projects\domains\fiper
```

3. Start the managed server.

- Use the following command on Windows:

```
startManagedWebLogic.cmd <managed_server> http://<admin_url>:7001
```

- Use the following command on UNIX/Linux:

```
startWebLogic.sh <managed_server> http://<admin_url>:7001
```

where *<managed_server>* specifies the name of the Managed Server and *<admin_url>* specifies the listen address (host name or IP address) and port number of the domain's Administration Server.

The servers specified here are those that you created earlier in *Creating a Managed Server Using the ServerTemplate*. The server names are case-sensitive.

For example, the following command starts a Managed Server named WebTop and the listen address for the domain's Administration Server is AdminHost:7001.

```
c:\bea\user_projects\domains\fiper\startManagedWebLogic.cmd
WebTop http://AdminHost:7001
```

The server begins to start, and several messages are displayed. You are prompted to specify a username and password.

4. Type the username and password for starting the WebLogic Administration Server. This information was specified during the Administration Server domain installation (under *Creating the WebLogic Domain Using the SIMULIA Execution Engine Administration Server Template*).

Additional server messages are displayed. After a few initial messages, no information appears in this dialog box. Instead, all the startup information is written to the `<Server_Name>.log` file, which is located in the following directory:

```
<weblogic_install_dir>\user_projects\domains\<domain_name>\servers\<Server_Name>\log\
```

where `<Server_Name>` is the name of the Managed Server.

When the server successfully completes its startup process, it writes the following message to the `<Server_Name>.log` file:

```
<Notice> <WebLogicServer> <hostname> <Server_Name> <000360>
<Server started in RUNNING mode>
```

5. For each Managed Server that you want to start, repeat steps 1 through 4. You should start each Managed Server in a different Command Prompt window (terminal window on UNIX/Linux).



Note: If you are starting Managed Servers on another machine, log in to that machine (remotely or locally) and then repeat steps 2 through 4.

6. Once you have set up all necessary managed servers and successfully started them, verify that they are running correctly by examining the **Summary of Servers** list in the WebLogic Administration Console (click the **Environment/Servers** option on the left side of the console). The list appears on the right side of the console. All servers that you have started should be in the RUNNING state.

Deploying the SIMULIA Execution Engine EAR File

You must deploy the EAR file for the SIMULIA Execution Engine application.

The first application that you should deploy is the SIMULIA Execution Engine itself. This application is the only one necessary to have a functional SIMULIA Execution Engine. All other applications provide additional features but are not required to have a fully functional system.

1. Click **Lock & Edit** on the left side of the WebLogic console.
2. Click **Deployments**.
3. Click **Install**.

The **Install Application Assistant** screen appears.

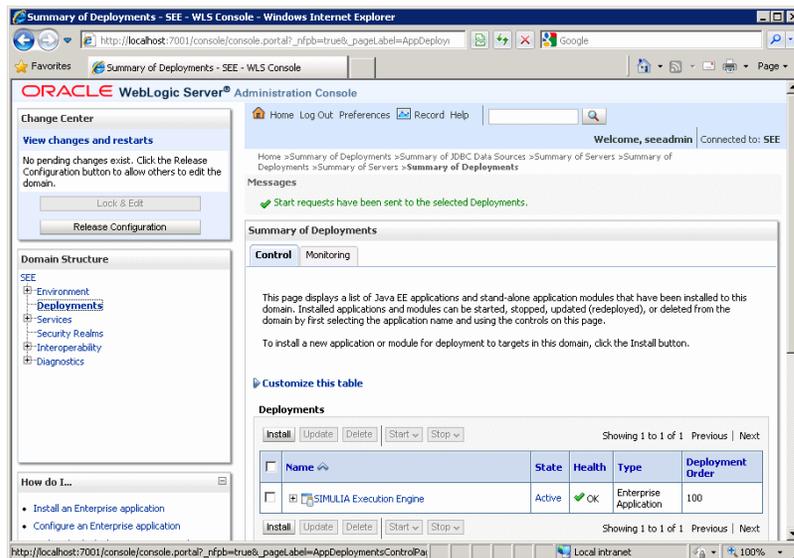
- Using the directory links, navigate to the directory where the `fiper.ear` file resides:

```
<SEE_install_dir>/<os_dir>/reffiles/SMAFIPserver/weblogic/
```

When you reach the appropriate directory, the EAR file is displayed.

- Click **fiper.ear**, and click **Next**.
- Verify that **Install this deployment as an application** is selected, and click **Next**.
The **Select deployment targets** screen appears.
- Specify the server that will host the application.
Note: If you are using the single-server deployment option, specify the **AdminServer** option.
- Click **Next**.
The **Optional Settings** screen appears.
- Type **SIMULIA Execution Engine** in the **Name** text box.
- Click **Next**.
- Click **No, I will review the configuration later**, and click **Finish**.
- Click **Activate Changes** in the upper left corner of the console.
- Click the check box that corresponds to the **SIMULIA Execution Engine** deployment in the **Deployments** table on the right side of the console.
- Click **Start**, and select **Servicing all requests** from the menu that appears.
- Click **Yes** to confirm the deployment.
- Verify that the deployment was successful.

It should be listed as *Active* in the **State** column of the **Deployments** table.



- Stop and restart the WebLogic Administration Server and any running managed servers. You can stop all of the managed servers at the same time.
- Proceed to one of the following topics, based on your installation:
 - If you are only installing the SIMULIA Execution Engine, proceed to [Configuring the Connection Profile](#).

-
- If you want to install other applications, including the WebTop, the WebDashboard, or the Federation feature, proceed to [Deploying the WebTop, WebDashboard, or Federation Applications](#).

Deploying the WebTop, WebDashboard, or Federation Applications

You must also deploy any other SIMULIA Execution Engine applications that you want to run: the WebTop, WebDashboard, or Federation (B2B). The process is similar to the one used for deploying the SIMULIA Execution Engine itself. However, these applications are optional and are not needed to have a fully functional system.

1. Click **Lock & Edit** on the left side of the WebLogic Administration Console.
2. Click **Deployments** on the left side of the console.
3. Click **Install**.

The **Install Application Assistant** screen appears.

4. Using the **Location** links, navigate to the following directory:

```
<SEE_install_dir>/<os_dir>/refiles/SMAFIPserver/weblogic/
```

Choose the file for the application you want to deploy:

- **webdashboard.war**
- **webtop.war**
- **federation.ear**

5. Click **Next**.
6. Verify that **Install this deployment as an application** is selected, and click **Next**.

The **Select deployment targets** screen appears.

7. Specify the Managed Server that will host the application. This server must be running before you attempt to deploy an application to it. For more information, see [Starting the Managed Servers](#).

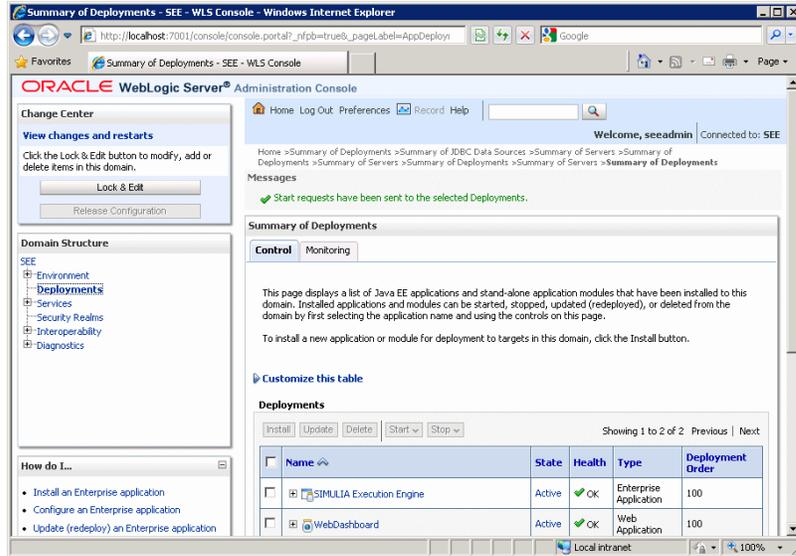
Important: If you are deploying the `federation.ear` file, you must select the same server as you did for the SIMULIA Execution Engine (under [Deploying the SIMULIA Execution Engine EAR File](#)).

8. Click **Next**.

The **Optional Settings** screen appears.

9. If desired, change the entry in the **Name** text box.
10. Click **Next**.
11. Click **No, I will review the configuration later**, and click **Finish**.
12. Click **Activate Changes** in the upper left corner of the console.
13. Click the check box that corresponds to the deployment in the **Deployments** table on the right side of the console.
14. Click **Start**, and select **Servicing all requests** from the menu that appears.
15. Click **Yes** to confirm the deployment.
16. Verify that the deployment was successful.

It should be listed as **Active** in the **State** column in the **Deployments** table.



17. Repeat step 1 through step 16 for all of the applications that you want to deploy.
18. Proceed to one of the following sections, based on your deployment:
 - WebTop or WebDashboard: Proceed to [Configuring the WebTop and WebDashboard Applications](#).
 - Federation. For complete details on configuring the Federation feature, refer to the SIMULIA Execution Engine Federation (B2B) Guide. Proceed to [Configuring the Connection Profile and Preloading the Library](#).

Configuring the WebTop and WebDashboard Applications

The method used to configure your SIMULIA Execution Engine web-based applications varies based on your deployment strategy.

Same-Server and Multiple Server (Vertical Deployment)

To configure the WebTop or WebDashboard for a vertical deployment, you must edit the corresponding properties file.

1. Navigate to the directory `<see_install_dir>/config/` in your installation.
2. In the text editor of your choice, open the `.properties` file that corresponds to the application you are configuring.
3. Locate the following entry in the file, where `<application>` is either `webtop` or `webdashboard`:

```
fiper.<application>.acs.cprfile=
```

4. Update the path following the = sign to point to the `.cpr` file that will be used to connect to the WebTop or WebDashboard (the `.cpr` file that matches the SIMULIA Execution Engine that the WebTop or WebDashboard will use). For example:

```
c:/simulia/ExecutionEngine/5.7/config/seecomputer.cpr
```

Important: Be sure to use forward slashes in your path.

-
5. Verify that the following two settings are commented out (have the # sign at the beginning of the setting) as shown below:

```
#fiper.<application>.acs.server=  
#fiper.<application>.acs.port=
```

6. Locate the `fiper.system.esihome=` entry.
7. Verify that the `fiper.system.esihome=` entry is correctly pointing to the top level of your SIMULIA Execution Engine installation directory. For example:

```
c:/simulia/ExecutionEngine/5.7
```

8. Save and close the file.
9. Stop and restart the managed servers. For more information, see [Stopping the SIMULIA Execution Engine Server](#).
10. Proceed to [Configuring the Connection Profile](#).

Multiple Server (Horizontal Deployment)

To configure the WebTop or WebDashboard for a horizontal deployment, you must edit the corresponding properties file.

1. Access the system that will host the application (not the system that is hosting the SIMULIA Execution Engine).
2. Verify that you have installed the SIMULIA Execution Engine.
3. Verify that you have created a WebLogic domain on the system that will host the web application as described in [Creating WebLogic Domains for Remote Managed Servers](#).
4. Navigate to the following directory:

```
<see_install_dir>/<os_dir>/code/bin/
```

5. Start the `editcpr.exe` file (`editcpr` file on UNIX/Linux). This file allows you to create a connection profile for connecting the WebTop or WebDashboard to a SIMULIA Execution Engine.
6. Create the new connection profile using the **Profile Editor** dialog box:
 - a) Type a name for the profile in the **Profile name** text box. Typically the name of the computer hosting the SIMULIA Execution Engine application is used.
 - b) Select **Weblogic Server 10.x** from the **Server type** list.
 - c) Type the name of the computer running the SIMULIA Execution Engine in the **Server name** text box.
 - d) Specify the **Listen Port** number for the managed server that is running the SIMULIA Execution Engine deployment.
 - e) Select **Save** from the **File** menu to save the connection profile. The **Save Connection Profile** dialog box appears.
 - f) Specify a file name, and click **Save**. This file name will be used in a later step.
 - g) Close the **Profile Editor** dialog box.
7. Navigate to the directory `<see_install_dir>/config/`.
8. Open the `.properties` file that corresponds to the application you are configuring in the text editor of your choice.
9. Locate the following entry in the file, where `<application>` is either `webtop` or `webdashboard`:

```
fiper.<application>.acs.cprfile=
```

-
10. Update the path following the = sign to point to the .cpr file that will be used to connect to the WebTop or WebDashboard. The path needs to point to the file created in step 6. For example:

```
c:/simulia/ExecutionEngine/5.7/config/seecomputer.cpr
```

Important: Be sure to use forward slashes in your path.

11. Verify that the following two settings are commented out (have the # sign at the beginning of the setting) as shown below:

```
#fiper.<application>.acs.server=  
#fiper.<application>.acs.port=
```

12. Locate the fiper.system.esihome= entry.
13. Verify that the fiper.system.esihome= entry is correctly pointing to the top level of your SIMULIA Execution Engine installation directory. For example:

```
c:/simulia/ExecutionEngine/5.7
```

14. Save and close the file.
15. If necessary, repeat step 7 through step 14 if you are installing both web-based applications on the same host. Otherwise, repeat step 1 through step 14 on another remote host.
16. Stop and restart the managed servers. For more information, see [Stopping the SIMULIA Execution Engine Server](#).

Restarting the SIMULIA Execution Engine in WebLogic

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

About Starting the SIMULIA Execution Engine Server

The SIMULIA Execution Engine is started and initialized when the WebLogic 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 startup scripts, described below, 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.
```

All SIMULIA Execution Engine WebLogic servers are started using startup scripts. When you use the SIMULIA Execution Engine configuration templates to create a domain, the domain directory of the host machine includes the following startup scripts:

-
- `startWebLogic`. This script starts the SIMULIA Execution Engine Administration Server. For more information on using this script, see [Starting the Administration Server](#).
 - `startManagedServer`. This script starts the Managed Servers (when using horizontal or vertical targeting). For more information on using this script, see [Starting the Managed Servers](#).

Important: This script does not use WebLogic Node Managers to start managed servers. In other words, the Administration Console cannot be used to start and stop any of the SIMULIA Execution Engine Managed Servers.

Stopping the SIMULIA Execution Engine Server

To stop the SIMULIA Execution Engine, you stop the WebLogic application server.

1. Stop the WebLogic application server using the normal WebLogic administration tools: either the Administration Console GUI command or shutdown scripts.

The server shutdown may 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
```

2. 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 can then be used to restart the SIMULIA Execution Engine. For more information, see [About Starting the SIMULIA Execution Engine Server](#).

About Stopping the SIMULIA Execution Engine Server

If you cannot gracefully shut down the SIMULIA Execution Engine using the WebLogic administration tools, you can resort to terminating the process in your operating system.

Each WebLogic server instance runs in its own JVM. It is highly recommended that all SIMULIA Execution Engine WebLogic servers be shutdown gracefully through the Administration Console. If you are unable to gracefully shutdown a server in this way, you can terminate the server JVM using the following methods:

- **Windows:** Type Ctrl-C in the Command Prompt window that you used to start the server, or use the Windows Task Manager to terminate the associated Java process.
- **UNIX/Linux:** Use the `ps` command to retrieve the server instance Java process ID, and use the `kill` command and the process ID to terminate the JVM.

Important: Terminating the JVM stops the server immediately and may lead to a loss of data.

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.

Configuring 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.

Configuring the Connection Profile

Depending on your installation, you will either need to edit the connection profile created during your SIMULIA Execution Engine installation, or you will need to create a new connection profile.

Proceed to one of the following topics:

- If you created a connection profile during the installation of the SIMULIA Execution Engine and you will use this connection profile to access the SIMULIA Execution Engine, proceed to [Editing a Connection Profile](#).
- If you created a connection profile during the installation of the SIMULIA Execution Engine, but you need to create a different connection profile to connect to the SIMULIA Execution Engine, proceed to [Creating a Connection Profile](#).

Editing a Connection Profile

Although a connection profile is created during the SIMULIA Execution Engine installation, you will need to verify and possibly edit the connection profile details to successfully connect to the SIMULIA Execution Engine.

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**.
- **UNIX/Linux:** Navigate to following directory and execute the `editcpr` file.

```
<see_install_dir>/<os_dir>/code/bin/
```

The **Profile Editor** dialog box appears.

2. From the **File** menu, select **Open**.

The **Open Connection Profile** dialog box appears.

3. Select the profile you need to edit. It should be located at the top level of your SIMULIA Execution Engine installation directory.

4. Click **Open**.

The connection profile details are loaded into the **Profile Editor** dialog box.

5. Verify that **Weblogic Server 10. x** is selected from the **Server type** list.

-
6. Verify that the name of the computer running the SIMULIA Execution Engine appears in the **Server name** text box.

Important: 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 type *seecomputer*, not *seecomputer.yourcompany.com*.

7. Do one of the following:
 - If you are using a multiple-server deployment, change the port number in the corresponding text box to match the **Listen Port** setting of the managed server you created to host the SIMULIA Execution Engine. This port number was specified in [Creating a Managed Server Using the ServerTemplate](#).
 - If you are using a single-server deployment, specify the **Listen Port** setting for the Administration Server. The default port number is usually 7001. For more information on the port numbers used by your network, contact your local system administrator.
8. Select **Save** from the **File** menu, and click **Yes** to verify your action.

The connection profile changes are saved.
9. Close the **Profile Editor** dialog box.

Creating a Connection Profile

If you have not already created a connection profile during the installation of the SIMULIA Execution Engine, you may need to do so now.

If you have existing connection profiles from a previous release of the 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**.
 - **UNIX/Linux:** Navigate to following directory and execute the `editcpr` file.

```
<see_install_dir>/<os_dir>/code/bin/
```

The **Profile Editor** dialog box 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 **Weblogic Server 10. x**.

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.

Important: 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 type *seecomputer*, not *seecomputer.yourcompany.com*.
5. Do one of the following:
 - If you are using a multiple-server deployment, change the port number in the corresponding text box to match the **Listen Port** setting of the managed server you created to host the SIMULIA Execution Engine. This port number was specified in [Creating a Managed Server Using the ServerTemplate](#).

-
- If you are using a single-server deployment, specify the **Listen Port** setting for the Administration Server. The default port number is usually 7001. For more information on the port numbers used by your network, contact your local system administrator.

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 on Windows or a terminal window on UNIX/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 into the SIMULIA Execution Engine.

You may need to edit or create a connection profile to successfully access the SIMULIA Execution Engine. For more information, see [Configuring the Connection Profile](#).



Important: If security is enabled on the SIMULIA Execution Engine, the user that logs into the SIMULIA Execution Engine to publish the components must be granted the Admin role in WebLogic. For more information, see [Configuring Client Authentication](#).

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 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.

<code>fiper.acs.name</code>	This setting defines the logical name of this SIMULIA Execution Engine. For single server systems it is the IP host name. For clustered servers it is the cluster (cell) name.
<code>fiper.system.esihome</code>	This setting points to the location of your SIMULIA Execution Engine installation.
<code>fiper.acs.isWindowsService</code>	This setting is used if the DRM is set to <code>lsf</code> and the <code>fiper.security.runas.drm</code> setting is set to <code>true</code> . 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 <code>fiper</code> , this setting is not used.
<code>fiper.system.filemgr.rootFilePath</code>	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: <ul style="list-style-type: none">• 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 WebLogic), though a NAS device is acceptable.• Can be written by the SIMULIA Execution Engine user ID (the one that is used to start WebLogic).• Is not a temporary file system. This disk space must never be reclaimed automatically. This rules out <code>/tmp</code>, <code>/var/tmp</code>, 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 <code>/export/ExecutionEngineFilemgr</code> instead of <code>/export</code>.
<code>fiper.cluster.controller.location</code>	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 WebLogic node and server names, not the IP (system) names.
<code>fiper.system.logfilter</code>	These settings are used for debugging purposes. They should be used only at the direction of SIMULIA technical support.

fiper.system.temp	<p>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 <code>/var/tmp</code> on UNIX/Linux. This directory needs to have the following characteristics:</p> <ul style="list-style-type: none"> • 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 WebLogic). If there is insufficient local storage space, a NAS device can be used, but this setup is not recommended.
fiper.system.drm	<p>These settings allows you to configure the distributed resource management option used by the SIMULIA Execution Engine. For more information on using LSF with the SIMULIA Execution Engine, see Using Distributed Resource Management with the SIMULIA Execution Engine.</p>
fiper.stranded.workitem.interval	<p>This setting defines the interval between checks for workitems left stranded in the PENDING state due to the lack of availability of a station with the required affinities. Workitems 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 workitem is waiting for a station that can process a workitem with the workitem’s affinities.</p>
fiper.system.drm.fipertimelimit	<p>This setting defines the 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. The default value is 0.</p>
fiper.system.bsubpath	<p>This setting is used only when <code>fiper.system.drm</code> is set to <code>lsf</code>. 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.</p>
fiper.system.lsfQueueName	<p>This setting is used if the DRM is set to <code>lsf</code>. This value specifies the name of the LSF queue to be used for SIMULIA Execution Engine jobs.</p>
fiper.acl.default.enabled	<p>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 <code>false</code> before the SIMULIA Execution Engine is started, any explicitly set default permissions are changed to <code>none</code>. The default setting is <code>true</code> so new users will not be accidentally locked out of their own new installations of SIMULIA Execution Engine.</p>

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 <code>fiper.ssl.keystore.file</code> 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 <code>true</code> , 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 <code>false</code> , 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 used for the WebLogic Federation feature. For more information, refer to the SIMULIA Execution Engine Federation (B2B) Guide.
fiper.security.runas.enabled	This setting is used with the Run-As security feature. For more information, see Configuring Station (Run-As) Security .
fiper.security.runas.domain	This setting is used with the Run-As security feature. For more information, see Configuring Station (Run-As) Security .
fiper.slm.url	This setting is used only if the SIMULIA Execution Engine is connecting to SIMULIA Scenario Definition.
fiper.acs.jms.persistent	<p>This setting specifies whether or not to use persistent JMS messaging between the station and the SIMULIA Execution Engine. The default value is <code>true</code>.</p> <p>For maximum reliability of the system, this setting should be left set to <code>true</code>. A value of <code>false</code> can result in job failures due to intermittent network connectivity issues or server failures that could otherwise be handled.</p> <p>There is a corresponding configuration entry in the <code>station.properties</code> file, <code>fiper.station.jms.persistent</code>, which must also be set to <code>true</code> to use persistent JMS messaging.</p>
fiper.acs.continueRunningJobs	<p>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 <code>true</code>.</p> <p>Generally, this setting should be set to <code>false</code> 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 <code>true</code> to be ready for the next time the SIMULIA Execution Engine is restarted.</p>

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.

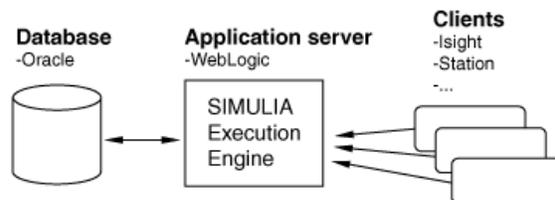


Figure 3: SIMULIA Execution Engine Architecture

The only access to the database is via the WebLogic 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. Users cannot open JDBC or ODBC connections to the database unless the database administrator has allowed it. 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.

The SIMULIA Execution Engine, however, requires that a security policy be defined in the WebLogic server. This action ensures that no user can gain access to the SIMULIA Execution Engine without first supplying credentials.

Administrators should be aware that any user that is authenticated by the SIMULIA Execution Engine server can perform any SIMULIA Execution Engine function on that server including running models, deleting job results, using the SIMULIA Execution Engine Dashboard to view lists of stations and running jobs, etc.

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 ID and password).

About SIMULIA Execution Engine Access Control Lists

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

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

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

The following permission levels are available:

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

- **READ.** The user or group can only load or use the object by reference. Although the model and its contents (components, simulation process flows, parameters, etc.) can be viewed and altered, and the model itself can be executed, no new versions of the model can be published to the library.
- **REFERENCE.** This protection level applies only to models stored in the SIMULIA Execution Engine library. This protection level is a limited read access that provides information about the model inputs and outputs but does not provide any access to the model structure or internal configuration. If this level of permission is set for a user who incorporates a published model into another model, the content of the referenced model is available and it can be executed by the reference.
- **NONE.** The user or group will have no access to the published object. Any model that references this object cannot be used.



Important: User names are case-sensitive.

For instructions on how to apply these permissions to library data and jobs, refer to 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.

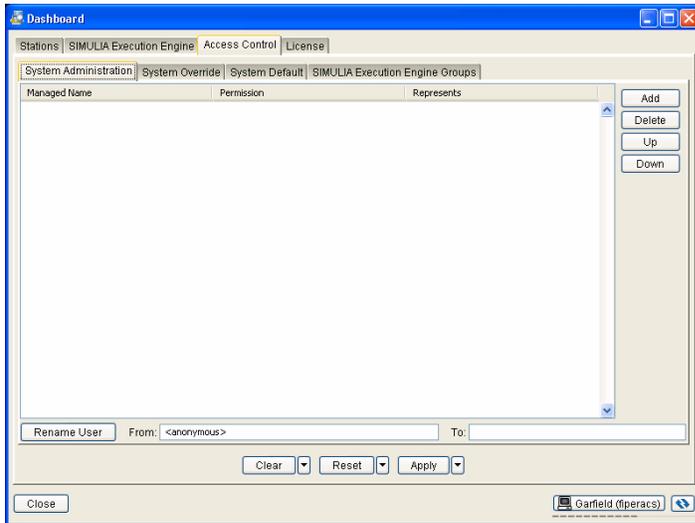


Figure 4: Access Control Tab on the SIMULIA Execution Engine Dashboard

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 `fiberadmin` security role. These users can add other ACL system administrators using the **System Administration** subtab; these users will have complete control over all published objects and assigned permissions in the SIMULIA Execution Engine but will not have any WebLogic administrative privileges.

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

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

Configuring Station (Run-As) Security

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

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

About Station Run-As Security

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

The term “security context” refers to the operating system level security information about a particular user (e.g., it is the operating system’s identification of a particular user and the user’s associated permissions to system resources such as files, network resources, etc.). When any program is started, the operating system associates the program (process) with the security context of a particular user (usually the user that started it). The process has access only to the resources that the user is authorized for at the operating system level. For example, the process would only be able to access files for which the user had appropriate file permissions.

When the Run-As feature is not active, all work is performed in a single operating system process, meaning that SIMULIA Execution Engine jobs could, in theory, access system resources (such as files) on behalf of a user when that user did not have permission to access the file. Through the SIMULIA Execution Engine infrastructure, users could retrieve a file from the SIMULIA Execution Engine station for which they would not typically have permission. In an extreme case it would be possible to write an Isight component (such as the Script component) to access other user’s in-progress work on the SIMULIA Execution Engine station where the component executes. This situation can be partially mitigated by running stations with restricted user IDs 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 (UNIX/Linux only).

Credentials (ID 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.

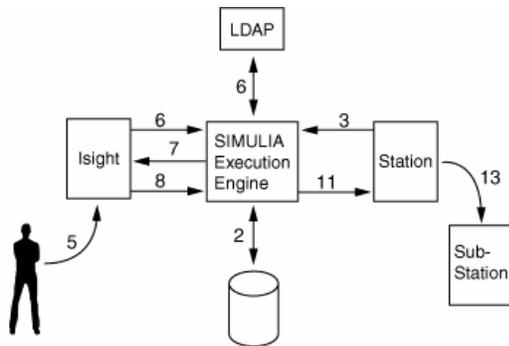


Figure 5: SIMULIA Execution Engine Encryption Process

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 ID 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 ID to run the SIMULIA Execution Engine station process. The restricted user ID 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 ID 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 ID, 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, all users must have the same user ID and password in both realms.
- When this feature is enabled and there are a mix of UNIX/Linux and Windows SIMULIA Execution Engine stations, users must have the same user ID and password in both the Windows security domain, and the UNIX/Linux environments.
- The Run-As feature is not supported on AIX 5.1 or AIX 5.2. Because Run-As is a system-wide feature, any SIMULIA Execution Engine environment that includes SIMULIA Execution Engine stations running on those operating systems will be unable to use this feature.
- On Windows, each station launched under Run-As security is automatically given its own Windows job, and all subprocesses created by the station, directly or indirectly, are associated with that job. This is done to guarantee that all subprocesses are terminated when the station is terminated (i.e., no failure in a user application can leave stray processes). Windows does not allow any process associated with a job to create a subprocess associated with a different job. Therefore, if you use Isight to run a program

that creates its own job for its own subprocesses, this program can fail to run if it is dispatched to a Run-As station. It will definitely fail if that station is running Windows 7 or Windows Vista. A workaround is to launch a station with Run-as security disabled and to configure the model to dispatch this program to that station (see the `station.properties` file to learn how to launch a station with Run-as security disabled).

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 Client Authentication](#).
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 WebLogic.
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 UNIX/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.

Important: 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 UNIX/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 XP*: Navigate to **Start / Control Panel / Performance and Maintenance / 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 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 Vista / 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. Restart the WebLogic server. For more information, see [Restarting the SIMULIA Execution Engine in WebLogic](#).

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 UNIX/Linux

To configure a UNIX/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 UNIX/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
```

Important: 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. To run a SIMULIA Execution Engine station with the Run-As feature on AIX 5.3, modify the `pam.conf` file as described below. (Run-As is not supported and will not work on AIX 5.1 or 5.2.) If these changes are not made, the SIMULIA Execution Engine station will not be able to authenticate the users that submit jobs.

Open the `/etc/pam.conf` file with a text editor, and do the following:

- a) Add the following line to the Authentication section, after the `telnet` entry:

```
fiper  auth      required      /usr/lib/security/pam_aix
```

- b) Add the following line to the Account Management section, after the `telnet` entry:

```
fiper  account required      /usr/lib/security/pam_aix
```

- c) Save your changes and close the text editor.

8. Repeat step 2 through step 7 (as necessary based on the computer's operating system) for each system that will be running a SIMULIA Execution Engine station.
9. Restart the WebLogic server.

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

Setting Station-Specific Run-As Options

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

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

For more information on setting this option, see [Station-Specific Run-As Behavior](#).

Using SIMULIA Execution Engine Interfaces

This section describes the interfaces that a system administrator can use to control and monitor the SIMULIA Execution Engine and the SIMULIA Execution Engine stations. The interfaces include the station itself, the Dashboard, 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 [Deploying the WebTop, WebDashboard, or Federation Applications](#) 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.

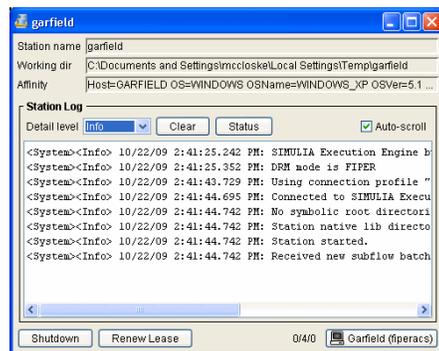
The SIMULIA Execution Engine station is a multithreaded Java application that can handle many concurrent requests. For an overview of the complete SIMULIA Execution Engine system, including stations, see .

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 for more details.

About the SIMULIA Execution Engine Station Interface

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

The station interface appears similar to the example below.



This interface provides you with the following information and options:

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

About Log Message Detail Levels

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

You can choose any of the following settings:

- **Debug**. These messages are intended for debugging system or component code. Typically, these messages are meaningful only to the program developers. This level can produce a large quantity of messages, which can affect system performance. This option is the lowest level, providing the most information. Using this setting will send all types of messages to your log file: debug, informational, warning, error, and system error.
- **Info**. These messages contain routine status or other informational items that are not generally significant. Choosing this setting will provide all **Info** messages as well as all **Warning**, **Error**, and **SysError** messages.
- **Warning**. These messages indicate a condition of which the end user should be aware but do not generally indicate a failure. Choosing this setting will provide all **Warning** messages as well as all **Error** and **SysError** messages.

-
- **Error**. These messages indicate an error condition that was caused by the end user, operational data, or some other condition that can be corrected. Choosing this setting will provide all **Error** messages as well as all **SysError** messages.
 - **SysError**. These messages indicate a software system failure. The message may indicate that some part of the infrastructure has become unusable (for example, a database has gone down), or it may indicate a programming error. These errors should be reported to system administrators for analysis. This option is the highest level, providing the smallest amount of messages. Using this setting will send only system error messages to your log file.

About Station Affinities

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

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

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

Custom Affinities

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

Required Affinities

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

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

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

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

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

Station Status Reporting in the Dashboard 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) as described in [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`
 - **UNIX/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 a Connection Profile](#).
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 UNIX/Linux), and navigate to one of the following directories, based on your operating system:
 - **Windows:** `<SEE_install_dir>\os_dir\code\command\`
 - **UNIX/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`)
 - **UNIX/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 Manually](#) (Windows) or [Installing a SIMULIA Execution Engine Station as a Service Manually](#) (UNIX/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 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.

For a quick reference listing of all station properties and command line arguments, see . 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.fcs.enginedir fiper.station.fcs.giveownership fiper.station.fcs.stagingdir	not available	SEE-SCE (ENOVIA) Integration

station.properties entry	Equivalent Command Line Argument	Full Description
fiper.station.jms.persistent	<i>not available</i>	<i>Persistent JMS Messaging</i>
fiper.station.leaseinterval	leaseinterval:<minutes>	<i>Lease Interval</i>
fiper.station.logfile	logfile:<log_file_name>	<i>Log File Location</i>
fiper.station.loglevel	loglevel:[debug info warn error syserror]	<i>Default Logging Level</i>
fiper.station.maxquiescetime	<i>not available</i>	<i>Shutdown Timeout Command</i>
fiper.station.name	name:<station_name>	<i>Station Name</i>
fiper.station.nogui	nogui:[true false]	<i>Running a Station Without a GUI</i>
fiper.station.retrydelay	<i>not available</i>	<i>Windows Service Retry Delay</i>
fiper.station.runas	<i>not available</i>	<i>Station-Specific Run-As Behavior</i>
fiper.station.saveLogOnError	<i>not available</i>	<i>LSF Station Error Logs</i>
fiper.station.serviceconfigfile	serviceconfigfile:<filename>	<i>Windows Service Configuration File</i>
fiper.station.tempdir	tempdir:<directory>	<i>Temporary Directory</i>
fiper.station.workdir	<i>not available</i>	<i>Working Directory</i>
fiper.logon.prompt	logonprompt:[yes no]	<i>Logon Prompt</i>
fiper.logon.profile	profile:<cpr_filename>	<i>Connection Profile</i>
fiper.logon.prop.user	user:<username>	<i>User Name</i>
fiper.logon.prop.pw	pw:<password>	<i>Password</i>
fiper.grid.ssh.configfile	<i>not available</i>	<i>Grid Distributed Execution</i>
fiper.file.RootA	<i>not available</i>	<i>Shared File System Symbolic Root Directories for File Parameters</i>
fiper.security.station.domain	<i>not available</i>	<i>Security Domain for Run-As Mode on Windows</i>
<i>not available</i>	configfile:<filename> or @filename	<i>Configuration File of Command Line Arguments</i>
<i>not available</i>	-help	<i>Command Line Argument Help</i>
<i>not available</i>	locale:<locale_string>	<i>Locale</i>
<i>not available</i>	logonmode:[GUI console]	<i>Logon Mode</i>

station.properties entry	Equivalent Command Line Argument	Full Description
<i>not available</i>	service:[stop start]	<i>Starting the Station as a Windows Service</i>
<i>not available</i>	unix_user:<user_id>	<i>Unix User for Extended Grid Credentials</i>
<i>not available</i>	unix_pw:<password>	<i>Unix Password for Extended Grid Credentials</i>
<i>not available</i>	win_user:<user_id>	<i>Windows User for Extended Grid Credentials</i>
<i>not available</i>	win_domain:<domain_name>	<i>Windows Domain for Extended Grid Credentials</i>
<i>not available</i>	win_pw:<password>	<i>Windows Password for Extended Grid Credentials</i>

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. See [About Station Affinities](#) for more details.

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

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\5.7\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\`
- **UNIX/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 (UNIX/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\5.7\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
```

Unix User for Extended Grid Credentials

This argument allows you to specify a UNIX/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>`

Unix Password for Extended Grid Credentials

This argument allows you to specify a UNIX/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 Manually](#) (for Windows stations) and [Installing a SIMULIA Execution Engine Station as a Service Manually](#) (for UNIX/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 Manually](#).

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 UNIX workstation and `/net/host/user` on other UNIX 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. For more information on using this tab and setting preferences, refer to the *Isight User's Guide*.

The following examples could be defined on Windows:

```
fiper.file.root.CDFfiles=//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 UNIX/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 UNIX/Linux. On UNIX/Linux, stations always authenticate against the domain configured by the system administrator.

Shutdown Timeout Command

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

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

Command Line Argument Help

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

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

Example:

```
station -help
```

Configuration File of Command Line Arguments

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

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

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

Command line examples:

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

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

```
profile:ExecutionEngine1 logonprompt:no
# provide user name and password to connect
user:seeadmin pw:seeadmin
```

You can now start a station connected to `ExecutionEngine1` with the following command:

```
station @see.txt
```

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

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

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

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

Locale

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

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

Command line examples to set the locale to German:

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

Persistent JMS Messaging

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

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

Property file example:

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

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

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

- `fiper.acs.jms.persistent`. This property must also be left set to `true` to change the server side of the infrastructure configuration to guarantee delivery. It does this by changing the delivery mode on the message producers to Persistent.
- `fiper.acs.jms.ttl`. This property sets the time-to-live value (in minutes) for persistent messages. The default value for this entry is 0, which means that the messages are kept in the cache until they are successfully delivered to all known recipients.

SEE-SCE (ENOVIA) Integration

Three properties in the `station.properties` file can be used to integrate SIMULIA Execution Engine (SEE) with SIMULIA Scenario Definition (SCE) for running simulations.

- **Property names:**
`fiper.station.fcs.enginedir`

```
fiper.station.fcs.giveownership
fiper.station.fcs.stagingdir
```

- **Command line argument:** not available

These properties should be set as follows:

- `fiper.station.fcs.enginedir`. The full path to the ENOVIA File Collaboration Server (FCS) engine directory.
- `fiper.station.fcs.giveownership`. The full path and command name of the ENOVIA File Collaboration Server (FCS) “giveownership.”
- `fiper.station.fcs.stagingdir`. The location of the checkin and checkout staging directory.

See the *SIMULIA Scenario Definition Administrator’s Guide* for more information.

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/5.7/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.

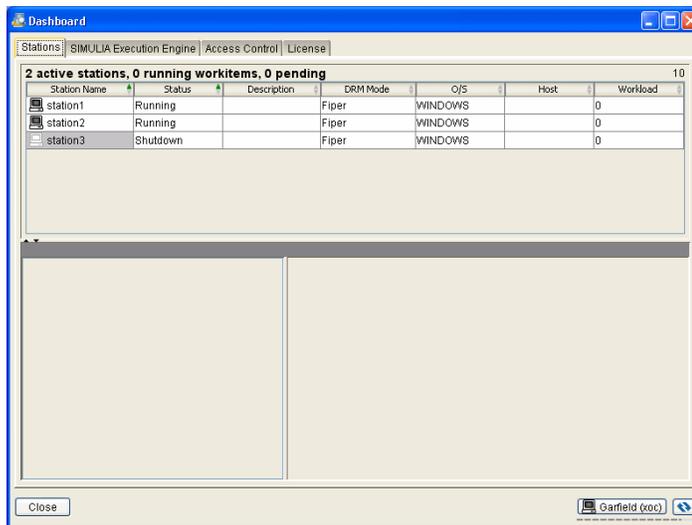
To run the Dashboard on a computer other than the one containing the SIMULIA Execution Engine installation, you must install the SIMULIA Execution Engine.

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 .

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.



Station Name	Status	Description	DRM Mode	O/S	Host	Workload
station1	Running		Fiper	WINDOWS		0
station2	Running		Fiper	WINDOWS		0
station3	Shutdown		Fiper	WINDOWS		0

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\bin\dashboard.exe`
 - **UNIX/Linux:** `<SEE_install_dir>/<os_dir>/code/bin/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\5.7\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 UNIX/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 UNIX/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, refer to 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.

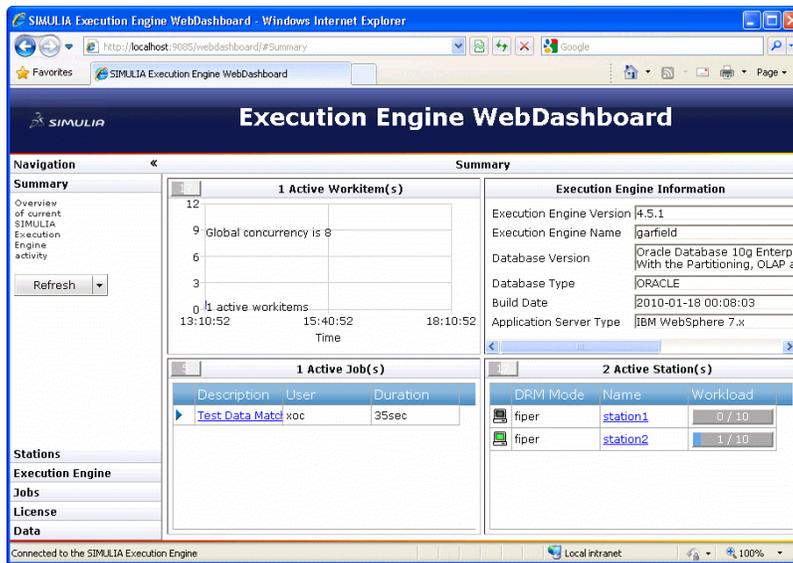
Using the WebDashboard

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.

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 <i>Job Name</i> 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. <ul style="list-style-type: none">• 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 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 <i>Job Name</i> 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 allow you 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. 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, refer to 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`
- **UNIX/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 a Connection Profile](#).

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, refer to 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, refer to 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 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 the OS Command, Simcode, and Abaqus 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 5.7 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:

-
- **UNIX/Linux:** WebLogic cannot be installed as root unless the Run-As feature is enabled. WebLogic 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, WebLogic can run as root.
 - You must install the SIMULIA Execution Engine normally and deploy on WebLogic.
 - 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 UNIX/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. For more information, see [Installing a SIMULIA Execution Engine Station on Windows](#) or [Installing a SIMULIA Execution Engine Station on UNIX and Linux](#).

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 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.

For additional information on the use of Mixed-Mode DRM and setting the DRM mode for Isight components, see the Configuring the Component Properties section of 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 a Connection Profile](#).
- 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 one of the following directories:

```
<SEE_install_dir>\intel_a\code\bin\      (for Windows 32-bit)
<SEE_install_dir>\win_b64\code\bin\    (for Windows 64-bit)
```

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\5.7\win_b64\code\bin
```

3. If you are installing on a UNIX/Linux system:

Important: This entry assumes that all the UNIX/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 UNIX/Linux system and share the installation on all the UNIX/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/5.7/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/5.7/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/5.7/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/bsub.exe
```

11. Save and close the file.
12. Create a connection profile for the SIMULIA Execution Engine. For more information, see [Creating a Connection Profile](#).
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`, `badadmin reconfig`, `badadmin mbdrestart`).
14. If you are installing on a UNIX/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 and Configuring a Local User on Windows

Prior to installing and accessing the SIMULIA Execution Engine on Windows, you must set up and configure a new local user. The purpose of this section is to walk you through the steps of creating a single default user for the application server and the database. 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 New User

You must set up a local user, 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: Prerequisite: 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 this example the display name `Fiper User` and the user name `seeadmin` are 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.

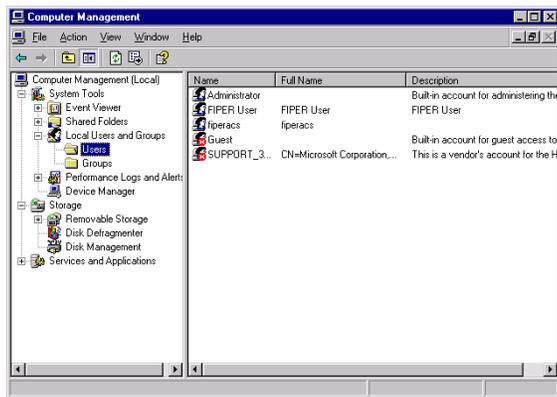
5. Type the name of the new user in the **User name** text box; for example, `seeadmin`.
6. If desired, type the full name of the user in the corresponding text box.
7. Type the user's description in the corresponding text box; for example, `FIPER User`.

8. Type the user's password 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 installation purposes only. Once installation is complete, you should convert the user to a regular (standard) 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 New User Privileges](#).

Setting the New User Privileges

You may need to configure the new user to function in the Windows environment.

The following guidelines should be used when determining if your user has sufficient privileges:

- Verify that the new user is a member of the Administrators group as described in [Creating the New User](#). This group assignment is necessary to administer WebLogic and the SIMULIA Execution Engine database.
- If the new 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 new 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 at 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 `fiperacs`, the password could not contain `fip`, `acs`, or even `era`.

User Security Identity Configuration

You must configure the security identity of the `seeadmin` user prior to installing an application server and database.

Important: The following information pertains to the `seeadmin` user that is used throughout this installation. It is applicable to all supported operating systems and must be adhered to install a fully functional SIMULIA Execution Engine.

The SIMULIA Execution Engine database employs a reliable asynchronous messaging infrastructure that is based on the JMS (Java Message Service) standard, which allows its application components to create, send, receive, and read messages. When a SIMULIA Execution Engine message is received from a JMS topic, the SIMULIA Execution Engine Server (J2EE) uses a credential map, also known as the J2EE RunAs Security model, to obtain the security identity (user name and password) used when establishing the JMS connection. This credential mapping occurs only once, and the JMS server uses the established security identity to retrieve all messages.

The WebLogic SIMULIA Execution Engine application deployment descriptor has already been preconfigured with the `seeadmin` user, whose valid security identity (user name and password) must be known within the target security realm of the user authentication registry (such as LDAP). The `seeadmin` user is the assigned J2EE RunAs user, and its existence is required for the SIMULIA Execution Engine asynchronous messaging infrastructure to function properly.

An additional requirement must also be adhered to when creating the `seeadmin` user. The 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 Client Authentication](#).

For further and detailed assistance on implementing these user security settings, especially with regard to working with your local user authentication registry, contact your local system administrator or WebLogic administrator.

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 10g or 11g

You use the Oracle Database Configuration Assistant to create the database for your SIMULIA Execution Engine. The steps provided here are the same for Oracle 10g and Oracle 11g databases unless specified. 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 1Xg/ Configuration and Migration Tools**, and click **Database Configuration Assistant**.
 - UNIX/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 the following is selected:
 - Oracle 11g database: **General Purpose or Transaction Processing**
 - Oracle 10g database: **General Purpose**

6. Click **Next**.

The **Database Identification** screen appears.

7. Enter the database name in the **Global Database Name** text box (for example, `Fipex`).
8. Verify that the database name appears in the **SID** text box.
9. Click **Next**.

The **Management Options** screen appears.

10. Verify that the following is selected:
 - Oracle 11g database: **Configure Enterprise Manager**
 - Oracle 10g database: **Configure the Database with Enterprise Manager**
11. Verify that the following is selected:
 - Oracle 11g database: **Configure Database Control for local management**

-
- Oracle 10g database: Use Database Control for Database management
12. Click **Next**.

The **Database Credential** screen appears.
 13. Verify that the following is selected:
 - Oracle 11g database: **Use the Same Administrative Password for All Accounts**
 - Oracle 10g database: **Use the Same Password for All Accounts**
 14. Enter the password (for example, f1peracs) 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. If you are creating an Oracle 11g database, 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. If you are creating an Oracle 10g database, do the following:
 - a) Click **Custom**.
 - b) Click **Manual** for the **Shared Memory Management** option, and alter the following values, as necessary, in the corresponding text boxes and lists:
 - **Shared Pool: 160, M Bytes**
 - Buffer Cache: 80, M Bytes**
 - Java Pool: 48, M Bytes**
 - Large Pool: 24, M Bytes**
 - PGA Size: 24, M Bytes**

-
26. Click the **Sizing** tab.
 27. Type 300 in the **Processes** text box.
 28. Click the **Character Sets** tab.
 29. Click **Use Unicode (AL32UTF8)**.
 30. In the **National Character Set** list, select the following:
 - Windows: **UTF8 - Unicode 3.0 UTF-8 Universal character set**
 - UNIX/Linux: **UTF**
 31. Click the **Connection Mode** tab.
 32. Verify that **Dedicated Server Mode** is selected.
 33. If you are creating an Oracle 11g database, do the following:
 - a) Click **Next**.
The **Security Settings** screen appears.
 - b) Click **Next**.
The **Automatic Maintenance Tasks** screen appears.
 34. Click **Next**.
The **Database Storage** screen appears.
 35. Click **Next**.
The **Create Options** screen appears.
 36. Verify that **Create Database** is selected.
 37. Click **Finish**.
A **Confirmation** message appears.
 38. Click **OK**.
A message appears when the database is created.
 39. 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`).
 40. Click **Exit**.
For more information on using the Enterprise Manager, see [Initializing an Oracle Database](#).

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 UNIX/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` (UNIX/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
```

- UNIX/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 WebLogic:

- **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 XP/Windows Server 2003:** `C:\Documents and Settings\<user_name>\local settings\temp\<station_name>`

-
- **Windows Vista / Windows Server 2008:** C:\Users*<user_name>*\AppData\Local\Temp
 - **UNIX/Linux:** /tmp/*<station_name>*
- **<server_name>.log.** The WebLogic server-generated files contain application server startup, status, and execution information. They can help when debugging WebLogic errors. All server startup information and messages are written to the *<Server_Name>.log* file, which is located in the following directory:
<weblogic_install_dir>\user_projects\domains<domain_name>*\servers*<Server_Name>*\log*
where *<Server_Name>* is AdminServer for the administration server instance or the name of the managed server for each configured managed server instance.

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 WebLogic](#).

Configuring the Windows Firewall for WebLogic

You need to open several ports to ensure that your WebLogic-based SIMULIA Execution Engine will function correctly behind the Windows Firewall.

Important: These instructions assume that you are using WebLogic's default port settings. To ensure that your port settings match those in this procedure, contact your local WebLogic 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: WL_administration
- Port number: 9002

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: `WL_listener_nonsecure`
 - Port number: 7001
 - Name: `WL_listener_secure`
 - Port number: 7002
9. Click **OK** to close the **Windows Firewall** dialog box.
10. Click **OK** to close the **Properties** dialog box.

Updating Internet Explorer Settings for the WebLogic Administration Console

Depending on your Internet Explorer configuration, you may need to update the browser's settings to access and correctly view the WebLogic Administration Console.

You may need to update your Internet Explorer settings as discussed in:

Adding the Console Address to the Trusted Sites List

The default Internet Explorer 7 security settings will not allow you to access the WebLogic Administration Console. You will be unable to get past the login screen, despite entering a user name and password. To fix this problem, you need to add your server Web site address to your trusted sites list.

1. Open Internet Explorer.
2. Click **Tools** on the right side of the browser's toolbar, and select **Internet Options**.

The **Internet Options** dialog box appears.

3. Click the **Security** tab.
4. Click the **Trusted sites** icon at the top of the tab.
5. Click **Site**.

The **Trusted sites** dialog box appears. The address of your server should appear automatically in the first text box, similar to the following: `http://seecomputer` .

6. Verify that the address appears in the **Add this website to the zone** text box or add it manually.
7. Click **Add**.

The address now appears on the **Web sites** list.

8. Click **Close**.
9. Click **OK**.
10. To access WebLogic, type the following address:

```
http://<computer-name>:7001/console
```

Note: The `<computer-name>` can either be the computer's name or the IP address on which the WebLogic Server is running.

Installing Java for Internet Explorer

You must have Java installed on your system to view the left side (navigator) of the WebLogic Administration Console.

Access the following Web site:

<http://www.java.com/en/>

Click **Free Java Download**, and follow the instructions provided by the Java installation software.

Note: If you have trouble downloading Java because of Internet Explorer's security settings, you can download Java using another Web browser. Alternatively, you can download Java on another system and copy the installation file to your SIMULIA Execution Engine system.

To access WebLogic, type the following address:

```
http://<computer-name>:7001/console
```

Note: The *<computer-name>* can either be the computer's name or the IP address on which the WebLogic Server is running.

The left side of the console should now load normally.

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.

UNIX/Linux-based SIMULIA Execution Engine Stops Functioning Correctly

If you installed the SIMULIA Execution Engine on a UNIX/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 Software](#).

The default setting for this directory is `/tmp`. However, some temporary directories on UNIX/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.

- 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 UNIX/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>/bin` directory.

On Windows, the password is stored in the `Wrapper.conf` file, which is located in the `<SEE_install_directory>\bin\win32` 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.

WebLogic Client JAR File

The WebLogic client jar files are not distributed with the SIMULIA Execution Engine. You may follow the instructions on the web site below, using the client JAR files that were provided with your installation of WebLogic version 10.

The single WebLogic client JAR file (`wlfullclient.jar`) for JDK 1.6 can be built by following the directions on this web site:

```
http://download.oracle.com/docs/cd/E12840_01/wls/docs103/client/jarbuilder.html#wp1078098
```

Place the JAR file in this directory:

```
<fiperinstall>/server/weblogic10/client/
```

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. For more information, see [Publishing to 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.

Otherwise, the SIMULIA Execution Engine is now ready for use.

Index

C

Creating the Database
Oracle 10g 143

F

File extensions
Displaying in Windows 2000 11

G

Group Base DN 62

P

publishall command 81

S

SIMULIA Execution Engine stations 6, 96
StartWebLogic.cmd file
Editing 57, 58

U

User Base DN 62

About SIMULIA

SIMULIA is the Dassault Systèmes brand that delivers a scalable portfolio of Realistic Simulation solutions including the Abaqus product suite for Unified Finite Element Analysis; multiphysics solutions for insight into challenging engineering problems; and lifecycle management solutions for managing simulation data, processes, and intellectual property. By building on established technology, respected quality, and superior customer service, SIMULIA makes realistic simulation an integral business practice that improves product performance, reduces physical prototypes, and drives innovation. Headquartered in Providence, RI, USA, with R&D centers in Providence and in Vélizy, France, SIMULIA provides sales, services, and support through a global network of regional offices and distributors. For more information, visit www.3ds.com/simulia.

About Dassault Systèmes

As a world leader in 3D and Product Lifecycle Management (PLM) solutions, Dassault Systèmes brings value to more than 100,000 customers in 80 countries. A pioneer in the 3D software market since 1981, Dassault Systèmes develops and markets PLM application software and services that support industrial processes and provide a 3D vision of the entire lifecycle of products from conception to maintenance to recycling. The Dassault Systèmes portfolio consists of CATIA for designing the virtual product, SolidWorks for 3D mechanical design, DELMIA for virtual production, SIMULIA for virtual testing, ENOVIA for global collaborative lifecycle management, and 3DVIA for online 3D lifelike experiences. Dassault Systèmes' shares are listed on Euronext Paris (#13065, DSY.PA), and Dassault Systèmes' ADRs may be traded on the US Over-The-Counter (OTC) market (DAST4). For more information, visit www.3ds.com.

Isight, 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.

© Dassault Systèmes, 2012

