

Abaqus 6.10

Installation and Licensing Guide



Abaqus

Installation and Licensing

Guide

Legal Notices

CAUTION: This documentation is intended for qualified users who will exercise sound engineering judgment and expertise in the use of the Abaqus Software. The Abaqus Software is inherently complex, and the examples and procedures in this documentation are not intended to be exhaustive or to apply to any particular situation. Users are cautioned to satisfy themselves as to the accuracy and results of their analyses.

Dassault Systèmes and its subsidiaries, including Dassault Systèmes Simulia Corp., shall not be responsible for the accuracy or usefulness of any analysis performed using the Abaqus Software or the procedures, examples, or explanations in this documentation. Dassault Systèmes and its subsidiaries shall not be responsible for the consequences of any errors or omissions that may appear in this documentation.

The Abaqus Software is available only under license from Dassault Systèmes or its subsidiary and 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.

The Abaqus Software is a product of Dassault Systèmes Simulia Corp., Providence, RI, USA.

© Dassault Systèmes, 2010

Abaqus, the 3DS logo, SIMULIA, CATIA, and Unified FEA are trademarks or registered trademarks of Dassault Systèmes or its subsidiaries in the United States and/or other countries.

Other company, product, and service names may be trademarks or service marks of their respective owners. For additional information concerning trademarks, copyrights, and licenses, see the Legal Notices in the Abaqus 6.10 Release Notes and the notices at: http://www.simulia.com/products/products_legal.html.

Locations

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.simulia.com>
SIMULIA European Headquarters Gaetano Martinolaan 95, P. O. Box 1637, 6201 BP Maastricht, The Netherlands, Tel: +31 43 356 6906,
Fax: +31 43 356 6908, simulia.europe.info@3ds.com

Technical Support Centers

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, Tel: +1 248 349 4669, simulia.greatlakes.info@3ds.com
Woodbury, MN, Tel: +1 612 424 9044, simulia.central.support@3ds.com
Beachwood, OH, Tel: +1 216 378 1070, simulia.erie.info@3ds.com
West Chester, OH, Tel: +1 513 275 1430, simulia.central.support@3ds.com
Warwick, RI, Tel: +1 401 739 3637, simulia.east.support@3ds.com
Lewisville, TX, 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 Huiizen, The Netherlands, Tel: +31 35 52 58 424, 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
Czech & Slovak Republics Synerma s. r. o., Psáry, Prague-West, Tel: +420 603 145 769, abaqus@synerma.cz
Finland Vantaa, Tel: +358 46 712 2247, 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
Greece 3 Dimensional Data Systems, Crete, Tel: +30 2821040012, support@3dds.gr
India Chennai, Tamil Nadu, Tel: +91 44 43443000, simulia.in.info@3ds.com
Israel ADCOM, Givataim, Tel: +972 3 7325311, shmulik.keidar@adcomsim.co.il
Italy Lainate MI, Tel: +39 02 39211211, simulia.ity.info@3ds.com
Japan Tokyo, Tel: +81 3 5442 6300, simulia.tokyo.support@3ds.com
Osaka, Tel: +81 6 4803 5020, simulia.osaka.support@3ds.com
Yokohama-shi, Kanagawa, Tel: +81 45 470 9381, isight.jp.info@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
Malaysia WorleyParsons Advanced Analysis, Kuala Lumpur, Tel: +603 2039 9000, abaqus.my@worleyparsons.com
New Zealand Matrix Applied Computing Ltd., Auckland, Tel: +64 9 623 1223, abaqus-tech@matrix.co.nz
Poland BudSoft Sp. z o.o., Poznań, Tel: +48 61 8508 466, info@budsoft.com.pl
Russia, Belarus & Ukraine TESIS Ltd., Moscow, Tel: +7 495 612 44 22, info@tesis.com.ru
Scandinavia Västerås, Sweden, Tel: +46 21 150870, simulia.nordic.info@3ds.com
Singapore WorleyParsons Advanced Analysis, Singapore, Tel: +65 6735 8444, abaqus.sg@worleyparsons.com
South Africa Finite Element Analysis Services (Pty) Ltd., Parklands, Tel: +27 21 556 6462, feas@feas.co.za
Spain & Portugal Principia Ingenieros Consultores, S.A., Madrid, Tel: +34 91 209 1482, simulia@principia.es
Taiwan Simutech Solution Corporation, Taipei, R.O.C., Tel: +886 2 2507 9550, lucille@simutech.com.tw
Thailand WorleyParsons Advanced Analysis, Singapore, Tel: +65 6735 8444, abaqus.sg@worleyparsons.com
Turkey A-Ztech Ltd., Istanbul, Tel: +90 216 361 8850, info@a-ztech.com.tr
United Kingdom Warrington, Tel: +44 1 925 830900, simulia.uk.info@3ds.com
Sevenoaks, Tel: +44 1 732 834930, simulia.uk.info@3ds.com

Complete contact information is available at <http://www.simulia.com/locations/locations.html>.

Preface

This section lists various resources that are available for help with using Abaqus Unified FEA software.

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 Abaqus are offered through a network of local support offices. Regional contact information is listed in the front of each Abaqus manual and is accessible from the **Locations** page at www.simulia.com.

SIMULIA Online Support System

The SIMULIA Online Support System (SOSS) provides a knowledge database of SIMULIA Answers. The SIMULIA Answers are solutions to questions that we have had to answer or guidelines on how to use Abaqus, SIMULIA SLM, Isight, and other SIMULIA products. You can also submit new requests for support in the SOSS. All support incidents are tracked in the SOSS. If you are contacting us by means outside the SOSS to discuss an existing support problem and you know the incident number, please mention it so that we can consult the database to see what the latest action has been.

To use the SOSS, you need to register with the system. Visit the **My Support** page at www.simulia.com to register.

Many questions about Abaqus can also be answered by visiting the **Products** page and the **Support** page at www.simulia.com.

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.

Training

All offices and representatives offer regularly scheduled public training classes. We also provide training seminars at customer sites. All training classes and seminars include workshops to provide as much practical experience with Abaqus as possible. For a schedule and descriptions of available classes, see www.simulia.com or call your local office or representative.

Feedback

We welcome any suggestions for improvements to Abaqus software, the support program, or documentation. We will ensure that any enhancement requests you make are considered for future releases. If you wish to make a suggestion about the service or products, refer to www.simulia.com. Complaints should be addressed by contacting your local office or through www.simulia.com by visiting the **Quality Assurance** section of the **Support** page.

Contents

1. Overview	
2. Installing Abaqus	
Installation procedures	2.1
Abaqus documentation installation details	2.2
Abaqus licensing installation details	2.3
Abaqus product installation details	2.4
Uninstalling Abaqus licensing or products	2.5
Uninstalling Abaqus documentation	2.6
3. Abaqus licensing	
Network licensing	3.1
Updating an Abaqus license file	3.2
Upgrading a license server and administration tools	3.3
Redundant license server configurations	3.4
FLEXnet license server manager lmgrd	3.5
FLEXnet Licensing administration tools	3.6
Using the options file	3.7
Reporting license usage data	3.8
4. Customizing the Abaqus environment	
Using the Abaqus environment file	4.1
Defining analysis batch queues	4.2
Examples	4.3
5. Customizing Abaqus/CAE and Abaqus/Viewer	
Customizing the user interface	5.1
Configuring printers	5.2
Tuning graphics cards	5.3
A. System requirements	
Software	A.1
Disk space	A.2
Memory	A.3
Graphics	A.4
Compilers	A.5

CONTENTS

MPI	A.6
3Dconnexion motion controllers	A.7
B. Abaqus release directories and files	
The SIMULIA parent directory	B.1
The Abaqus parent directory	B.2
Directory structure for Abaqus 6.7 and earlier	B.3
C. Configuring Abaqus documentation	
Using an existing non-Abaqus web server	C.1
Setting the documentation URL after installation	C.2
D. Accessing remote file systems for installation and execution	
Accessing the installation files from a remotely mounted DVD	D.1
Running Abaqus remotely	D.2
Using a network ODB connector	D.3
E. Verification procedure	
F. Troubleshooting Abaqus network licensing	
General questions	F.1
License server operation	F.2
Special situations	F.3
Other licensing questions and problems	F.4

1. Overview

This guide contains information about installing and customizing Abaqus 6.10. This guide is also accessible from the **Documentation** section of the **Support** page at www.simulia.com.

The installation of Abaqus involves the following tasks:

- Installing the Abaqus documentation.
- Installing the Abaqus license server.
- Installing Abaqus products.

For a description of the installation procedures, see Chapter 2, “Installing Abaqus.”

Network licensing and procedures to administer the license server are described in Chapter 3, “Abaqus licensing.”

The environment file provides options for customizing an Abaqus installation. Specifying the file system to be used to store scratch files is particularly important. For details on how to set local defaults and adjust the system configuration to run Abaqus jobs efficiently, see Chapter 4, “Customizing the Abaqus environment.”

For more information on customizing the Abaqus user interface, printer configuration, and graphic card tuning, see Chapter 5, “Customizing Abaqus/CAE and Abaqus/Viewer.”

Additional information is provided in the following appendices:

- Appendix A, “System requirements”
- Appendix B, “Abaqus release directories and files”
- Appendix C, “Configuring Abaqus documentation”
- Appendix D, “Accessing remote file systems for installation and execution”
- Appendix E, “Verification procedure”
- Appendix F, “Troubleshooting Abaqus network licensing”

2. Installing Abaqus

This chapter describes the installation procedures for Abaqus documentation, licensing, and products, as well as the procedures to uninstall these components. An Abaqus installation requires two DVDs: one for documentation and one for licensing and products.

Before starting the Abaqus installation procedures, you should confirm that your computer is compatible with this release of Abaqus. You can do this by checking the software requirements specified in Appendix A, “System requirements,” or by using the system information utility available on the licensing and products DVD. The system information utility compares your system configuration to the published system requirements. Use the following commands or option to run the script:

UNIX and Linux platforms

```
/dvd_dir/setup -sysinfo
```

where *dvd_dir* is the directory used when the DVD was mounted or where the installation files reside.

Windows platforms

- From the startup screen, select **Launch the Abaqus System Information Utility**.
- From the command line, type

```
dvd_drive:\setup.exe -sysinfo
```

where *dvd_drive* is the DVD-ROM drive.

The latest information on system requirements is available on the **Support** page at www.simulia.com. For information on installation procedures and requirements specific to Microsoft Vista, refer to the Answers in the SIMULIA Online Support System (SOSS), which is accessible from the **My Support** page at www.simulia.com.

The disk space used for installation will be charged against the account of the owner of the Abaqus installation directory. Administrator privileges are required on Windows platforms for all installation activities.

A complete Abaqus installation consists of three procedures. **You should perform the installation procedures in the following order:**

1. Abaqus documentation.
2. Abaqus licensing.
3. Abaqus products.

Abaqus documentation should be installed on at least one computer on your network. Abaqus documentation should be installed before licensing and products because the product installation procedure uses the documentation URL to configure network access to the online documentation.

You must have an Abaqus 6.10 license file to install Abaqus licensing and start the license server. A new license file is required for each new general release, and this license file allows you to run

subsequent Abaqus 6.10 maintenance deliveries and previous releases dating back to Abaqus 6.2-7. For example, a new license file is required for Abaqus 6.10-1, and this license file can be used to run all Abaqus 6.10 maintenance deliveries (Abaqus 6.10-2, Abaqus 6.10-3, etc.) and any earlier releases of Abaqus (Abaqus 6.9, Abaqus 6.8, etc.) still installed on your network. To request an Abaqus license file, contact your local sales office. For the definitions of a general release and a maintenance delivery, see the Abaqus Glossary.

Abaqus licensing should be installed on only one computer unless you are using a redundant license server configuration. The server computers that you choose for documentation and licensing should be accessible by all computers on the network where you plan to run Abaqus and should be stable machines that are not frequently rebooted or shut down.

Before installing the Abaqus products, you must have an Abaqus network license server that is compatible with Abaqus 6.10 installed and running on your network. Abaqus 6.10 licensing is compatible with FLEXnet Licensing Version 10.8.7 or higher.

During the installation procedures, default values are supplied whenever possible; it is highly recommended that you use the defaults unless you have good reason to change them. The SIMULIA directory structure is designed to facilitate multiple releases of Abaqus. The Abaqus products, Abaqus licensing, and Abaqus documentation should be installed in the same SIMULIA parent directory to take full advantage of this directory structure. For an illustration of the directory structure, see Appendix B, “Abaqus release directories and files.”

2.1 Installation procedures

This section summarizes the commands used to perform an Abaqus installation on a computer that has a locally mounted DVD. If you are installing Abaqus on a computer using a remotely mounted DVD, see “Accessing the installation files from a remotely mounted DVD,” Section D.1. For a more detailed discussion of each installation procedure, see the following sections:

- “Abaqus documentation installation details,” Section 2.2
- “Abaqus licensing installation details,” Section 2.3
- “Abaqus product installation details,” Section 2.4

If you are performing an installation on a computer with previous releases of Abaqus installed on it, Abaqus 6.10 is installed using the same directory structure as the previous installations (see Appendix B, “Abaqus release directories and files,” for details). If you are installing Abaqus for the first time, create a parent directory structure for Abaqus. The recommended path for this directory structure is **SIMULIA/Abaqus**. Throughout this manual, the **SIMULIA** directory is referred to as *simulia_dir* in directory path listings; Abaqus documentation and licensing are installed in *simulia_dir*. The **SIMULIA/Abaqus** directory is referred to as *abaqus_dir* in directory path listings; the Abaqus products are installed in *abaqus_dir*.

Insert the appropriate DVD in the drive. On Windows platforms a startup screen is provided from which you can launch the commands used to perform an Abaqus installation, the command to run the system information utility, and other installation-related tasks.

If the DVD does not mount automatically, issue the DVD mount command for the platform on which you are installing Abaqus.

2.1.1 Documentation installation procedure

The Abaqus online documentation is provided in two formats, HTML and PDF. There are several options for installing the documentation and making it available to your Abaqus users:

- Install HTML and PDF on a server on your network, and use web server software to serve both formats to users (this is the most common choice).
- Install multiple copies of the PDF files on individual users' machines, where they can view it locally using a PDF viewer such as Adobe Acrobat Reader.
- Skip the documentation installer, and simply copy the PDF files from the DVD to any computers or disks.

You must have write permission in the directory where documentation is installed. Administrator privileges are required to run the installer on Windows platforms.

The commands and options used to install documentation are described below. For more details about the documentation installation options, see “Abaqus documentation installation details,” Section 2.2.

To install Abaqus documentation:

1. From the SIMULIA parent directory, execute the following command or use the following option:

UNIX and Linux platforms

```
/doc_dir/setup [-replay full_path_to_replay_file]
```

where *doc_dir* is the directory used when the Documentation DVD was mounted or where the documentation installation files reside.

Windows platforms

- From the startup screen, select **Install Abaqus Documentation**.
- From the command line, type

```
doc_drive:\setup.exe [-replay full_path_to_replay_file]
```

where *doc_drive* is the DVD-ROM drive that contains the Documentation DVD.

Command option

-replay *full_path_to_replay_file*

Use this option to perform additional identical documentation installations without having to reenter the same information for each installation. For more information, see “Repeating documentation installations on multiple computers,” Section 2.2.2.

2. Follow the instructions in the dialog boxes that appear to complete the installation. Choose which formats of documentation to install:
 - **HTML and PDF**
 - **PDF only**
3. If you installed the HTML format and selected the “Abaqus web server” or “no web server” option, you are provided with a URL that points to your HTML documentation when the installer completes. This URL is also written to a log file in the following location:

simulia_dir/Documentation/installation_info/v6.10/info.log

If you selected the “existing non-Abaqus web server” option, a sample URL is written to the **info.log** file. To access the HTML documentation from the Abaqus products, you must provide a valid URL during each subsequent Abaqus product installation.

4. If you installed only the PDF documentation, the path to the **index.pdf** file is shown.

To copy the PDF documentation from the DVD (without running the installer)

1. Find the directory **pdf_docs/** on the documentation DVD. Copy the entire contents of this directory.
2. To view the documentation, open the file **index.pdf** in a PDF viewer or web browser.

You may copy the PDF documentation in accordance with the following terms:

- Use of this material is governed by your Abaqus software license agreement.
- Keep the documentation confidential and utilize your best efforts to prevent unauthorized disclosure or use of any Abaqus Program including the Abaqus documentation for such program.
- Treat all Abaqus Programs including the Abaqus documentation with the same degree of care that you treat like information of your own that you do not want publicly disclosed or the subject of unauthorized access or use.
- Do not make or permit to be made any more copies of any Abaqus Program including the Abaqus documentation than are authorized by your Abaqus software license agreement.

2.1.2 Licensing installation procedure

You must have an Abaqus 6.10 license file to set up an Abaqus license server. Abaqus 6.10 licensing uses FLEXnet Licensing Version 10.8.7. Select the licensing installation you want to perform from the following scenarios:

- Automatic configuration is recommended for most users, particularly those whose primary responsibility is not computer system or network administration. Use the installation procedure to install, configure, and start the Abaqus license server for you automatically. An existing license server using a version of FLEXnet Licensing prior to Version 10.8.7 is upgraded automatically during the installation procedure if it was previously installed and configured using the Abaqus licensing installation procedure. For redundant license servers the installation will install and configure the Abaqus license servers automatically, but you will have to manually start each server. Use this method for ease of maintenance so that future upgrades can be configured automatically.
- Manual configuration is required for combining license files with other vendors. At the beginning of the installation procedure, select **Just install the licensing utilities**. You will need to configure and start the Abaqus license server manually. Use this method to allow the system administrator more control.

Abaqus 6.10 requires **lmgrd** Version 10.8.7 or higher. If you integrate your Abaqus license features with other products using a FLEXnet Licensing combined license file, ensure that your version of **lmgrd** meets this requirement. For more information, refer to the FLEXnet Licensing End User Guide Version 10.8, which is available from the **Licensing** section of the **Support** page at www.simulia.com.

The commands and options used to install the Abaqus license server and to stop and remove a license server are described below. For a more complete discussion of the licensing installation options, see “Abaqus licensing installation details,” Section 2.3. To replace an existing or expired license file, see “Updating an Abaqus license file,” Section 3.2; for details on installing a redundant license server, see “Redundant license server configurations,” Section 3.4. For information on installation procedures and requirements specific to Microsoft Vista, refer to the SIMULIA Online Support System (SOSS), which is accessible from the **My Support** page at www.simulia.com.

To install Abaqus 6.10 licensing:

1. From the SIMULIA parent directory, execute the following command or use the following option:

UNIX and Linux platforms

```
/prod_dir/setup
```

where *prod_dir* is the directory used when the Licensing & Products DVD was mounted or where the licensing installation files reside.

Windows platforms

- From the startup screen, select **Install Abaqus Product and Licensing**.
- From the command line, type

```
prod_drive:\setup.exe
```

where *prod_drive* is the DVD-ROM drive that contains the Licensing & Products DVD.

2. In the **Installation type** panel, choose **License**.
3. If you have an existing Abaqus license server on this system, the installer attempts to stop the license server; if unable to do so, it provides the option to install only the licensing administration tools. If you install only the licensing administration tools, you must manually configure the license server. If the installer is unable to stop the existing license server and you want to complete a full automatic license server installation, you need to stop the existing server using the procedure described below.
4. Follow the instructions in the dialog boxes that appear to complete the licensing installation.
5. On UNIX and Linux platforms, the Abaqus license server does not restart automatically after a system reboot by default. If you are installing a new Abaqus license server, you must add the Abaqus licensing command to your system startup script if you wish to have the Abaqus license server restart automatically. The command that was used to start the server is written to

```
simulia_dir/License/installation_info/license_startup.txt
```

On Windows platforms, an automatically configured license server starts automatically upon reboot.

Stopping and removing an existing license server

If you have an existing license server from a previous release of Abaqus running on your system, you will need to remove it before installing the new license server. The license installation procedure will attempt to remove the existing server automatically. If it is unable to do so, you should remove the existing server using one of the following procedures:

UNIX and Linux platforms

1. Enter the command

```
simulia_dir/License/lmdown -c [port]@license_server_host  
-vendor ABAQUSLM
```

where *license_server_host* is the hostname of the license server. If the license server is using a port that is not between 27000 and 27009, you must specify the port number as well.

2. Verify that the server has been stopped by running the command

```
simulia_dir/License/lmstat -c [port]@license_server_host
```

Verify that no **ABAQUSLM** vendor daemons are reported. An error status message may appear if the Abaqus license server was successfully stopped and there are no other vendor daemons running on the specified host.

Windows platforms

In addition to terminating the license server, you must remove licensing as a Windows service.

1. From the **Start** menu, select **Programs**→**Abaqus release**→**Licensing**, where *release* refers to the Abaqus release, to display the **LMTTOOLS** dialog box.
2. Verify **Configuration using Services** is enabled on the **Service/License File** tabbed page.
3. Select the **Start/Stop/Reread** tab, select the appropriate Abaqus 6.x license server, where 6.x refers to the Abaqus release, and click **Stop Server**.
4. Select the **Config Services** tab, select the appropriate Abaqus 6.x license server, and click **Remove Service**.
5. Select **File**→**Exit** to close the **LMTTOOLS** dialog box.

2.1.3 Abaqus product installation procedure

To install the Abaqus products, you must have an Abaqus network license server that is compatible with Abaqus 6.10 installed and running on your network. It is recommended that you perform the documentation installation before installing Abaqus products to obtain the URL for the HTML documentation (or the location of the PDF files) and to enable access to the documentation from Abaqus. You will be asked to provide the following information relating to the documentation and licensing installations:

- The hostname of your license server(s) (REQUIRED).
- One of the following (depending on which formats of documentation were installed):
 - The URL for the Abaqus HTML documentation (optional, but recommended). If you did not install the Abaqus HTML documentation before the products or if you do not specify the URL, you must perform additional steps to enable access to the HTML documentation from the Abaqus products (see “Setting the documentation URL after installation,” Section C.2).
 - The path to the directory where the PDF documentation files are installed (if only the PDF, not HTML, was installed).

The commands and options used to install the Abaqus products are described below. For a more complete discussion of the Abaqus product installation options, see “Abaqus product installation details,” Section 2.4.

To install the Abaqus products:

1. From the Abaqus parent directory, execute the following command or use the following option:

UNIX and Linux platforms

```
/prod_dir/setup [-jre system|path_to_jre] [-crossplatform]
[-replay full_path_to_replay_file] [-sysinfo] [-nosystemchecks]
[-help]
```

where *prod_dir* is the directory used when the Licensing & Products DVD was mounted or where the product installation files reside.

Windows platforms

- From the startup screen, select **Launch the Abaqus System Information Utility** or select **Install Abaqus Product and Licensing**.
- From the command line, type

```
prod_drive:\setup.exe [-replay full_path_to_replay_file] [-sysinfo]
[-nosystemchecks] [-help]
```

where *prod_drive* is the DVD-ROM drive that contains the Licensing & Products DVD.

Command options

-jre system|path_to_jre

Specify the Java Runtime Environment (JRE) to use for installation. This option is not normally required as JREs for most supported platforms are included with the installation disk and used by default. Specify the value **system** to use the system JRE, if installed; or specify the full path to a user-defined JRE. To use a system JRE, the **java** command must be in your PATH environment variable and the JRE must be Version 1.5 or later.

-crossplatform

Install the Abaqus products for various platforms on a single file server using a single system to perform all of the installations. You can install Abaqus for only one platform at a time. When you are installing for a platform that is not the local platform, pre-installation system checks and post-installation verification are not performed. The verification procedure should be run manually once the user is logged on to the local computer (see Appendix E, “Verification procedure”). The **-crossplatform** option can be used on platforms listed on the installation media.

-replay full_path_to_replay_file

Use this option to perform additional identical installations without having to reenter the same information for each installation. For more information, see “Repeating product installations on multiple computers,” Section 2.4.3.

-sysinfo

This option lists information about your system's software and hardware resources and compares these resources to the requirements for running Abaqus. Abaqus is not installed if this option is used, and all other command options are ignored.

-nosystemchecks

Skip pre-installation system checks. By default, the installation procedure runs checks before performing an installation to confirm that your system meets the Abaqus minimum system requirements. If any requirements affecting the Abaqus products are not met, the installation terminates. Use this option to bypass the system checks and prevent termination of the installation procedure.

-help

Display **setup** options.

2. In the **Installation type** panel, choose **Product**.
3. Follow the instructions in the dialog boxes that appear to complete the product installation.

2.2 Abaqus documentation installation details

The Abaqus online documentation is provided in two formats, HTML and PDF. Most installations share the HTML and PDF documentation from a single server. The PDF documentation can also be installed on as many computers as desired, for local access by individual users.

The commands and options used to install Abaqus documentation are listed in “Documentation installation procedure,” Section 2.1.1.

Abaqus HTML documentation can be installed on platforms that are supported for Abaqus 6.10 and should generally be installed on only one computer on your network. The server machine that you choose for the HTML and PDF documentation should be accessible by all computers where you plan to run Abaqus.

The disk space requirement for the Abaqus HTML plus PDF documentation is approximately 2.0 GB. Abaqus HTML documentation can be viewed with the following web browsers:

- Firefox 2 and 3 on Linux and Windows platforms
- Internet Explorer 6, 7, and 8 on Windows platforms only

Later versions of these web browsers are expected to offer acceptable performance.

The Abaqus PDF documentation is not meant as a replacement for the HTML-format documentation. However, it has the following advantages:


- Requires less disk space.
- Convenient portable reference.
- Ability to print a range of pages, such as a section or chapter.

- Higher quality printed output than that available from the HTML documentation.
- Searchable alternative to the HTML documentation, particularly for situations where the HTML documentation is not searchable because it was installed with no web server.

Installing only the PDF documentation has the following disadvantages:

- The detailed, step-by-step instructions for using each of the Abaqus/CAE functions are available only in the HTML version of the Abaqus/CAE User's Manual and as context-sensitive help. They do not appear in the PDF version of the manual.
- The hyperlinks in context-sensitive help will not function if you have not installed the Abaqus HTML documentation.
- Some of the items under the **Help** menu in Abaqus/CAE will connect only to the HTML help, not to the PDF documentation.

The Abaqus PDF collection contains all the books in the HTML collection. The PDF version of all manuals except the Abaqus Scripting Reference Manual and the Abaqus GUI Toolkit Reference

Manual includes a  icon in the top left corner of each page that allows you to print a selected section without specifying a page range. More information about navigating and printing the Abaqus PDF documentation is available in Chapter 5, "Overview of the Abaqus PDF documentation," of Using Abaqus Online Documentation.

2.2.1 Configuration of documentation installation

When the documentation installer runs, it determines the configuration of any existing Abaqus documentation:

- If you have documentation from a previous release of Abaqus installed (Abaqus 6.9, for example), Abaqus 6.10 documentation is installed in the same location as the documentation from the previous release. A directory named **v6.10** is created to hold the Abaqus 6.10 documentation (see Appendix B, "Abaqus release directories and files," for information on the directory structure). If you are using the Abaqus web server software, you are prompted to choose whether or not to serve all releases of the documentation simultaneously.
- If documentation from a previous release of Abaqus is not installed, you are prompted for the installation directory. This should be the same as your SIMULIA parent directory (referred to as *simulia_dir*). The default location is the current working directory on UNIX and Linux platforms and **C:\SIMULIA** on Windows platforms. You should *not* install Abaqus documentation in a directory that indicates a specific Abaqus release number. The installation procedure creates a subdirectory named **Documentation** under the specified directory.

During an HTML installation, you must select one of the following options:

- Install and start the Abaqus web server software (recommended option, to allow search functionality).

- Install the HTML documentation and use an existing non-Abaqus web server. In this case you must configure your web server software manually to access the Abaqus HTML documentation (see “Configuring an existing non-Abaqus web server,” Section C.1.1).
- Install the HTML documentation with no web server. In this case no search functionality is available in the HTML documentation. Select this option if you want to view the HTML documentation from your local system without executing an active web server.

With the exception of “no web server,” the installation option that you select applies to both the Abaqus 6.10 HTML documentation and, if it exists, any previous release of Abaqus documentation. If you select the “no web server” option for the Abaqus 6.10 installation, the Abaqus 6.10 HTML documentation has no web server and the documentation from previous releases is not affected.

Note: On Windows platforms the Abaqus web server software will start up automatically after a reboot. On UNIX and Linux platforms the Abaqus web server is started by the installation procedure, but you should include the script to restart the web server (*simulia_dir/Documentation/installation_info/v6.10/startServer*) in the system startup file for the server host machine to have the Abaqus web server restart automatically after a system reboot. Contact your system administrator or platform vendor for information on the system startup file for your computer.

Log files are written to the *simulia_dir/Documentation/installation_info/v6.10* directory when the installation is complete. The log files are helpful for troubleshooting problems, and they should not be removed. For more information on accessing and using the Abaqus HTML documentation, see “Abaqus HTML documentation,” Section 3.2.7 of the Abaqus Analysis User’s Manual, and the online book Using Abaqus Online Documentation.

2.2.2 Repeating documentation installations on multiple computers

When you install documentation, a file called **installer.properties** is created in the *simulia_dir/Documentation/installation_info/v6.10* directory. This file, called the replay file, contains information that was used during the documentation installation, such as the installation directory and web server configuration. The **-replay** command option is available to repeat a previous installation on identical platforms without reentering the same information for each installation. You can use this option to perform a silent installation (no graphical user interface is displayed). It performs an installation identical to the previous installation (same installation directory, same configuration, etc.) using information stored in the replay file.

The information that was entered for the first installation must be valid for subsequent installations that use the **-replay** option. All installation error checking is skipped when this option is used. You should make sure that the system meets the requirements as specified in Appendix A, “System requirements,” and that you have write permissions and sufficient disk space in the installation directory before performing the installation. To use the **-replay** option, do the following:

INSTALLING Abaqus

1. Install documentation on a computer using the graphical user interface (without the **-replay** option). You can install either option: HTML and PDF, or PDF-only.
2. Copy the file **installer.properties** (the replay file) to the other computers on which you want to install Abaqus documentation (must be the same platform) or to a mounted network drive.
3. Install the documentation on another computer using the **-replay** option as follows:

UNIX and Linux platforms

```
/doc_dir/setup -replay full_path_to_replay_file
```

Windows platforms

```
doc_drive:\setup.exe -replay full_path_to_replay_file
```

The documentation is installed silently.

4. Review the installation log files for errors:
 - **html_InstallLog.log** and **info.log** (for an installation of HTML and PDF)
 - **pdf_InstallLog.log** (for a PDF-only installation)

These files are located in the following directory:

```
simulia_dir/Documentation/installation_info/v6.10
```

This step is optional but recommended.

5. Repeat the documentation installation using the **-replay** option on each of the remaining computers.

2.3 Abaqus licensing installation details

Abaqus 6.10 licensing uses FLEXnet Licensing Version 10.8.7. The licensing installation procedure can be used for the following scenarios:

- to install the FLEXnet Licensing administration tools and to configure and start the Abaqus 6.10 license server automatically; or
- to install only the FLEXnet Licensing administration tools.

An existing license server from a previous version of FLEXnet Licensing is upgraded automatically during the installation procedure if it was previously installed and configured using the Abaqus licensing installation procedure. If the installation procedure detects an existing license server on your machine from a version of FLEXnet Licensing higher than Version 10.8.7, you have the option of either overwriting the existing server or terminating the installation procedure. To update your license file without overwriting the existing license server, see the procedure in “Updating an Abaqus license file,” Section 3.2.

The commands and options used to install the Abaqus license server or to stop and remove an existing license server are listed in “Licensing installation procedure,” Section 2.1.2.

For redundant license server configurations, you can install the license administration tools and configure the license server, but you will need to start the license servers manually (see “Redundant license server configurations,” Section 3.4). If you choose to install only the license administration tools, you must start the license server manually using the FLEXnet license server manager **lmgrd** (see “FLEXnet license server manager lmgrd,” Section 3.5).

If you are installing an Abaqus 6.10 license server, you must have an Abaqus 6.10 license file saved on the computer on which you are installing the license server. The license file is the file containing the Abaqus 6.10 license that was attached to the e-mail message sent by SIMULIA (identified by “Subject: LK: *site ID* Abaqus 6.10 License File,” where *site ID* is the customer number that was assigned to your site) or by your local office or representative.

Abaqus licensing needs to be installed on only one computer on your network unless you are using a redundant license server configuration. The disk space requirement for the Abaqus licensing utilities ranges from 10–25 MB depending upon the platform. The computer that you choose for licensing should be accessible by all computers where you plan to run Abaqus and should be a stable machine that is not frequently rebooted or shut down. For additional information on licensing, see Chapter 3, “Abaqus licensing.”

When the installation procedure is executed, a browse function is provided for prompts where you are requested to specify a directory or file location. You will be prompted for the following information:

- The location of the license file. The installation procedure will save the license file in the *simulia_dir/License* directory using the default name **abaquslm.lic**.
- The installation directory. Specify the SIMULIA parent directory (*simulia_dir*), which is the location where the licensing directory will be stored. The default location for the installation directory is the current working directory on UNIX and Linux platforms and **C:\SIMULIA** on Windows platforms.

Note: On Windows platforms the license server is started by the installation procedure and will start up automatically upon reboot. On UNIX and Linux platforms the license server is started by the installation procedure, but you should include the command to restart the license server in the system startup file for the license server host computer to have the Abaqus license server restart automatically after a system reboot. The command to restart the license server is written to the **license_startup.csh** file in the *simulia_dir/License/installation_info* directory. Contact your system administrator or platform vendor for information on the system startup file for your computer.

Log files are written to the *simulia_dir/License/installation_info* directory when the installation is complete. The log files are helpful for troubleshooting problems, and they should not be removed.

After the Abaqus documentation is installed and the license server is installed and running, you should proceed to the Abaqus product installation on each computer that you will use to run the products.

You need not install the Abaqus products on the license server host unless you will run Abaqus on that computer.

2.4 Abaqus product installation details

It is recommended that you install Abaqus documentation before you install the Abaqus products; if you do not, you must perform additional steps to provide access to the HTML documentation from Abaqus.

To install the Abaqus products, an Abaqus network license server that is compatible with Abaqus 6.10 must be installed and running; you must know the name of at least one active Abaqus license server to perform the product installation.

All Abaqus products and sample problems are installed as part of the Abaqus product installation. Some Abaqus environment settings are defined appropriately for the parameters defined in your Abaqus license. Verification of a subset of licensed products is performed automatically during the installation.

The commands and options used to install the Abaqus products are listed in “Abaqus product installation procedure,” Section 2.1.3.

You must have write permission in the Abaqus parent directory to install the Abaqus products. On Windows platforms Abaqus shortcuts and required environment variables, including **PATH**, are added to the **All User** profile.

2.4.1 Visual C++ and MPI Libraries

When you select **Install Abaqus Product & Licensing** from the startup screen on Windows platforms, the Microsoft Visual C++ 2005 SP1 and 2008 SP1 runtime libraries are installed automatically along with Abaqus. These libraries are required to run Abaqus. On Windows/x86-32 platforms, the 32-bit 2005 and 2008 runtime libraries are required and installed. On Windows/x86-64 platforms, the 32-bit 2005 and 2008 libraries and 64-bit 2008 libraries are required and installed.

Installation of the Microsoft Visual C++ 2005 and 2008 runtime libraries can instead be performed independently of the Abaqus product installation if necessary, by selecting **Install Microsoft Visual C++ 2005 SP1 and 2008 SP1 Runtime Libraries** from the startup screen. Alternatively, you can execute the following files on the Abaqus Licensing & Products DVD:

`win86_32/2008_SP1_vccredist_x86.exe` (32-bit libraries)

`win86_32/2005_SP1_vccredist_x86.exe` (32-bit libraries)

`win86_64/2008_SP1_vccredist_x86.exe` (32-bit libraries)

`win86_64/2005_SP1_vccredist_x86.exe` (32-bit libraries)

`win86_64/2008_SP1_vccredist_x64.exe` (64-bit libraries)

Message Passing Interface (MPI) components must be installed to use MPI-based parallel execution in Abaqus/Standard, to use domain-level parallelization in Abaqus/Explicit, or to run any job

in Abaqus/CFD (regardless of the number of CPUs). If your Abaqus users will be running these types of simulations, you must have the required MPI components preinstalled or allow the Abaqus installer to install them for you:

- On Windows/x86-64 platforms, the Abaqus product installer automatically installs the Microsoft MPI libraries. If desired, you can install these libraries separately by selecting **Install Microsoft HPC MPI Redistributable Pack** from the startup screen. You can also launch the Microsoft MPI installer separately by executing the following file on the Abaqus Licensing & Products DVD:

```
win86_64/msmpi_x64.msi
```

(If you have Windows HPC Server 2008, however, this is not needed because the MPI components are included with the operating system.)

- On Windows/x86-32 platforms, the Abaqus product installer optionally launches the installer for the Hewlett-Packard HP Message Passing Interface library (HP-MPI). If you choose this option, the HP-MPI installer runs as a separate process, and you must click through all panels to completion before continuing with the Abaqus product installer. Windows Administrator privileges are required to run the HP-MPI installer.

During the HP-MPI installation, it is recommended that you make the following choices to ensure compatibility with Abaqus:

- Accept the default installation path in the **Select Destination Location** panel.
- Select **HP-MPI Libraries, tools, and services for Windows 2003/XP** in the **Select Components** panel.
- Uncheck **Register HP-MPI Service with system (reboot on restart)** in the **Select Additional Tasks** panel.

If desired, you can install the Hewlett-Packard MPI libraries separately by selecting **Install HP-MPI** from the startup screen. You can also launch the Hewlett-Packard MPI installer separately by executing the following file on the Abaqus Licensing & Products DVD:

```
win86_32/HP-MPI-V1.1_x86.exe
```

The HP-MPI installer does not run automatically during a silent installation with the **-replay** option on Windows/x86-32 platforms (because no GUI can be displayed). See “Repeating product installations on multiple computers,” Section 2.4.3 below.

- On Linux 64-bit and HP-UX UNIX operating systems, the Abaqus product installer automatically installs the Hewlett-Packard HP Message Passing Interface library (HP-MPI).

During the Abaqus product installation on the Linux 64-bit and HP-UX platforms, you will be prompted to configure the MPI-based parallel execution mode by choosing one of the following:

- **Configure MPI now.** To simplify usage of the parallel features in Abaqus, it is recommended that you use MPI-based parallel processing if it is supported on your platform. You will need to provide the full path to the appropriate MPI launcher.

- **Configure MPI later.** Abaqus will use the default MPI settings. You can change the MPI settings later by editing the environment file located in the *abaqus_dir/release/site* directory. For more information, see “Using the Abaqus environment settings,” Section 3.3.1 of the Abaqus Analysis User’s Manual.
- **Configure THREADS now.** The thread-based parallel processing mode is used.

Supported MPIs are listed in the **System Information** section of the **Support** page at www.simulia.com. For more information on MPI and thread-based parallel processing modes, see “Parallel execution,” Section 3.5 of the Abaqus Analysis User’s Manual.

Any other third-party software that will be used in conjunction with the Abaqus products (compilers, operating system patches, etc.) should be installed before installing the Abaqus products.

2.4.2 Specifying information for product installation

When the installation procedure is executed, a browse function is provided for prompts where you are requested to specify a directory or file location. You will be prompted for the following information:

- The network license server host name. The installation procedure will attempt to determine this information automatically, but you have the option of changing the default choice. Make sure the appropriate license server host is entered. If a set of redundant servers should be used, enter the host names of the redundant servers in the boxes provided in the dialog box (see “Redundant license server configurations,” Section 3.4, for details on installing redundant license servers). The installation will define settings in the Abaqus environment file according to information detected in the Abaqus license file on the server. If the license server contains an academic teaching license, Abaqus will be configured to use this license by default (see “License management parameters,” Section 4.1.6).
- The URL for the Abaqus HTML documentation or directory where the PDF documentation is located. (This step is optional, but recommended.) If you did not install the Abaqus HTML documentation before the products or if you do not specify the URL, you must perform additional steps to provide access to the HTML documentation (see “Setting the documentation URL after installation,” Section C.2).
- The installation directory. Specify the Abaqus parent directory (referred to as *abaqus_dir*), which is the location where all installed Abaqus releases will be stored. The default location is the current working directory on UNIX and Linux platforms and **C:\SIMULIA\Abaqus** on Windows platforms. You should NOT install Abaqus in a directory that indicates a specific Abaqus release number; the release-specific directory will be created by the installation procedure. For example, all files needed to run Abaqus 6.10-1 will be installed in *abaqus_dir/6.10-1*.
- The location of the Abaqus start-in directory (Windows platforms only). Specify the location where user files created during Abaqus interactive sessions are stored and the home directory for the Abaqus command window. The default location is **C:\temp**. (This can be changed by users

later; see “Common customizations on Windows platforms,” Section 5.1.2.) Ensure that the start-in directory has write permission for all users.

- If the installation program detects a wide area network (WAN) license file, you will be prompted for information about the geographic location of your computer. This information is used to automatically set the **computer_location** environment file parameter and enable license usage logging for report generation. For more information, see “Reporting license usage data,” Section 3.8.

When the installation is complete, a record of the installation and a listing of the licensed products is written to a file called **Abaqus_release_InstallLog.log** in the *abaqus_dir/release/installation_info/* directory. This file is helpful for troubleshooting problems, and it should not be removed.

2.4.3 Repeating product installations on multiple computers

When you install the Abaqus products on a computer, the installation creates a file called **installer.properties** in the *abaqus_dir/release/installation_info* directory. This file, called the replay file, contains information that was used during the product installation, such as the installation directory, license server host, and documentation URL. The **-replay** command option is available to repeat a previous installation on identical platforms without reentering the same information for each installation. You can use this option to perform a silent installation (no graphical user interface is displayed). It performs an installation identical to the previous installation (same installation directory, same license server host, etc.) using the information stored in the replay file.

The information that was entered for the first installation must be valid for subsequent installations that use the **-replay** option. All installation error checking is skipped when this option is used. You should make sure that the system meets the requirements as specified in Appendix A, “System requirements,” and that you have write permissions and sufficient disk space in the installation directory before performing the installation. To use the **-replay** option, do the following:

1. Install the Abaqus products on a computer using the graphical user interface (without the **-replay** option).
2. Copy the file **installer.properties** (the replay file) to the other computers on which you want to install the Abaqus products (must be the same platform) or to a mounted network drive.
3. Install the Abaqus products on another computer using the **-replay** option as follows:

UNIX and Linux platforms

```
/prod_dir/setup -replay full_path_to_replay_file
```

Windows platforms

```
prod_drive:\setup.exe -replay full_path_to_replay_file
```

The Abaqus products are installed silently, and pre-installation system checks and post-installation verification are bypassed.

4. Run the verification procedure manually to check the installation of the Abaqus products (see Appendix E, “Verification procedure”). This step is optional, but strongly recommended.
5. Repeat the product installation using the **-replay** option on each of the remaining computers.

2.4.4 Client installation for Windows

The client product installation allows you to configure a computer on a Windows platform to run the Abaqus products that have been installed on a mounted network drive. Use this installation to configure multiple computers to run Abaqus from a central server location without installing Abaqus on each computer. The client computer must be able to access the directory containing the Abaqus installation on the network and the Abaqus license server. The shortcuts and uninstaller that are placed on the client computer require about 50 MB of disk space.

When you install the Abaqus products on a computer, the installation creates a file called **installer.properties** in the *abaqus_dir/release/installation_info* directory. The client installation utilizes this properties file to perform a silent installation. No graphical user interface is displayed, and all error checking is skipped during the installation on the client computer. To perform a client installation, do the following:

1. Install the Abaqus products on a mounted network drive.
2. On the client computer, open a command prompt and execute the following command:

```
Network_drive:\path_to_installation\installation_info\client_setup.exe
```

The client computer is configured to run Abaqus from the network drive. Pre-installation checks and post-installation verification are bypassed.

3. Run the verification procedure manually to check the installation of the Abaqus products (see Appendix E, “Verification procedure”). This step is optional, but strongly recommended.
4. Repeat Steps 2 and 3 on each of the remaining client computers.

An installation log and an uninstaller are installed on the client computer in the directory **C:\Documents and Settings\user\Abaqus release**, where *user* refers to your user name and *release* refers to the release of Abaqus you are installing. The uninstaller is silent; no graphical user interface or dialog boxes are shown.

This installation procedure does not install either the Microsoft Visual C++ or Message Passing Interface (MPI) libraries on the client computer. (See “Visual C++ and MPI Libraries,” Section 2.4.1.)

2.4.5 Using Abaqus commands

The installation procedure creates a unique command for each release of Abaqus. This command is written in the **Commands** directory below the Abaqus parent directory (*abaqus_dir*). For example, the command to run Abaqus 6.10-1 is **abq6101**. The command to run subsequent releases of Abaqus 6.10 is **abq610n**, where *n* is the maintenance delivery. The installation procedure also creates a link called **abaqus** to the command for the latest release of Abaqus installed. For a list of all Abaqus command line options, use the **help** command line option with the appropriate Abaqus execution command. For example, to see the command line options for Abaqus 6.10-1, type **abq6101 help**. For a detailed description of Abaqus command line options, see Chapter 3, “Job Execution,” of the Abaqus Analysis User’s Manual.

On Windows platforms the **Commands** directory path is added automatically to all users’ paths during installation. On UNIX and Linux platforms the **Commands** directory should be added to each user’s path manually. Alternately, a link for each Abaqus release can be defined at the system level or for each user, pointing to the appropriate command in the **Commands** directory.

2.5 Uninstalling Abaqus licensing or products

This section describes how to remove Abaqus licensing or product releases from your computer.

2.5.1 UNIX and Linux platforms

The following procedures explain how to remove Abaqus licensing or product releases from UNIX and Linux platforms.

To remove Abaqus licensing:

This procedure must be performed on the license server host, and you must be logged in as root or the owner of the license server.

1. Before you can remove the license server and license files, you must terminate the license server using the FLEXnet Licensing tool **lmdown** (see “lmdown,” Section 3.6.2).
2. Create a backup of any files you wish to keep, such as your Abaqus license file (typically named **abaquslm.lic**) and the FLEXnet Licensing server debug log file (typically named **abaquslm.log**).
3. Remove the files using the following procedure:

```
cd simulia_dir
rm -rf License
```

4. Remove the license server restart command from the computer's system startup file. Contact your system administrator or platform vendor for information on the system startup file for your computer.

To remove the Abaqus products:

You can remove a release of Abaqus by deleting the release-specific directory and all subordinate files. Your user account must have permission to delete the Abaqus release-specific directory.

1. Type the command

```
cd abaqus_dir
```

where *abaqus_dir* is the Abaqus parent directory.

2. Change the permissions of the directories and files by entering the command

```
chmod -R 755 6.10-n
```

where *n* refers to the update you wish to delete.

3. Remove the release or maintenance delivery by typing the command

```
rm -rf 6.10-n
```

where *n* refers to the update you wish to delete.

4. Links to the uninstalled Abaqus release will no longer be valid. You can remove an invalid link from the **Commands** directory by typing the command

```
rm link
```

where *link* refers to an invalid link name.

Alternatively, you can remove the entire **Commands** directory from the Abaqus parent directory (*abaqus_dir*) by typing the command

```
rm -rf Commands
```

2.5.2 Windows platforms

You can use the Windows uninstall shortcuts provided, the **Add or Remove Programs** tool, or the Abaqus silent uninstaller batch files to remove Abaqus licensing or products. The silent uninstallers allow you to automate these tasks from another batch/script file. The silent uninstallers are available only on Windows platforms. You must have Windows Administrator privileges to uninstall Abaqus components.

Both the Windows uninstall shortcuts and the silent uninstaller batch files generate a log file that you can review. The log file indicates whether the uninstall was successful and if you should reboot/restart your computer. This information is shown in the last two lines of the log file. For example,

Uninstall Status: SUCCESS
Restart Needed: YES_RECOMMENDED

The **Uninstall Status** line will indicate either **SUCCESS** or **INCOMPLETE**. The **Restart Needed** line will indicate either **YES_RECOMMENDED**, **YES_REQUIRED**, or **NO**.

The uninstallers may be unable to remove some files; for example, files that are open in another application. In addition, the uninstallers remove only directories and files that were written by the Abaqus installation procedure. You may want to delete any remaining directories and files manually.

To remove Abaqus licensing:

Abaqus licensing can be removed in any of the following ways:

- From the **Start** menu, select **Programs→Abaqus Licensing→Uninstall Abaqus Licensing**.
- From the **Start** menu, select **Settings→Control Panel→Add or Remove Programs**. Select Abaqus licensing, and click **Change or Remove Programs**.
- To use the Abaqus licensing silent uninstaller, execute the following batch file from the command line or from another batch/script file:

```
simulia_dir\License\installation_info\uninstaller\  
silent_license_uninstall.bat
```

The licensing uninstallers generate the following log file:

```
simulia_dir\License\installation_info\uninstall_log.log
```

You should create a backup of any files you wish to keep, such as your Abaqus license file (typically named **abaquslm.lic**) and the FLEXnet Licensing server debug log file (typically named **abaquslm.log**).

To remove the Abaqus products:

Abaqus products can be removed in any of the following ways:

- From the **Start** menu, select **Programs→Abaqus *release*→Uninstall Abaqus *release***, where *release* refers to the Abaqus release.
- From the **Start** menu, select **Settings→Control Panel→Add or Remove Programs**. Select Abaqus, and click **Change or Remove Programs**.
- To use the Abaqus product silent uninstaller, execute the following batch file from the command line or from another batch/script file:

```
simulia_dir\abaqus_dir\release\installation_info\uninstaller\  
silent_product_uninstall.bat
```

The product uninstallers generate the following log file:

```
simulia_dir\abaqus_dir\release\installation_info\uninstall_log.log
```


The uninstallers should remove the Abaqus shortcuts from all user profiles. The client uninstall program (see “Client installation for Windows,” Section 2.4.4) removes shortcuts and files installed on the client computer; the Abaqus products on the mounted network drive are not affected.

2.6 Uninstalling Abaqus documentation

This section describes how to remove Abaqus documentation from your computer.

2.6.1 UNIX and Linux platforms

The following procedures explain how to remove Abaqus documentation from UNIX and Linux platforms.

To remove an installation of HTML and PDF documentation:

The Abaqus HTML and PDF documentation can be removed safely by using one of the following procedures to stop the web server and delete all of the installed documentation releases or a release-specific directory and all subordinate files. You must have administrator privileges to uninstall Abaqus components.

- To remove all installed releases of the Abaqus documentation:
 1. Set your current directory to the parent directory for the Abaqus documentation (*simulia_dir*).
 2. If you use the monitor process to serve the HTML documentation (i.e., the Abaqus web server) or use the monitor process to perform searches (i.e., an existing non-Abaqus web server), stop the monitor process using the command

```
Documentation/bin/monitor -k
```

3. Remove the documentation directory by typing the command

```
rm -rf Documentation
```

4. Remove the web server restart command from the computer’s system startup file. Contact your system administrator or platform vendor for information on the system startup file for your computer.
 5. Additional steps may be necessary if you are using an existing non-Abaqus web server. For more information, consult the SIMULIA Answers database in the SIMULIA Online Support System. The SIMULIA Online Support System is accessible from the **My Support** page at www.simulia.com.
- To remove a specific release of the Abaqus documentation:
 1. Set your current directory to the *simulia_dir/Documentation* directory.

2. If the release that you are removing uses the monitor process to serve the HTML documentation (i.e., the Abaqus web server) or uses the monitor process to perform searches (i.e., an existing non-Abaqus web server), stop the monitor process using the command

```
bin/monitor -k
```

3. Type the following commands

```
rm -rf installation_info/release
rm -rf taxis/release
rm -rf docs/release
```

where *release* refers to the release of the Abaqus documentation you wish to delete.

After you remove a release of the documentation, you can configure the web server to serve the remaining HTML collection using the following procedure:

1. Set your current directory to the *simulia_dir/Documentation* directory.
2. Verify that the release number of the release that you wish to serve is referenced in the *taxis.cnf* file as follows:

```
Default Database = %INSTALLDIR%/taxis/release
```

where *release* refers to the documentation release you wish to serve.

3. Run the command

```
installation_info/release/initializeExecutable
```

4. To start the web server to serve the remaining documentation collection, run the command

```
installation_info/release/startServer
```

To remove a PDF-only documentation installation:

1. Set your current directory to the parent directory for the Abaqus documentation (*simulia_dir*).
2. Remove the documentation directory by typing the command

```
rm -rf Documentation
```

2.6.2 Windows platforms

You can use the Windows uninstall shortcuts provided, the **Add or Remove Programs** tool, or the Abaqus silent uninstaller batch files to remove Abaqus documentation. The silent uninstallers allow you to automate these tasks from another batch/script file. The silent uninstallers are available only on Windows platforms. You must have Windows Administrator privileges to uninstall Abaqus components.

INSTALLING Abaqus

Both the Windows uninstall shortcuts and the silent uninstaller batch files generate a log file that you can review. The log file indicates whether the uninstall was successful and if you should reboot/restart your computer. This information is shown in the last two lines of the log file. For example,

```
Uninstall Status: SUCCESS
Restart Needed: YES_RECOMMENDED
```

The **Uninstall Status** line will indicate either **SUCCESS** or **INCOMPLETE**. The **Restart Needed** line will indicate either **YES_RECOMMENDED**, **YES_REQUIRED**, or **NO**.

The uninstallers may be unable to remove some files; for example, files that are open in another application. In addition, the uninstallers remove only directories and files that were written by the Abaqus installer. You may want to delete any remaining directories and files manually.

To remove an installation of HTML and PDF documentation:

The Abaqus HTML and PDF documentation can be removed in any of the following ways:

- From the **Start** menu, select **Programs**→**Abaqus release Documentation**→**Uninstall Documentation**, where *release* refers to the release of the Abaqus documentation you wish to delete.
- From the **Start** menu, select **Settings**→**Control Panel**→**Add or Remove Programs**. Select the Abaqus documentation, and click **Change or Remove Programs**.
- To use the Abaqus documentation silent uninstaller, execute the following batch file from the command line or from another batch/script file:

```
simulia_dir\Documentation\installation_info\release\  
html_uninstaller\silent_doc_uninstall.bat
```

The documentation uninstallers generate the following log file:

```
simulia_dir\Documentation\installation_info\release\  
uninstall_log.log
```

Other releases of the Abaqus documentation on your system are not affected by this procedure.

To remove a PDF-only documentation installation:

The Abaqus PDF documentation can be removed in any of the following ways:

- From the **Start** menu, select **Programs**→**Abaqus release PDF Documentation**→**Uninstall Documentation**, where *release* refers to the release of the documentation you wish to delete.
- From the **Start** menu, select **Settings**→**Control Panel**→**Add or Remove Programs**. Select the Abaqus documentation, and click **Change or Remove Programs**.
- To use the Abaqus documentation silent uninstaller, execute the following batch file from the command line or from another batch/script file:

```
simulia_dir\Documentation\installation_info\release\  
html_uninstaller\silent_doc_uninstall.bat
```

The documentation uninstallers generate the following log file:

```
simulia_dir\Documentation\installation_info\release\  
uninstall_log.log
```

Other releases of the Abaqus documentation on your system are not affected by this procedure.

3. Abaqus licensing

This chapter describes network licensing; updating an Abaqus 6.10 license file; upgrading a license server and tools; redundant license server configurations; and the FLEXnet license server manager, licensing administration tools, and options file.

3.1 Network licensing

Abaqus 6.10 licensing is provided in a network licensing format. Network licensing allows a limited number of analysis jobs and interactive sessions to be run simultaneously on any supported computer connected over a network. Abaqus network licensing uses the FLEXnet network license manager from Acreo Software (formerly Macrovision) to control an Abaqus license server, which is a process running on a single computer (license server host) on a network. Abaqus products can run on any supported computer on the network, including the license server host, as long as the necessary tokens are available.

Note: A network card is required for a computer to operate as a license server.

Multiple license servers can be selected to provide redundancy in case the license server or its host fails. For information on redundant license servers, see “Redundant license server configurations,” Section 3.4.

Any supported computer on the network can be used to run the license server as long as the computers that will run Abaqus can communicate with the license server. A computer running Windows can act as the license server host for a UNIX network and vice versa.

Each Abaqus analysis or interactive session must contact the license server prior to beginning execution and periodically while Abaqus is running. Therefore, Abaqus must have uninterrupted communication with the license server throughout the analysis or interactive session. If the license server or the computer on which it runs fails, it should be restarted as quickly as possible to ensure minimal interruption of Abaqus jobs.

License activity is recorded in a log file, called the server log. Tools to manage licensing activities are described in “FLEXnet Licensing administration tools,” Section 3.6, and The FLEXnet Licensing End User Guide Version 10.8; the latter document is available for download from the **Licensing** section of the **Support** page at www.simulia.com.

Restrictions can be applied to tokens to control access to Abaqus. A number of tokens can be reserved for certain users, or access can be denied to specific users and computers. These restrictions are defined in the network license options file. See “Using the options file,” Section 3.7, for more information. You can also customize the behavior of a license server through settings in the Abaqus environment file. See “License management parameters,” Section 4.1.6, for information on the environment file settings.

The license server continuously tracks which tokens on the network are “available” and which are “in use.” Prior to beginning an analysis or interactive session, the Abaqus job (client) requests tokens from the license server. If tokens are in the available pool, the license server transfers tokens from the

available pool to the in-use pool, and the tokens remain in the in-use pool until the end of the analysis or interactive session. If no tokens are available, an analysis job is placed in the license queue and an interactive session exits with an error message stating that no tokens are available. At the normal completion of the analysis or interactive session, the Abaqus client notifies the license server and the server returns the tokens to the available pool. If the analysis or interactive session aborts, the tokens are returned immediately.

For help in resolving questions relating to Abaqus licensing, see Appendix F, “Troubleshooting Abaqus network licensing.”

3.2 Updating an Abaqus license file

You may need to update an Abaqus 6.10 license file; for example, when you obtain a new license file to replace an expired license file or to revise license features. You may also need to replace a license file from a previous release of Abaqus. For example, if you manually upgraded an existing license server to a version of FLEXnet Licensing higher than Version 10.8.7, you need to update the license file for Abaqus 6.10 without reinstalling and overwriting the existing licensing utilities.

You must have write permissions to update the license file. If your license file is independent of other products using FLEXnet Licensing, you can replace the current license file with the new one using two methods: restarting the license server or forcing the server to reread the license file.

To update an Abaqus license file and restart the license server:

1. Backup the current license file by saving it with a new name.
2. Copy the new license file to the same location as the original license file using the name of the original license file.
3. Run the FLEXnet Licensing tool **lmdown** (see “lmdown,” Section 3.6.2) to shut down the currently running Abaqus license server.
4. Restart the Abaqus license server. For Windows platforms use the method described in “Starting the server using **LMTOOLS**,” Section 3.6.9. For UNIX and Linux platforms use the method described in “Abaqus licensing installation details,” Section 2.3.

To update and reread an Abaqus license file:

1. Backup the current license file by saving it with a new name.
2. Copy the new license file to the same location as the original license file using the name of the original license file.
3. Run the FLEXnet Licensing tool **lmreread** (see “lmreread,” Section 3.6.6) to force the license server to reread the new license file.

The license file that you update must maintain its file name and path name (location). If either the file name or path name change, you should uninstall licensing and then run the licensing installation procedure (see Chapter 2, “Installing Abaqus”). The default name of the Abaqus license file is **abaquslm.lic**. You can determine the name of the Abaqus license file by running the command

```
simulia_dir/License/lmstat -c [port]@license_server_host
```

where *simulia_dir* refers to the SIMULIA parent directory and *license_server_host* is the hostname of the license server. If the license server is using a port that is not between 27000 and 27009, you must specify the port number as well.

If your Abaqus license features are integrated with other products using a combined FLEXnet license file, replace the Abaqus licensing information in the combined license file with the updated information. You must run the FLEXnet Licensing tool **lmreread** (see “lmreread,” Section 3.6.6) to force the license server to reread the new license file. For more information on combined license files, see The FLEXnet Licensing End User Guide Version 10.8, which is available for download from the **Licensing** section of the **Support** page at www.simulia.com.

3.3 Upgrading a license server and administration tools

Abaqus 6.10 licensing requires FLEXnet Licensing Version 10.8.7; the license server must be upgraded if it is an earlier version. You can use the FLEXnet Licensing tool **lmver** or **lmstat** (see “lmstat,” Section 3.6.7) to check the version number of the license server. You must upgrade a license server that was installed with a previous release of Abaqus.

If you need to upgrade the license server, you should use the licensing installation procedure to upgrade the license server automatically. If you want to configure the license server manually, you can use the licensing installation procedure to install only the FLEXnet Licensing administration tools. For more information, see “Licensing installation procedure,” Section 2.1.2.

If your Abaqus license file is integrated with FLEXnet license files for other products using a combined license file, you must ensure that the **lmgrd** version meets the FLEXnet version compatibility rules. For more information, see the FLEXnet Licensing End User Guide Version 10.8, which is available for download from the **Licensing** section of the **Support** page at www.simulia.com.

3.4 Redundant license server configurations

Abaqus licensing offers two forms of server redundancy to guard against license disruptions: redundant license server triads and license file lists. A redundant license server triad consists of three license servers that function as one using the same license key. A license file list consists of multiple independent license servers, each with its own license key; token requests will check each license server in the list until available tokens are found. For a detailed comparison of the two redundancy configurations, refer to Answer 891 in the SIMULIA Online Support System. (The SIMULIA Online Support System is

accessible through the **My Support** page at www.simulia.com.) Configuration of license file lists is not covered in detail in this manual.

A redundant license server triad requires three computers. At least two license servers must be running for licenses to be granted. Network communications must be consistent and reliable among the three computers used in a redundant server triad. Redundant license server triads are considered to be an advanced use of FLEXnet Licensing. More detailed requirements for redundant license servers appear in the FLEXnet Licensing End User Guide Version 10.8, which is available for download from the **Licensing** section of the **Support** page at www.simulia.com.

You can install and configure a redundant license server triad using either an automatic or a manual process. The manual process is required if your servers use different port numbers; otherwise, the automatic installation is recommended. In both cases you will have to start the servers manually using the FLEXnet license server manager **lmgrd**. In their initial startup, the redundant license servers must be started simultaneously (within one minute of each other).

3.4.1 Installing a redundant license server triad automatically

Use the following procedure to automatically install and configure a redundant license server triad.

1. Copy the license file that you received to each of the three license server computers.
2. On one of the server computers, use the Abaqus licensing installation procedure to automatically install, configure, and start the Abaqus license server (see “Licensing installation procedure,” Section 2.1.2).

Once the installation program detects a redundant license file, it will not be able to start the server. However, it will install and configure the license server automatically.

3. Follow the instructions that appear in the dialog boxes to complete the licensing installation. When prompted, enter the port number on which the servers will run. In addition, enter the hostnames or IP addresses of your servers next to the corresponding host ids.
4. Repeat Steps 2 and 3 for the remaining two server computers. The server information must be entered in the same order for each installation.
5. Start the servers using the procedure described in “Starting a redundant license server triad,” Section 3.4.3.

3.4.2 Installing a redundant license server triad manually

Use the following procedure to manually install and configure a redundant license server triad. This procedure is necessary if the three servers in your configuration use different ports.

1. On each of the three license server computers, use the Abaqus licensing installation procedure to install the Abaqus FLEXnet Licensing administration tools (see “Licensing installation procedure,” Section 2.1.2).

The licensing tools are saved in the *simulia_dir/License* directory, where *simulia_dir* is the SIMULIA parent directory.

2. Copy the license file that you received to the licensing directory on one of the server machines. Rename this license file **abaquslm.lic**.
3. Edit the **abaquslm.lic** file, and replace the string **this_host** with the actual hostname of your computer. Verify that the hostname corresponds to the host id provided. Use *hostname.domain_name* if you have client machines that are not local to the server network.
4. Add the network port number to the **SERVER** line. It is recommended that you do not use a port number between 27000 and 27009.
5. Repeat this process for each **SERVER** line.
6. Append the following keywords to the first **SERVER** line:

```
PRIMARY_IS_MASTER HEARTBEAT_INTERVAL=60
```

For example:

```
SERVER bifrost.simulia.com 69084992 2501 \  
PRIMARY_IS_MASTER HEARTBEAT_INTERVAL=60  
SERVER tank.simulia.com 69094954 2501  
SERVER willow.simulia.com 69094344 2501  
VENDOR ABAQUSLM
```

where each server is configured to use port number 2501 and the numbers beginning with 690 are the host ids provided in the **abaquslm.lic** file for each server host. The port number must be assigned for each server, but the number need not be the same on every server.

Note: Each line in the license file must be 80 characters or less. You can use a backslash character (\) to indicate a line continuation; keywords that appear on the line after a backslash are considered part of the previous line in the license file.

7. Save the **abaquslm.lic** file, and copy the file to the licensing directory on each of the other two server computers.
8. On Windows platforms, you must install licensing as a service on each license server using the procedure described in “Installing licensing as a Windows service,” Section 3.6.8.
9. Start the license servers using the procedure described in “Starting a redundant license server triad,” Section 3.4.3.

3.4.3 Starting a redundant license server triad

After installing and configuring the license servers on all three computers, you must use the following procedure to start each of the servers within one minute of each other.

UNIX and Linux platforms

Start the Abaqus license server by executing the following command from the licensing directory:

```
./lmgrd -c abaquslm.lic -l +abaquslm.log
```

Windows platforms

You must be logged in as Administrator to start the server.

1. From the **Start** menu, select **Programs**→**Abaqus Licensing**→**Licensing utilities**.
2. Verify that **Configuration using Services** is enabled on the **Service/License File** tabbed page, and select the **Start/Stop/Reread** tab.
3. Click **Start Server**.

3.5 FLEXnet license server manager lmgrd

The license server manager **lmgrd** handles the initial contact with the client application programs, passing the connection on to the appropriate vendor daemon. It also starts and restarts the vendor daemons.

The **lmgrd** program is the main daemon program for FLEXnet Licensing. When you invoke **lmgrd**, the program looks for a license file that contains information about vendors and features. On UNIX and Linux platforms we recommend that you run **lmgrd** as a non-privileged user (not root).

See The FLEXnet Licensing End User Guide Version 10.8 for more information on FLEXnet Licensing. This manual is available for download from the **Licensing** section of the **Support** page at www.simulia.com.

Syntax and Options

```
lmgrd [-c license_file_list] [-l [+]debug_log_path] [-2 -p] [-local]
[-x lmdown] [-x lmremove] [-z] [-v] [-help]
```

-c *license_file_list*

Specifies the license files to use.

-l [+]*debug_log_path*

Writes debugging information to the file *debug_log_path*. This option uses the letter **l**, not the numeral 1. Prepending *debug_log_path* with the + character appends logging entries.

-2 -p

Restricts usage of **lmdown** (see “lmdown,” Section 3.6.2), **lmreread** (see “lmreread,” Section 3.6.6), and **lmremove** (see “lmremove,” Section 3.6.5) to a FLEXnet Licensing administrator who is by default root. If there is a UNIX group called “ladmin,” use is restricted to members of that group. If root is not a member of this group, root does not have

permission to use any of the three tools listed above. If you use **-2 -p** when starting **lmgrd**, no user on Windows platforms can shut down the license server with **lmdown**.

-local

Restricts **lmdown** and **lmreread** to be run only from the same machine where **lmgrd** is running.

-x lmdown

Disables the **lmdown** command (no user can run **lmdown**). If **lmdown** is disabled, you need to stop **lmgrd** via **kill pid** (UNIX platforms) or stop the **lmgrd** and vendor daemon processes through the Windows **Task Manager** or Windows service. On UNIX platforms, verify that the **kill** command does not contain the **-9** option.

-x lmremove

Disables the **lmremove** command (no user can run **lmremove**).

-z

Runs in foreground. The default behavior is to run in the background. If **-l debug_log_path** is present, no windows are used. If no **-l** options are specified, separate windows are used for **lmgrd** and each vendor daemon.

-v

Prints **lmgrd** version number and copyright and exits (does not launch **lmgrd**).

-help

Displays usage information and exits.

3.6 FLEXnet Licensing administration tools

FLEXnet Licensing provides administration tools to help manage the network licensing activities. The Abaqus licensing installation procedure installs the license file and the Abaqus license server. The FLEXnet Licensing administration tools are installed in *simulia_dir/License*. If you have installed Abaqus products on the license server, you can access the **License** directory using the **abaqus** command. Running the command

```
abaqus licensing
```

without additional arguments displays a command usage summary of all available FLEXnet Licensing administration tools. The FLEXnet Licensing End User Guide Version 10.8 contains detailed information on the syntax and options of the FLEXnet Licensing administration tools. You can download this manual from the **Licensing** section of the **Support** page at www.simulia.com.

On Windows platforms the licensing tools are also available using the FLEXnet Licensing toolchest **LMTOOLS**. The **LMTOOLS** toolchest can be accessed from the **Start** menu or by executing one of the following commands:

```
abagus licensing lmtools
```

or

```
simulia_dir/License/lmtools
```

The following sections describe the syntax and options of several commonly used FLEXnet Licensing administration tools and **LMTOOLS** procedures:

- “lmdiag,” Section 3.6.1
- “lmdown,” Section 3.6.2
- “lmhostid,” Section 3.6.3
- “lmpath,” Section 3.6.4
- “lmremove,” Section 3.6.5
- “lmreread,” Section 3.6.6
- “lmstat,” Section 3.6.7
- “Installing licensing as a Windows service,” Section 3.6.8
- “Starting the server using **LMTOOLS**,” Section 3.6.9

3.6.1 lmdiag

The **lmdiag** tool allows you to diagnose problems when you cannot check out a license.

Syntax and Options

```
lmdiag [-c license_file_list] [-n] [feature[:keyword=value]]
```

-c *license_file_list*

Diagnose the specified files.

-n

Run in non-interactive mode; **lmdiag** will not prompt for any input in this mode. In this mode extended connection diagnostics (see below) are not available.

feature

Diagnose this feature only.

keyword=value

If a license file contains multiple lines for a particular feature, you can select a particular line for **lmdiag** to report on.

For example,

```
lmdiag f1:HOSTID=12345678
```

attempts a checkout on the line with the hostid “12345678.” *keyword* can be one of the following:

- VERSION
- HOSTID
- EXPDATE
- KEY
- VENDOR_STRING
- ISSUER

If no *feature* is specified, **lmdiag** operates on all features in the license files in your list. **lmdiag** first prints information about the license, then attempts to check out each license. If the checkout succeeds, **lmdiag** indicates this. If the checkout fails, **lmdiag** gives you the reason for the failure. If the checkout fails because **lmdiag** cannot connect to the license server, you have the option of running “extended connection diagnostics.”

These extended diagnostics attempt to connect to each TCP/IP port on the license server machine and detect if the port number in the license file is incorrect. **lmdiag** indicates each TCP/IP port number that is listening. If it is an **lmgrd** process, **lmdiag** indicates this as well. If **lmdiag** finds the vendor daemon for the feature being tested, it indicates the correct port number for the license file to correct the problem.

See The FLEXnet Licensing End User Guide Version 10.8 for additional information. This manual is available for download from the **Licensing** section of the **Support** page at www.simulia.com.

To use LMTTOOLS (Windows platforms) to run **lmdiag**:

1. From the **Start** menu, select **Programs→Abaqus Licensing→Licensing utilities**.
2. Verify that **Configuration using Services** is enabled on the **Service/License File** tabbed page, and select the **Server Diags** tab.
3. Click **Perform Diagnostics**.

3.6.2 **lmdown**

The **lmdown** tool allows for the graceful shutdown of selected license daemons (both **lmgrd** and selected vendor daemons) on all machines.

Syntax and Options

```
lmdown -c license_file_list [-vendor vendor] [-q] [-all]
```

-c *license_file_list*

Use the specified license files. Specifying **-c** *license_file_list* is always recommended with **lmdown**.

-vendor *vendor*

Shut down only this vendor daemon. **lmgrd** continues running.

-q

Do not prompt or print a header. Otherwise, **lmdown** asks “Are you sure? [y/n]:.”

-all

If multiple servers are specified, automatically shut down all of them. **-q** is implied with **-all**.

You can protect the unauthorized execution of **lmdown** when you start up the license manager daemon **lmgrd**. Shutting down the servers causes users to lose their licenses. See the **-local**, **-2 -p**, and **-x** options in “FLEXnet license server manager lmgrd,” Section 3.5, for details about securing access to **lmdown**.

If **lmdown** encounters more than one server (for example, if **-c** specifies a directory with many ***.lic** files), a choice of license servers to shut down is presented.

Note: On UNIX platforms, do *not* use **kill -9** to shut down the license servers. On Windows platforms, if you must use the **Task Manager** to kill the FLEXnet Licensing service, be sure to end the **lmgrd** process first, then all the vendor daemon processes.

To stop and restart a single vendor daemon, use **lmdown -vendor vendor**, then use **lmreread -vendor vendor** to restart the vendor daemon (see “lmreread,” Section 3.6.6).

When shutting down a three-server redundant license server, there is a one-minute delay before the servers shut down. **lmdown** shuts down all three license servers of a set of redundant license servers. If you need to shut down one of a set of redundant license servers (not recommended because if either of the remaining machines becomes unavailable, the license server will stop serving licenses), you must kill both the **lmgrd** and vendor daemon processes on that license server machine.

To use LMTTOOLS (Windows platforms) to run lmdown:

1. From the **Start** menu, select **Programs**→**Abaqus Licensing**→**Licensing utilities**.
2. Verify that **Configuration using Services** is enabled on the **Service/License File** tabbed page, and select the **Start/Stop/Reread** tab.
3. Click **Stop Server**.

3.6.3 lmhostid

The `lmhostid` tool returns the FLEXnet Licensing host id of the current platform.

Syntax and Options

```
lmhostid [-n]
```

-n

Only the host id is returned as a string, which is appropriate to use with `HOSTID=` in the license file. Header text is suppressed.

Example

The following is an example of `lmhostid` output:

```
lmutil - Copyright (c) 1989-2008 Acresto Software
The FLEXlm host ID of this machine is "69021c89"
```

To use LMTOOLS (Windows platforms) to run lmhostid:

1. From the **Start** menu, select **Programs**→**Abaqus Licensing**→**Licensing utilities**.
2. Verify that **Configuration using Services** is enabled on the **Service/License File** tabbed page, and select the **System Settings** tab.
The host id is listed under **Ethernet Address**.

3.6.4 lmpath

The `lmpath` tool allows direct control over FLEXnet license path settings. It is most useful for checking current license path settings for diagnostic purposes.

Syntax and Options

```
lmpath {-add | -override} {vendor | all} license_file_list
```

-add

Prepends *license_file_list* to the current license file list or creates the license file list, if it does not exist, initializing it to *license_file_list*. Duplicates are discarded.

-override

Overrides the existing license file list with *license_file_list*. If *license_file_list* is the null string (""), the specified list is deleted.

- `lmpath -override all ""`
Deletes the value of `LM_LICENSE_FILE`.
- `lmpath -override vendor ""`
Deletes the value of `VENDOR_LICENSE_FILE`

vendor

A vendor daemon name. Affects the value of *VENDOR_LICENSE_FILE*.

all

Refers to all vendor daemons. Affects the value of *LM_LICENSE_FILE*.

license_file_list

A colon-separated list on UNIX platforms and a semicolon-separated list on Windows platforms. If *license_file_list* is the null string (""), the specified entry is deleted.

Note: *lmpath* works by setting *\$HOME/.flexmrc* on UNIX platforms and the FLEXnet Licensing registry entry on Windows platforms.

To display the current license path settings, use the command

```
lmpath -status
```

The following information is displayed:

```
lmutil - Copyright (C) 1989-2008 Acresto Software  
Known Vendors:
```

```
demo: ./counted.lic:./uncounted.lic
```

```
Other Vendors:
```

```
/usr/local/flexlm/licenses/license.lic
```

Where the path is set to a directory, all of the *.lic files are listed separately.

To use LMTOOLS (Windows platforms) to run *lmpath*:

1. From the **Start** menu, select **Programs**→**Abaqus Licensing**→**Licensing utilities**.
2. Verify that **Configuration using Services** is enabled on the **Service/License File** tabbed page, and select the **Utilities** tab.
3. Click **List All Vendor Paths**.

3.6.5 **lmremove**

The **lmremove** tool allows you to remove a single user's license for a specified feature. If the application is active, it rechecks out the license shortly after it is freed by **lmremove**. If an Abaqus process terminates abnormally, it may not return license tokens to the license pool even though the tokens are no longer needed. In this situation **lmremove** can be used to return Abaqus/CAE and Abaqus/Viewer tokens to the license pool. The *user*, *user_host*, *display*, *server_host*, *port*, and *handle*

information must be obtained from the output of **lmstat -a**. The **lmremove** tool should not be used to return Abaqus analysis job tokens to the license pool; see Appendix F, “Troubleshooting Abaqus network licensing,” for more information.

Note: The **lmremove** tool does not free licenses for use by other jobs. To temporarily free a license for use by another job, a running analysis job can be suspended using the Abaqus **suspend** utility (refer to “Job execution control,” Section 3.2.29 of the Abaqus Analysis User’s Manual, for details). A running analysis job can be terminated using the Abaqus **terminate** utility or the appropriate operating system utility to stop the executable for the analysis job. For an example of using the **lmremove** tool, see “lmstat,” Section 3.6.7.

Syntax and Options

lmremove [-c *license_file_list*] *feature user user_host display*

or

lmremove [-c *license_file_list*] -h *feature server_host port handle*

-c *license_file_list*

Specifies the license files.

feature

Name of the feature checked out by the user.

user

Name of the user whose license you are removing, as reported by **lmstat -a**.

user_host

Name of the host the user is logged on to, as reported by **lmstat -a**.

display

Name of the display where the user is working, as reported by **lmstat -a**.

server_host

Name of the host on which the license server is running, as reported by **lmstat -a**.

port

TCP/IP port number where the license server is running, as reported by **lmstat -a**.

handle

License handle, as reported by **lmstat -a**.

The **lmremove** tool removes all instances of *user* on *user_host* and *display* from usage of *feature*. If the -c *license_file_list* option is specified, the indicated file is used as the license file. The -h variation uses *server_host*, *port*, and license *handle*, as reported by **lmstat -a**.

3.6.6 lmreread

The **lmreread** tool causes the license manager to reread the license file and start any new vendor daemons that have been added. In addition, all currently running vendor daemons are signaled to reread the license file and their end-user options files for changes in feature licensing information or option settings. If report logging is enabled, any report log data still in the vendor daemon's internal data buffer are flushed. **lmreread** recognizes changes to server machine host names but cannot be used to change server TCP/IP port numbers.

If the optional vendor daemon name is specified, only the named daemon rereads the license file and its end-user options file (in this case **lmgrd** does not reread the license file).

Syntax and Options

```
lmreread [-c license_file_list] [-vendor vendor] [-all]
```

-c *license_file_list*

Use the specified license files.

-vendor *vendor*

Only this vendor daemon rereads the license file. **lmgrd** restarts the vendor daemon if necessary.

-all

If more than one **lmgrd** is specified, instructs all **lmgrds** to reread.

You may want to protect the execution of **lmreread**. See the **-2 -p** and **-x** options in "FLEXnet license server manager **lmgrd**," Section 3.5, for details about securing access to **lmreread**.

To stop and restart a single vendor daemon, use **lmdown -vendor vendor**, then use **lmreread -vendor vendor**, which restarts the vendor daemon.

Note: If you use the **-c license_file_list** option, the license files specified are read by **lmreread**, not by **lmgrd**; **lmgrd** rereads the file it read originally.

To use LMTTOOLS (Windows platforms) to run lmreread:

1. From the **Start** menu, select **Programs**→**Abaqus Licensing**→**Licensing utilities**.
2. Verify that **Configuration using Services** is enabled on the **Service/License File** tabbed page, and select the **Start/Stop/Reread** tab.
3. Select **Abaqus FLEXnet License Manager**, and click **ReRead License File**.

3.6.7 lmstat

The **lmstat** tool helps you monitor the status of all network licensing activities, including:

- Daemons that are running
- License files
- Users of individual features
- Users of features served by a specific vendor daemon

The **lmstat** tool prints information that it receives from the license server; therefore, it does not report on unserved licenses such as uncounted licenses. To report on an uncounted license, the license must be added to a served license file and the application must be directed to use the license server for that license file (via *@host*, *port@host*, or **USE_SERVER**). Queued users and licenses shared due to duplicate grouping are also not returned by **lmstat**.

Syntax and Options

```
lmstat [-a] [-c license_file_list] [-f [feature]] [-i [feature]] [-s [server]]
[-S [vendor]] [-t timeout_value]
```

-a

Displays all information.

-c *license_file_list*

Specifies the license files to use.

-f [*feature*]

Displays the users of *feature*. If *feature* is not specified, usage information for all features is displayed.

-i [*feature*]

Displays information from the **FEATURE/INCREMENT** line for the specified *feature* or for all features if *feature* is not specified.

-s [*server*]

Displays the status of all license files in **\$VENDOR_LICENSE_FILE** or **\$LM_LICENSE_FILE** on *server* or on all servers if *server* is not specified.

-S [*vendor*]

Lists all users of the specified vendor's features.

-t *timeout_value*

Sets the connection timeout to *timeout_value*. This limits the amount of time **lmstat** spends attempting to connect to *server*.

Note: The `lmstat -a` command is a potentially expensive command. With many active users, this command generates a lot of network activity.

The `lmremove` tool requires the output of the `lmstat -a` command, as shown in the example below.

Example

The output for the command

```
simulia_dir/License/lmstat -a
```

where *simulia_dir* is the SIMULIA parent directory, looks similar to the following:

```
License server status: 27000@firestar
  License file(s) on firestar: simulia_dir/License/abaquslm.lic:
  firestar: license server UP (MASTER) v10.8
Vendor daemon status (on firestar):
  ABAQUSLM: UP v10.8
Feature usage info:
Users of cae: (Total of 4 licenses issued; Total of 1 license
in use)
  "cae" v6.10, vendor: ABAQUSLM
  floating license
    smith watt watt (v6.10) (firestar/27000 101),
start Tue 3/1 9:29
```

where

<code>smith</code>	<i>user</i>	User name
<code>watt</code>	<i>user_host</i>	Host where user is running
<code>watt</code>	<i>display</i>	Display where user is running
<code>v6.10</code>	<i>release</i>	Release of feature
<code>firestar</code>	<i>server_host</i>	Host where license server is running
<code>27000</code>	<i>port</i>	Port on <i>server_host</i> where license server is running
<code>101</code>	<i>handle</i>	License handle
<code>start Tue 3/1 9:29</code>	<i>checkout_time</i>	Time that this license was checked out

Note: The `lmremove` tool does not free licenses for use by other jobs (see “`lmremove`,” Section 3.6.5, for more information).

To use the **lmremove** tool to free the license for a job run by user **smith**, you would use the command

```
abaqus licensing lmremove cae smith watt watt
```

where

cae	<i>feature</i>	Name of the feature checked out by the user
smith	<i>user</i>	Name of the user whose license you are removing, as reported by lmstat -a
watt	<i>user_host</i>	Name of the host the user is logged into, as reported by lmstat -a
watt	<i>display</i>	Name of the display where the user is working, as reported by lmstat -a

To use LMTTOOLS (Windows platforms) to run `lmstat -a`:

1. From the **Start** menu, select **Programs**→**Abaqus Licensing**→**Licensing utilities**.
2. Verify that **Configuration using Services** is enabled on the **Service/License File** tabbed page, and select the **Server Status** tab.
3. Verify that **Display Everything** is enabled, and click **Perform Status Enquiry**.

3.6.8 Installing licensing as a Windows service

You must be logged in as Administrator to install licensing as a Windows service. You use the installation procedure to install the licensing administration tools. You must use the following procedure to install and start the service:

1. To access **LMTTOOLS** from the **Start** menu, select **Programs**→**Abaqus Licensing**→**Licensing utilities**.
2. Verify that **Configuration using Services** is enabled on the **Service/License File** tabbed page, and select the **Config Services** tab.
3. In the **Service Name** text field, type **Abaqus FLEXnet License Manager**.
4. Specify the paths to the **lmgrd.exe** file, the license file, and the debug log file, as shown in Figure 3-1.
5. Toggle on **Use Services**.
Start Server at Power Up becomes available.
6. Toggle on **Start Server at Power Up**, and click **Save Service**.
7. Select the **Start/Stop/Reread** tab, and click **Start Server**.
8. From the main menu bar, select **File**→**Exit** to close the dialog box.

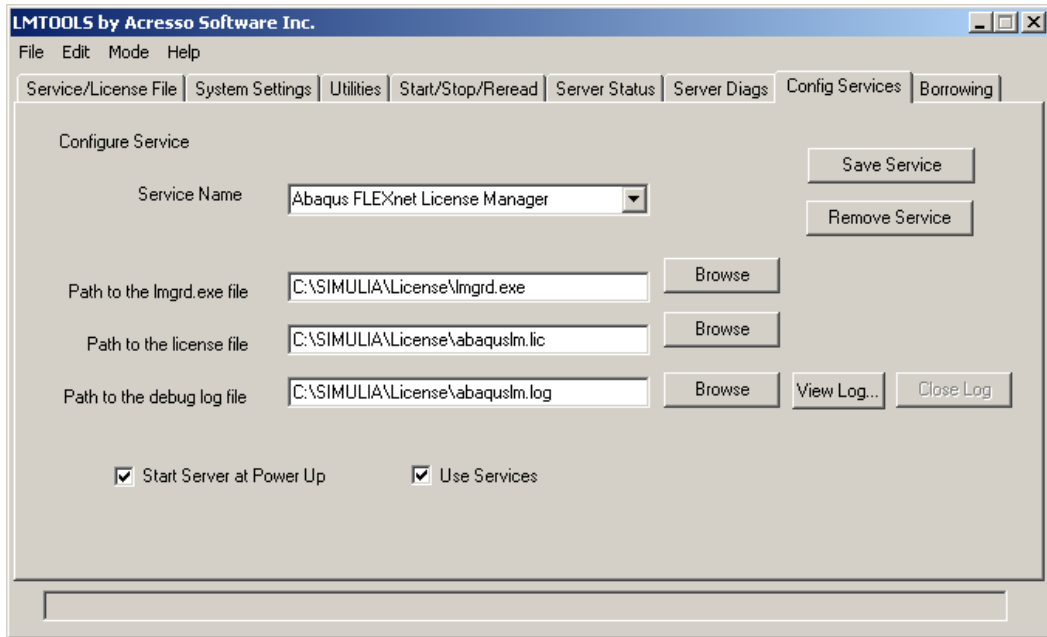


Figure 3–1 The **Config Services** tabbed page of **LMTOOLS**.

3.6.9 Starting the server using LMTOOLS

You must be logged in as Administrator to start the server.

1. To access **LMTOOLS** from the **Start** menu, select **Programs**→**Abaqus Licensing**→**Licensing utilities**.
2. Verify that **Configuration using Services** is enabled on the **Service/License File** tabbed page, and select the **Start/Stop/Reread** tab.
3. Click **Start Server**.
4. From the main menu bar, select **File**→**Exit** to close the dialog box.

3.7 Using the options file

The options file allows the license administrator to control various operating parameters of FLEXnet Licensing. Parameters, such as **EXCLUDE**, **INCLUDE**, **MAX**, and **RESERVE**, can be used to control Abaqus license usage. Usage can be controlled by user name, host name, display, IP address, or

project. When specifying the features to control, you may specify features listed on a **FEATURE** line in the license file. Using a **PACKAGE** name in place of a **FEATURE** name applies the option to all of the components in the package. You should not specify individual **COMPONENTS** listed on the **PACKAGE** line, as it may lead to unexpected behavior. See Chapter 5 of the FLEXnet Licensing End User Guide Version 10.8 for detailed information on creating an options file. (This manual is available from the **Licensing** section of the **Support** page at www.simulia.com.)

The following is a sample options file:

```
INCLUDEALL INTERNET 192.168.0.*
RESERVE 5 abaqus USER robert
RESERVE 10 abaqus HOST demeter
MAX 5 abaqus USER jim
EXCLUDE abaqus USER khan
EXCLUDE viewer HOST guardian
```

In this options file:

- Connections are accepted only from machines with the IP addresses 192.168.0.*.
- Five tokens of feature **abaqus** are reserved for user *robert*.
- Ten tokens of feature **abaqus** are reserved for users on host *demeter*.
- The user *jim* is limited to five tokens of feature **abaqus** at a time.
- The user *khan* is excluded from accessing feature **abaqus**.
- The users on host *guardian* are excluded from accessing feature **viewer** (Abaqus/Viewer).

3.8 Reporting license usage data

This section describes methods for creating license usage reports. If your Abaqus license server is on a wide area network (WAN) and Abaqus users in different locations receive technical support and customer service from more than one regional SIMULIA office, you are required to submit license usage reports to SIMULIA. For more information about reporting requirements, see Answer 2604 in the SIMULIA Online Support System. The SIMULIA Online Support System is accessible through the **My Support** page at www.simulia.com.

The reporting tools are available to all Abaqus licensees. Two utilities are provided with Abaqus to generate license usage reports:

- The **usage** utility (described in “Manually creating license usage reports,” Section 3.8.1) can be used to manually generate license usage reports for a specified range of dates.
- The scheduled license reporting utility (**schLicRpt**, described in “Scheduling automated license usage reporting,” Section 3.8.2) can be used to set up automated license usage reports on a monthly or quarterly basis.

The license usage reporting utilities use information from the FLEXnet debug log file. You can ensure that this log is created by including the **-1** option when starting the license server using **lmgrd**

(see “FLEXnet license server manager lmgrd,” Section 3.5). To subtotal reports according to user location, you must set the **computer_location** environment file parameter. More details about the environment file parameters can be found in “License management parameters,” Section 4.1.6.

3.8.1 Manually creating license usage reports

The **usage** utility allows you to generate CSV-formatted reports detailing the usage of Abaqus licenses. Two types of usage reports are generated: summary usage reports and daily usage reports. Summary usage reports provide usage information for each Abaqus license feature subtotaled by user location or individual user. Summary usage reports list the total number of licenses requested, number of licenses granted, number of licenses denied, and total hours of license usage within a specified period. Daily usage reports list the peak and average number of analysis and Abaqus/CAE license tokens checked out on each day within a specified period. Daily usage reports provide aggregate data for all users and locations.

The license usage reports calculate token usage to two significant figures. If the total hours of license usage (in a summary report) or the average number of tokens checked out (in a daily report) is less than 0.005, the usage report lists these values as zero.

If the FLEXnet debug log file (or log files) is located on a remote server, you must copy it to a directory that is accessible to the local computer before running the **usage** utility.

Syntax and Options

```
abaqus licensing usage {-summary report_file | -daily report_file} [-nouser]
-log {debug_log_file | debug_log_directory} [-start start_date] [-end end_date]
```

-summary

Use this option to create a report summarizing usage of each license feature by user or location. The report is written to the path and file name specified by *report_file*.

-daily

Use this option to create a report providing peak and average daily usage of analysis and Abaqus/CAE license tokens. The report is written to the path and file name specified by *report_file*.

-nouser

This option suppresses usage data about individual users in a summary report. It reports aggregate data for the entire license server. If the **computer_location** environment file parameter is set, it also reports aggregate data for each location. This option has no effect when used in conjunction with the **-daily** option; daily reports do not include information about individual users or locations.

-log

Use this option to specify the path to the debug log file or to a directory that includes multiple debug log files; the utility reviews all log files located in the specified path for usage data within the specified reporting period.

-start

Use this option to specify the starting date and time for the reporting period. If the **-start** option is not specified, the reporting period begins with the oldest recorded item in the debug log files. The date and time must be entered in the following format:

dd-~~mmm~~-yyyy [_hh[:mm[:ss]]]

For example, **-start 01-jan-2008_09:00:00** indicates a starting time of 9:00 AM on January 1, 2008. Specifying a time is optional; the default start time is **00:00:00**. If the time is specified, the hour field (*hh*) must be entered in 24-hour format.

-end

Use this option to specify the ending date and time for the reporting period. If the **-end** option is not specified, the reporting period ends with the most recent recorded item in the debug log files. The date and time must be entered in the following format:

dd-~~mmm~~-yyyy [_hh[:mm[:ss]]]

For example, **-end 31-mar-2008_18:00:00** indicates an ending time of 6:00 PM on March 31, 2008. Specifying a time is optional; the default end time is **23:59:59**. If the time is specified, the hour field (*hh*) must be entered in 24-hour format.

3.8.2 Scheduling automated license usage reporting

Abaqus provides a utility that automatically generates both summary and daily license usage reports according to a monthly or quarterly schedule. The utility creates a script containing commands to generate the reports, then it installs a scheduled task (on Windows platforms) or a crontab entry (on UNIX or Linux platforms) to execute the script on a monthly or quarterly basis. The debug log file can be located on a server that is remote from the computer on which automated license usage reporting is scheduled. If desired, you can instruct the utility to automatically e-mail the reports to a specified address as soon as they are generated.

You should create a new directory dedicated to license usage reporting and run the scheduled license usage reporting utility from within this directory. The script for report generation (**usage_report.bat** or **usage_report.sh**), a utility configuration file (**schLicRpt.cfg**), and the generated reports are all saved to the directory in which you run the utility. Generated reports use the following file naming convention:

abaqus_usagen_siteID_date.csv

where *siteID* is the customer number that SIMULIA assigned to your site, and the *date* is written numerically in *yymmdd* format. The type of usage report is indicated by *n*: **1** indicates a summary usage report, and **2** indicates a daily usage report. For example, **abaqus_usage2_01ABCD_080515.csv** is the file name for a daily usage report created at site 01ABCD on May 15, 2008.

To run the utility, use the following command:

```
abaqus schLicRpt
```

The utility will prompt you for the required information, including the location of the debug log file, the frequency of report generation, and, if you choose to e-mail the reports, the fully qualified domain name of your SMTP server and the e-mail address to which reports are sent. If you run the utility again from the same directory, default settings will be provided during the configuration process based on the contents of the saved configuration file. You can, therefore, make minor modifications to the reporting configuration settings without reentering all of the required information.

You must have access to the **schTasks.exe** utility (on Windows) or **crontab** utility (on UNIX or Linux) to run the scheduled license reporting utility. For best performance, it is recommended that you execute the utility on the Abaqus license server host computer, which is possible only if Abaqus products have been installed on this computer. The utility should be executed only on one computer within your site. Otherwise, you will generate duplicate usage reports and may degrade the performance of the license server when the reports are generated.

Use the following procedures to remove automatically generated reports from your system:

UNIX and Linux platforms

To list current crontab entries, run the following command:

```
crontab -l
```

To edit the current crontab entries, run the following command:

```
crontab -e
```

To remove all of the current crontab entries, run the following command:

```
crontab -r
```

For more details on crontab, type **man crontab** from any prompt to view the crontab man page for your system.

Windows platforms

To list current scheduled tasks, run the following command:

```
schTasks /query
```

To remove the automated Abaqus license usage report, run the following command:

```
schTasks /delete /tn abaqus_usage_report
```

4. Customizing the Abaqus environment

This chapter describes how to use environment file parameters to customize the Abaqus execution procedure, including how to define analysis batch queues. Example files are provided at the end of the chapter.

4.1 Using the Abaqus environment file

The Abaqus execution procedure (see Chapter 3, “Job Execution,” of the Abaqus Analysis User’s Manual) reads environment files to determine the various parameters that are used to run a job. The procedure searches three directories for the environment file, **abaqus_v6.env**, in the following order:

1. The **site** subdirectory of the Abaqus installation (see Appendix B, “Abaqus release directories and files”). An environment file must exist in this directory.

The settings in this file are ignored when a job is submitted to a remote queue. Instead, the settings in the **site** subdirectory on the remote computer are used.

2. The user’s home directory. This environment file is optional and will affect all jobs submitted from the user’s account.

On Windows platforms, for Abaqus to locate the environment file in the user’s home directory, the full path to the user’s home directory must be specified using the **HOME** environment variable or a combination of the **HOMEDRIVE** and **HOMEPath** environment variables.

3. The current working directory. This environment file is optional and will affect all jobs submitted from the current working directory.

If the same parameter is defined in more than one environment file or is defined more than once within an environment file, the last definition encountered will be used.

The environment file for Abaqus 6.10 uses Python syntax. See the Abaqus Scripting User’s Manual for more detailed information on Python syntax. The environment file for Abaqus 6.10 must be named **abaqus_v6.env**. Attempting to use a Version 5 format environment file will cause errors.

Environment file entries have the following syntax:

```
parameter=value
```

All parameters must have a value. The following syntactic rules also apply:

- All parameters are case sensitive.
- A string value must be enclosed in a pair of double or single quotes.
- Comments are preceded by a number sign (#). All characters following a number sign on a line are ignored. Number signs within a quoted string are part of the string, not the beginning of a comment.
- Blank lines are ignored.
- Lists must be enclosed in square brackets ([]). Individual items in the list are separated by commas. Entries take the form:

```
parameter=[value1, value2, value3]
```

- Tuples must be enclosed in parentheses (()). Individual items in the tuple are separated by commas. If the tuple is enclosed in parentheses and contains only one value, a comma has to follow the value. Entries take the form:

```
parameter=(value1, value2)
```

- Embedded single quotes do not require special handling if they are placed within a double-quoted string. For example, "my value's" is translated as my value's. The same holds true for double quotes embedded in a single-quoted string. Quotes of the same type as the enclosing quotes can be embedded if they are prefixed by the backslash (\) character. Strings in a list or a tuple must be enclosed in quotes.
- Triple-quoted (""") strings can span more than one line, and no special treatment of quotes within the string is necessary. Entries take the form:

```
parameter="""  
multi-line  
value  
"""
```

Examples of many of the environment file parameters are available in the **abaqus_v6.env** and **abaqusinc.env** environment files found in the **site** subdirectory of the Abaqus installation. Care should be taken when merging customized **abaqus_v6.env** settings from an earlier release into the current release. Settings from the earlier release may not be compatible with the new release. The available parameters are described in the following sections.

4.1.1 Memory and disk management parameters

Management of memory and disk resources for Abaqus/Standard and Abaqus/Explicit is discussed in detail in “Managing memory and disk use in Abaqus,” Section 3.4.1 of the Abaqus Analysis User’s Manual. The relevant parameters are listed here along with a single parameter, **abq_ker_memory**, that is used to manage memory in Abaqus/CAE and Abaqus/Viewer.

The available units for memory sizes are mb (megabytes) and gb (gigabytes). If the units are not specified, the size is assumed to be in megabytes.

abq_ker_memory

The maximum amount of memory that can be allocated by the Abaqus/CAE and Abaqus/Viewer kernel. If the limit is exceeded, Abaqus/CAE displays an error message. The **abq_ker_memory** must be specified in MB (megabytes) and is used only for 32-bit systems. For Windows 32-bit systems the default is 1800 MB; for Linux 32-bit systems the default is 2800 MB.

Increasing the memory limit is not recommended unless you are using a Windows 32-bit system with the boot option **/3GB /userva=SizeInMBytes** to extend the amount of memory available for Abaqus/CAE. In this case the limit can be changed to 2800 MB (the same as Linux).

scratch

Full path name of the directory to be used for scratch files. The default value on UNIX is the value of the `$TMPDIR` environment variable or `/tmp` if `$TMPDIR` is not defined. On Windows the default value is the value of the `%TEMP%` environment variable or `\TEMP` if `%TEMP%` is not defined. During the analysis a subdirectory will be created under this directory to hold the analysis scratch files. The name of the subdirectory is constructed from the user's user name, the job ID, and the job's process number. The subdirectory and its contents are deleted upon completion of the analysis.

memory

Maximum amount of memory or maximum percentage of the physical memory that can be allocated during the input file preprocessing and during the Abaqus/Standard analysis phase. The default is different on different platforms; for details, refer to the SIMULIA Online Support System, which is accessible from the **My Support** page at www.simulia.com.

4.1.2 Parallelization parameters

Parallelization in Abaqus/Standard is discussed in detail in "Parallel execution in Abaqus/Standard," Section 3.5.2 of the Abaqus Analysis User's Manual. Parallelization in Abaqus/Explicit is discussed in detail in "Parallel execution in Abaqus/Explicit," Section 3.5.3 of the Abaqus Analysis User's Manual. The relevant parameters are listed here.

auto_convert

If this parameter is set equal to **ON** and an Abaqus/Explicit analysis is run in parallel with **parallel=domain**, the **convert=select**, **convert=state**, and **convert=odb** options will be run automatically at the end of the analysis if required. The default value is **ON**.

cpus

Number of processors to use during an analysis run if parallel processing is available. The default value for this parameter is 1.

domains

The number of parallel domains in Abaqus/Explicit. If the value is greater than 1, the domain decomposition will be performed regardless of the values of the **parallel** and **cpus** variables. However, if **parallel=DOMAIN**, the value of **cpus** must be evenly divisible into the value of **domains**. If this parameter is not set, the number of domains defaults to the number of processors used during the analysis run if **parallel=DOMAIN** or to 1 if **parallel=LOOP**.

max_cpus

Maximum number of processors allowed if parallel processing is available. If this parameter is not set, the number of processors allowed equals the number of available processors on the system.

mp_file_system

Type of file system available for an MPI-based parallel Abaqus analysis. The parameter has to be set to a tuple; for example,

```
mp_file_system=(SHARED,LOCAL)
```

The first item in the tuple refers to the directory where the job was submitted, while the second refers to the job's scratch directory. If the file system hosting a directory is **LOCAL**, Abaqus will copy the required analysis files to the remote host machines and, at the end of the run, copy the output files back. In this case it must be possible to create the job's directory structure on all the hosts in **mp_host_list**. A **SHARED** file system means that the host machines share the same file system and file transfer is not necessary. With the recommended default (**DETECT**, **DETECT**) setting, Abaqus will determine the type of file system that exists. An MPI-based parallel Abaqus/Explicit analysis will use the scratch directory only if a user subroutine is used, whereas Abaqus/Standard normally writes large temporary files in this directory. Running on a local file system will generally improve the performance.

mp_host_list

List of host machine names to be used for an MPI-based parallel Abaqus analysis, including the number of processors to be used on each machine; for example,

```
mp_host_list=[['maple',1],['pine',1],['oak',2]]
```

indicates that, if the number of **cpus** specified for the analysis is 4, the analysis will use one processor on a machine called **maple**, one processor on a machine called **pine**, and two processors on a machine called **oak**. The total number of processors defined in the host list has to be greater than or equal to the number of **cpus** specified for the analysis. If the host list is not defined, Abaqus will run on the local system. When using a supported queuing system, this parameter does not need to be defined. If it is defined, it will get overridden by the queuing environment.

mp_mode

Set this variable equal to **MPI** to indicate that the MPI components are available on the system. Set **mp_mode=THREADS** to use the thread-based parallelization method. The default value is **MPI** where applicable.

mp_mpi_implementation

This variable determines the underlying MPI implementation to use. Generally, this variable does not need to be specified.

mp_mpirun_options

String of options that are passed to the MPI launcher for an MPI-based parallel Abaqus analysis. Generally this variable does not need to be specified.

mp_mpirun_path

A dictionary to define the full path to the MPI launcher for a given MPI implementation; for example,

```
mp_mpirun_path={HP:'/opt/hpmpi/bin/mpirun'}
```

mp_rsh_command

Preferred command to open a remote shell on the machines specified by **mp_host_list**. Abaqus needs to open a remote shell to create and remove directories and files if the file system is not shared. The default value for this option is platform dependent; for example,

```
mp_rsh_command='rsh -n -l %U %H %C'
```

The following placemarkers are used:

- %U** Username.
- %H** The host where the remote shell is opened.
- %C** The command to be executed on the host.

Abaqus automatically uses secure copy to copy files to and from remote hosts if this parameter is set to use secure shell.

order_parallel

The default direct solver ordering mode in Abaqus/Standard if you do not specify the parallel ordering mode on the **abaqus** command line. If this parameter is set equal to **OFF**, the solver ordering will not be performed in parallel. If this parameter is set equal to **ON**, the solver ordering will be run in parallel. The default for parallel solver ordering is **ON**.

parallel

The default parallel equation solution method in Abaqus/Explicit if the user does not specify the parallel method on the **abaqus** command line. Possible values are **DOMAIN** or **LOOP**; the default value is **DOMAIN**.

standard_parallel

The default parallel execution mode in Abaqus/Standard if you do not specify the parallel mode on the **abaqus** command line. If this parameter is set equal to **ALL**, both the element operations and the solver will run in parallel. If this parameter is set equal to **SOLVER**, only the solver will run in parallel. The default parallel execution mode is **ALL**.

4.1.3 Job customization parameters

abq_cosimulation_lower_port

This variable specifies the lowest port number in the range of TCP/UDP port numbers available for co-simulation between two Abaqus analyses; it is valid only for Abaqus/CAE. The default value is 48000.

abq_cosimulation_upper_port

This variable specifies the highest port number in the range of TCP/UDP port numbers available for co-simulation between two Abaqus analyses; it is valid only for Abaqus/CAE. If this value is not specified, it is set to 1000 more than **abq_cosimulation_lower_port**.

auto_calculate

If this parameter is set to **ON**, the postprocessing calculator will be launched automatically at the end of an analysis if the execution procedure detects that output database postprocessing is necessary. If this parameter is set to **OFF**, the postprocessing calculator will not run at the end of an analysis even if the execution procedure detects that it is necessary to postprocess the output database file. The default value is **ON**.

average_by_section

This parameter is used only for an Abaqus/Standard analysis. If this parameter is set equal to **OFF**, the averaging regions for output written to the data (**.dat**) file and results (**.fil**) file are based on the structure of the elements. If this parameter is set equal to **ON**, the averaging regions also take into account underlying values of element properties and material constants. In problems with many section and/or material definitions the default value of **OFF** will, in general, give much better performance than the nondefault value of **ON**. See “Output to the data and results files,” Section 4.1.2 of the Abaqus Analysis User’s Manual, for further details on the averaging scheme.

cae_error_limit

This variable defines the maximum number of error messages that will be sent from an analysis job to Abaqus/CAE; it is valid only for Abaqus/CAE. The default value is 50.

cae_no_parts_input_file

This variable defines the format of the input file generated by Abaqus/CAE; it is valid only for Abaqus/CAE. If this variable is set to **ON**, Abaqus/CAE will generate an input file without parts and assemblies. The default value is **OFF**. For more information, see “Writing input files without parts and assemblies,” Section 9.9.4 of the Abaqus/CAE User’s Manual.

cae_warning_limit

This variable defines the maximum number of warning messages that will be sent from an analysis job to Abaqus/CAE; it is valid only for Abaqus/CAE. The default value is 200.

double_precision

The default precision version of Abaqus/Explicit to run if you do not specify the precision version on the **abacus** command line. Possible values are **EXPLICIT** (only the Abaqus/Explicit analysis is run in double precision), **BOTH** (both the Abaqus/Explicit packager and analysis are run in double precision), **CONSTRAINT** (the constraint packager and constraint solver in Abaqus/Explicit are run in double precision, while the Abaqus/Explicit packager and analysis continue to run in single precision), or **OFF** (both the Abaqus/Explicit packager and analysis are run in single precision). The default is **OFF**.

max_history_requests

This parameter specifies the maximum number of history requests allowed in an Abaqus analysis. The default value is 10,000. History output in Abaqus is intended for relatively frequent output requests for small portions of a model and is displayed in X - Y plots in the Visualization module of Abaqus/CAE (Abaqus/Viewer). Requesting large amounts of history output will cause performance problems in both analysis and postprocessing of an Abaqus job. For vector- or tensor-valued output variables, each component is considered to be a single request. In the case of element variables, history output will be generated at each integration point. For example, requesting history output of the tensor variable S (stress) for a C3D10M element will generate 24 history output requests: (6 components) \times (4 integration points). When requesting history output of vector- and tensor-valued variables, it is recommended that individual components be selected where applicable. In cases where large amounts of history output are required, it is recommended that the data be written to the output database (`.odb`) as field output from which history data can be extracted using the Visualization module of Abaqus/CAE.

madymo_home

Full path name to the directory containing the MADYMO installation.

odb_output_by_default

If this parameter is set to **ON**, output database output will be generated automatically. If this parameter is set to **OFF**, output database request keywords must be placed in an input file to obtain output database output and to allow the analysis to be restarted. The default value is **ON**.

onCaeGraphicsStartup

Optional function to be executed before Abaqus/CAE or Abaqus/Viewer begins. This function allows the user to change the graphics options. See “Tuning graphics cards,” Section 5.3, for more information on this function. This function should not normally be changed.

onCaeStartup

Optional function to be executed before Abaqus/CAE begins. See “Customizing Abaqus/CAE startup,” Section 4.3.3, for examples of this function.

onDesignStartup

Optional function to be executed before Abaqus/Design begins.

onJobCompletion

Optional function to be executed after the Abaqus job completes. A function specified in the Abaqus environment file in the current directory will be executed first, followed by the function in the user's home directory, and then the function in the Abaqus installation environment file. Multiple functions in the same environment file will result in only the last definition being used. See "Job variables," Section 4.1.9, for a list of variables available to this function.

onJobStartup

Optional function to be executed before the Abaqus job begins. See "Job variables," Section 4.1.9, for a list of variables available to this function.

onViewerStartup

Optional function to be executed before Abaqus/Viewer begins.

printed_output

By default, the values of all *PREPRINT parameters are NO and no results are printed to the data file. Set the **printed_output** parameter equal to **ON** to obtain the same preprint information in the data file as if

```
*PREPRINT, CONTACT=YES, ECHO=YES, HISTORY=YES, MODEL=YES
```

were included in the input file. Setting **printed_output** equal to **ON** can also cause a large volume of tabular results to be printed to the data file (unless printed output control options are used to limit the output). Setting **printed_output** equal to **ON** will replicate the default data file output from Abaqus 5.8. The default value for this variable is **OFF**.

run_mode

Default run mode (**INTERACTIVE**, **BACKGROUND**, or **BATCH**) if the user does not specify the run mode on the **abaqus** command line when running the analysis products. The default value is **BACKGROUND**. This variable should not be set to **BATCH** unless batch queues are defined.

split_dat

If this variable is set to **ON**, the data file will be split into two pieces. The output from the user input processing will be put in a file with a **.pre** extension. The analysis output file will still have a **.dat** file extension. The default value is **OFF**.

unconnected_regions

If this variable is set to **ON**, Abaqus/Standard will create element and node sets in the output database for unconnected regions in the model during a **datacheck** analysis. Element and node sets created with this option are named **MESH COMPONENT N**, where *N* is the component number.

4.1.4 System customization parameters

admin

This parameter prevents unauthorized modification of environment file parameters. Set this parameter equal to a list of environment file parameters that cannot be changed in a lower-level **abaqus_v6.env** file. Unless otherwise noted, all system and job customization parameters can be locked out. Commands in the installation directory have the highest precedence, followed by commands in the user's home directory, and then commands in the current working directory. Thus, an Abaqus user cannot change environment file commands that were locked out by the Abaqus account manager.

ask_delete

If this parameter is set to **OFF**, the user will not be asked whether old job files of the same name should be deleted; the files will be deleted automatically. The default value is **ON**.

compile_cpp

C++ compile command. The command used at SIMULIA is included in the **abaqus_v6.env** file in the **site** subdirectory of the Abaqus installation. This command should not normally be changed. It may be either a string or a tuple of strings. If the command is a tuple of strings, each string must represent a single command line argument. The values of the placemaker are determined by the Abaqus execution procedure or by the command line options and cannot be modified by the user. The values of the placemaker replace the placemaker in the **compile_cpp** string. The following placemaker is available:

%I Search directories for include files.

compile_fortran

FORTRAN compile command. The command used at SIMULIA is included in the **abaqus_v6.env** file in the **site** subdirectory of the Abaqus installation. This command should not normally be changed. It may be either a string or a tuple of strings. If the command is a tuple of strings, each string must represent a single command line argument.

The compilation of FORTRAN files using Fortran 90 freeform specifications is not supported by default. The **abaqus_v6.env** file in the **site** subdirectory of the Abaqus installation includes comments that discuss the compile options.

file_format

Format of results file output (**ASCII** or **BINARY**). This parameter is valid only for Abaqus/Standard. The default value is **BINARY**.

link_exe

Command to link a postprocessing program. The command used at SIMULIA is included in the **abaqus_v6.env** file in the **site** subdirectory of the Abaqus installation. This command should not normally be changed. It can be either a string or a tuple of strings. If the command is a tuple of strings, each string must represent a single command line argument. The values of the placemarkers are determined by the Abaqus execution procedure or by the command line options and cannot be modified by the user. User-specified external libraries can be linked with the usual link commands. The following placemarkers are used in **link_exe**:

- %J** The job name (in this case the name of the executable to be created).
- %F** The name of the object file created from the user's source file.
- %M** The name of the internally created main object file.
- %B** The name of the shared library of utility functions.
- %O** The list of utility shared libraries for the output database application public interface.
- %L** The list of directories containing shared libraries (HP only).

link_sl

Command to link a shared library. The command used at SIMULIA is included in the **abaqus_v6.env** file in the **site** subdirectory of the Abaqus installation. This command, once its placemarkers have been exchanged, must be a valid command on the computer system where the shared library is linked. This command can be either a string or a tuple of strings. If the command is a tuple of strings, each string must represent a single command line argument. The values of the placemarkers are determined by the Abaqus execution procedure or by the command line options and cannot be modified by the user. The values of the placemarkers replace the placemarkers in the **link_sl** string. User-specified external libraries can be linked with the usual link commands. The following placemarkers are available:

- %F** The name of the compiled object file.
- %U** The name of the shared library.
- %A** The name of the archive of compiled object files to be linked into the shared library.
- %B** The name of the shared library of utility functions.
- %E** The name of the file containing the names of the symbols to be exported from the shared library.
- %L** The list of directories containing shared libraries (HP only).

nodb_cache_limit

Maximum size of the cache in the temporary file directory. Abaqus/CAE uses this cache for local data storage when you use a network ODB connector to read from a remote output database. Set the **nodb_cache_limit** parameter to the number of megabytes to which the cache size will be limited. The minimum value of **nodb_cache_limit** is 500, indicating that the cache size is limited to 500 MB. If you set the maximum cache size to be greater than the available free space, Abaqus/CAE reduces it to a value that is equal to the available free space.

plugin_central_dir

Full pathname of the directory containing Abaqus/CAE plug-ins. In most cases this is a directory at a central location that is accessible to all users at your site. For more information, see Chapter 78, “The Plug-in toolset,” of the Abaqus/CAE User’s Manual.

usub_lib_dir

Full path name of the directory containing optional user-defined libraries of Abaqus/Standard and/or Abaqus/Explicit user subroutines. Valid user subroutine libraries are platform specific, but the file base names are the same for all platforms. The base names are **standardU**, **explicitU**, and **explicitU-D**. Use this variable to avoid the cost of recompiling and/or relinking frequently used user subroutines. The **abaqus make** utility is used to create the shared libraries for use with this variable (see “Making user-defined executables and subroutines,” Section 3.2.13 of the Abaqus Analysis User’s Manual). User libraries created by the **user** option of the Abaqus/Standard and Abaqus/Explicit execution procedure will supersede any user libraries in this directory.

verbose

If this parameter is in the environment file, the execution procedure will print more information on job submission. Possible values are **ON**, **OFF**, **1**, **2**, and **3**. Set the value to **ON** or **1** to print the commands used to run application executables and some performance data. Set the value to **2** to print licensing transaction information. Set the value to **3** to print operating system environment settings. The output associated with the **verbose** parameter is written to standard output. The default value is **OFF**.

The following system customization parameters are used during documentation installation and to specify a web browser to display the Abaqus HTML documentation and context-sensitive help in Abaqus/CAE.

WARNING: *If you customize browser behavior for Firefox or you use a web browser not supported by Abaqus, you may encounter restrictive behavior that prevents the display of the documentation collection window. For example, you may be prompted to select a profile under which to run when attempting to access the HTML documentation with a copy of the web browser already running.*

browser_path

Full path to the web browser executable on UNIX and Linux platforms. The value of this parameter can be either a string or a list of strings. If a list of strings is specified, the first string must be the full path to the web browser executable and subsequent strings are arguments to customize the browser behavior. If any argument strings are included, at least one of them must contain `%s` for which the full uniform resource locator (URL) to the Abaqus HTML documentation will be substituted.

Abaqus automatically configures supported browsers to correctly display the HTML documentation. The configurations are different for each browser. This parameter can be used in conjunction with **browser_type** to clarify the browser being used. If **browser_path** is set equal to a string and **browser_type** is not set, the system checks the specified browser path for Firefox. If Abaqus does not detect Firefox, Abaqus assumes that an unsupported browser will be used and does not perform an automatic configuration. This parameter is ignored on Windows platforms.

browser_type

Web browser on UNIX and Linux platforms. To correctly display the HTML documentation, Abaqus automatically configures the browser according to the specified browser type. The possible settings are **FIREFOX** and **CUSTOM_BROWSER**. If you set **browser_type=CUSTOM_BROWSER** to use a web browser other than Firefox, no support or automatic configuration is provided. For more information, see “Using a web browser not supported by Abaqus,” Section 4.3.4.

This parameter can be used in conjunction with **browser_path** to directly specify an executable command for the browser. If **browser_type** is set to Firefox and **browser_path** is not set, the system searches the system path for Firefox. If the specified browser is not found, an error is displayed. This parameter is ignored on Windows platforms.

doc_root

Full uniform resource locator (URL) or path to the Abaqus HTML or PDF documentation. This variable is set during product installation when you provide the documentation URL and should not normally be changed. The required format is one of the following.

- For HTML documentation with a web server:
`http://computername:port_number/v6.10/`
- For HTML documentation with no web server:
`file:///simulia_dir/Documentation/docs/v6.10/index.html`
- For PDF-only documentation, only the path to the installation base directory is required:
`simulia_dir`

doc_root_type

The documentation formats specified by the **doc_root** parameter. Set to the string value `'html'` (the default) for an installation of HTML and PDF, or set to the string value `'pdf'` for PDF-only. These values must be lowercase.

The following parameters disallow the execution of the corresponding modules prior to testing for license activation through the startup file. They can be used to provide “friendly” messages if an attempt is made to execute an analysis module for which your site does not have a license. Use these parameters in conjunction with the **admin** parameter to ensure uniformity across your site.

no_aqua

Block execution of Abaqus/Aqua if value is set to **ON**.

no_background

Block background execution of Abaqus analysis jobs if value is set to **ON**.

no_batch

Block batch queue execution of Abaqus analysis jobs if value is set to **ON**.

no_cae

Block execution of Abaqus/CAE if value is set to **ON**.

no_design

Block execution of Abaqus/Design if value is set to **ON**.

no_explicit

Block execution of Abaqus/Explicit if value is set to **ON**.

no_interactive

Block interactive execution of Abaqus analysis jobs if value is set to **ON**.

no_standard

Block execution of Abaqus/Standard if value is set to **ON**.

no_viewer

Block execution of Abaqus/Viewer if value is set to **ON**.

4.1.5 Executable parameters

The Abaqus executables for the licensed modules are placed automatically into the **exec** subdirectory of the Abaqus installation. Prefix parameters (***_prefix**) can be set to **"time"** on UNIX systems to determine the execution time for a particular stage of an Abaqus run.

exe_prefix

Optional executable prefix for all Abaqus analysis executables. The default value is an empty string.

explicit_prefix

Optional executable prefix for Abaqus/Explicit. The value given for this prefix will override the value given for **exe_prefix**.

explicit_dp_prefix

Optional executable prefix for the double precision version of Abaqus/Explicit. The value given for this prefix will override the value given for **exe_prefix**.

package_prefix

Optional executable prefix for the Abaqus/Explicit PACKAGE program. The value given for this prefix will override the value given for **exe_prefix**.

pre_prefix

Optional executable prefix for the Abaqus analysis input file processor. The value given for this prefix will override the value given for **exe_prefix**.

standard_prefix

Optional executable prefix for the Abaqus/Standard program. The value given for this prefix will override the value given for **exe_prefix**.

4.1.6 License management parameters

License management customization parameters control the behavior of the Abaqus license server based on current network conditions and user needs. The Abaqus license server has been designed with default parameters that should be suitable for most users. The following parameters are provided for customization:

abaquslm_license_file

This parameter provides the host name of the computer running the Abaqus license server for licensing and is set during the product installation. If a single server is used, this parameter should be set to *port@license_server_host*, where *port* is the port number and *license_server_host* is the name of the computer running the server. If the port number is between 27000 and 27009, you do not need to include it. If redundant servers are used, the parameter should be set to a list of the names of the redundant server hosts and their ports. For example, if the host names for a set of three redundant servers are maple, pine, and oak, the following parameter definition is appropriate:

```
abaquslm_license_file="27000@maple:27000@pine:27000@oak"
```

Use a colon to separate the host names on UNIX platforms, and use a semicolon to separate the host names on Windows platforms.

academic

This parameter indicates whether an academic Abaqus client should use research or teaching license tokens. Setting this parameter to **TEACHING** will force the Abaqus client to use only teaching license tokens. Setting this parameter to **RESEARCH** or removing the parameter will force Abaqus to use only research license tokens. This parameter is set automatically during the product installation: if the license server contains an Abaqus teaching license file, the installation sets the parameter to **TEACHING**; otherwise, the parameter is set to **RESEARCH**.

cae_timeout

The number of minutes that an Abaqus/CAE or Abaqus/Viewer session will remain idle due to no user activity before returning its token to the license server. The default value is 60 minutes.

computer_location

A string that indicates the location of the local client computer. This parameter allows you to subtotal license usage reports by location. The license usage reporting utility compiles and organizes data according to the **computer_location** name. The default value is an empty string. If you do not change this default, the license usage report will not distinguish between different locations in the report.

lmhanglimit

The number of minutes that an Abaqus client will wait in the license queue to obtain licenses if no licenses are currently available. The default value of 0 forces the job to remain in the license queue indefinitely unless it is killed by the user.

lmlicensequeuing

This parameter indicates whether an Abaqus analysis job or an Abaqus/CAE or Abaqus/Viewer session using the **noGUI** option should queue for a license if one is not available. The default value is **ON**. If this parameter is set to **OFF**, jobs and Abaqus/CAE or Abaqus/Viewer sessions will terminate immediately if a license is not available.

lmlog

This parameter indicates whether license usage data should be written to the FLEXnet debug log file. To use the Abaqus license usage reporting utility (see “Reporting license usage data,” Section 3.8), this parameter must be set to **ON**, which is the default value. To suppress license usage data in the debug log file, set this parameter equal to **OFF**.

lmqueuesleep

The number of seconds that an Abaqus client will wait before checking the license queue to see if enough tokens are free. The default value is 30 seconds, which is the minimum allowed. Increasing this value will decrease network traffic when license queuing occurs.

lmsvrdownlimit

The number of minutes that an Abaqus client will attempt to connect to the license server if the license server is currently unavailable. The default value of 0 forces the job to attempt to connect indefinitely unless it is killed by the user.

4.1.7 Object-oriented batch queue parameter

The object-oriented interface for defining an analysis batch queue is easily customized and extended. An analysis batch queue is available from the Abaqus execution procedure command line once it has been instantiated and inserted into the dictionary of analysis queues. See “Defining analysis batch queues,” Section 4.2, for more information.

queues

Dictionary of queue names and objects. Queue names and their corresponding object instances are inserted into the dictionary as key/value pairs. The special queue name **default** can be used to designate a default queue. The default queue is used when the **run_mode** is set to **BATCH** and the **queue** option is not specified on the command line. It is also used when the **queue** option is specified on the command line but the named queue is blank or does not exist in the **queues** dictionary.

4.1.8 String-based batch queue parameters

The string-based interface for defining an analysis batch queue allows access to a fixed number of options. The parameters below can be used to construct command strings to control the execution of an Abaqus analysis. The only requirement is the string must be a valid command on the computer system where it is executed. See “Defining a string-based analysis batch queue,” Section 4.3.7, for an example.

after_prefix

Optional prefix to be output as part of the submit command when the **after** option is specified on the command line. The default value is an empty string.

queue_cmd

Default command to be used to submit a batch job when the **run_mode** is set to **BATCH**. This parameter, once its placemarkers have been exchanged, must be a valid command on the computer system where it is executed. The values of the placemarkers are defined by input specified on the command line of the Abaqus execution procedure or by environment file parameters. The values of the placemarkers replace the placemarkers in the **queue_cmd** string. The following placemarkers are available:

- %%** A percent (%) character.
- %A** The **after_prefix** string is substituted for **%A** when the **after** option is specified on the command line.

- %L** The log file name. This token will be replaced by *job-name.log*, where *job-name* is the value defined by the **job** command line option.
- %P** The **queue_prefix** string (see description below) is substituted for **%P** when the **queue** option is specified on the command line.
- %Q** The queue name from the **queue** command line option.
- %S** The command script file name. This token will be replaced by *job-name.com*, where *job-name* is the value defined by the **job** command line option.
- %T** The time from the **after** command line option.
- %J** The job name from the **job** command line option.
- %E** The token is substituted by the full path to the Python executable.
- %O** The full path to the output directory.

queue_name

List of names for batch commands, typically used for submitting jobs to queues other than the default (defined by **queue_cmd**). Then, elsewhere in the environment file, each of these command aliases must appear on the left side of the equal sign, with the desired command string on the right. This command string has the same format as the **queue_cmd** parameter. It can use the replaceable placemarkers in its construction as long as the final result is a valid system command.

queue_prefix

Optional prefix to be output as part of the submit command when a queue name is specified. The default value is an empty string.

4.1.9 Job variables

The following variables can be used in an **onJobStartup** or **onJobCompletion** function:

id

The job identifier specified as the value of the **job** option on the command line.

savendir

The path name to the directory from which the job was submitted.

screddir

The path name to the scratch directory.

analysisType

The type of analysis to be executed. Possible values are **EXPLICIT** and **STANDARD**.

In addition, for an MPI-based parallel job the following variables are available in an **onJobStartup** or **onJobCompletion** function:

host_list

List of host machine names that were used for the analysis, including the number of processors used on each machine. The format is identical to the **mp_host_list** environment variable (see “Parallelization parameters,” Section 4.1.2).

local_host

List of identifiers used to determine the host machine name from which the job was submitted (e.g., host name, IP address, aliases, etc.).

rsh_command

Command used to open a remote shell on the machines that were used during analysis. The format is identical to the **mp_rsh_command** environment variable (see “Parallelization parameters,” Section 4.1.2).

file_system

Tuple showing the type of file system used for the MPI-based parallel job. The first item in the tuple refers to the directory where the job was submitted, while the second refers to the job’s scratch directory. For MPI-based parallel Abaqus/Explicit analyses that do not use a user subroutine, the scratch directory will remain as **DETECT** unless it was set by the user.

cpus

Number of total processors used for the analysis summed across all host machines.

The following variables are available outside of the **onJobStartup** and **onJobCompletion** functions:

abaqus_version

A string that contains the Abaqus release.

analysisType

The type of analysis to be executed. Possible values are **EXPLICIT** and **STANDARD**.

applicationName

The name of the Abaqus execution procedure to be executed; e.g., **analysis**, **cae**, or **viewer**.

4.2 Defining analysis batch queues

Analysis batch queues are used to configure the way that Abaqus analysis jobs are run. They are particularly useful for integrating Abaqus with third-party batch queueing systems.

Two pieces of information are needed to run an Abaqus job: the syntax of the command used to execute the job and the job-specific information. The command used to execute the job is obtained from a queue definition in the Abaqus environment file. The job-specific information is obtained from the

command line options and the analysis parameters defined in the Abaqus environment file. The command line options are described in Chapter 3, “Job Execution,” of the Abaqus Analysis User’s Manual.

The command syntax and the job-specific information are used to construct a command to run an Abaqus job in an analysis queue. An object-oriented interface and a string-based interface are available for defining the analysis batch queues. The object-oriented interface is preferred because it is easily customized and extended. This section discusses the object-oriented interface; see “String-based batch queue parameters,” Section 4.1.8, for information on the string-based interface.

A set of predefined (built-in) queue classes is available within an Abaqus environment file. These classes can be instantiated to create user-defined queue objects. See “Object-oriented batch queue parameter,” Section 4.1.7, for information on the **queues** environment file parameter. In addition to the predefined queue classes, users can create their own queue classes to customize the way analysis jobs are executed. The new queue classes can be derived from the predefined queue classes to minimize coding by the user. The **driverQueues** module must be imported to instantiate a predefined queue or to derive a custom queue class. The predefined classes are described in “Queue class,” Section 4.2.1, and “Predefined derived classes,” Section 4.2.2. In addition, examples that illustrate how to extend predefined classes and create custom queue classes are provided in “Using predefined derived classes,” Section 4.3.5, and “Deriving and using custom queue classes,” Section 4.3.6, respectively.

4.2.1 Queue class

The **Queue** class is an abstract base class. All other analysis batch queue classes are derived from it. The class has no explicit constructor or members. The following methods of the **Queue** class are common to all derived classes:

__repr__(...)

This method returns the class name as a string. The string is printed as a description for the queue when **abaqus information=environment** is executed. Derived classes should override this method to provide a useful description of the queue objects instantiated from them.

createScript(...)

This method creates a Python script named *job-name.com* in the current working directory. This script is used to run the analysis. This method is called by the analysis execution procedure prior to the **submit** method. If the *job-name.com* file cannot be written in the current working directory, a `FileCreationError` exception is raised. The following argument is required:

options A dictionary containing the analysis options.

getDriverName(...)

This method returns the name of the command used to invoke the Abaqus execution procedure.

getPython(...)

This method returns the absolute path to the Abaqus Python interpreter as a string.

getNumRequiredTokens(...)

This method returns the number of license tokens required for an analysis as an integer. The following argument is required:

options A dictionary containing the analysis options.

spawn(...)

This method executes a command in a new process, waits for it to complete, and returns an integer representing the exit status of the command. If the command cannot be executed, a `SpawnError` exception is raised. The following arguments are required:

cmd A valid command string to be executed in the new process. If analyses are submitted to this queue from Abaqus/CAE, the command must return the status immediately; otherwise, the ability to monitor the progress of the analysis jobs in Abaqus/CAE may fail. Examples of commands that return the status immediately are **qsub**, **bsub**, **at**, **batch**, etc.

env A dictionary of environment variables available to the process.

The following argument is optional:

verbose A Boolean specifying whether the command string is printed to **stdout**. The default value is OFF.

submit(...)

This abstract method must be implemented by a derived class. This method is called by the analysis execution procedure to submit the analysis to a queue. The **submit** method must return an integer; a value of 0 indicates success, and a nonzero value indicates failure. When this method is called, the analysis execution procedure supplies the following required arguments:

options A dictionary containing the analysis options.

env A dictionary of environment variables available to the process.

Most derived classes call the **spawn** method from this method and return its exit status.

4.2.2 Predefined derived classes

The following analysis batch queue classes are derived from the **Queue** base class:

AtQueue class

The **AtQueue** class executes an analysis using the UNIX **at** command. This class overrides the following base class methods:

`__repr__(...)`

This method returns a string describing the class.

submit(...)

This method executes the UNIX **at** command to run the *job-name.com* analysis script at the time specified on the command line **after** option. If the **after** option is not specified on the command line, a `QueueError` exception is raised.

BatchQueue class

The **BatchQueue** class executes an analysis using the UNIX **batch** command. This class overrides the following base class methods:

__repr__(...)

This method returns a string describing the class.

submit(...)

This method executes the *job-name.com* analysis script under the UNIX **batch** command.

HoldQueue class

The **HoldQueue** class creates a *job-name.com* file and exits. This class overrides the following base class methods:

__repr__(...)

This method is reimplemented to provide a useful description.

submit(...)

This method prints a message stating that the *job-name.com* script was not submitted and returns a value of 0.

LSFQueue class

The **LSFQueue** class submits an analysis to the LSF queue named when the object was instantiated. If a name was not specified, the analysis is submitted to the default LSF queue. The following constructor argument is optional:

name Name of a valid LSF queue.

This class overrides the following base class method:

submit(...)

This method invokes the LSF **bsub** command to submit the *job-name.com* analysis script to an LSF batch cluster and returns the exit status of the **bsub** command.

NQSQueue class

The **NQSQueue** class submits an analysis to the NQS queue named when the object was instantiated. If a name was not specified, the analysis is submitted to the default NQS queue. The following constructor argument is optional:

name Name of a valid NQS queue.

This class overrides the following base class method:

submit(...)

This method invokes the NQS **qsub** command to submit the *job-name.com* analysis script to an NQS system and returns the exit status of the **qsub** command.

PBSQueue class

Queues instantiated from the **PBSQueue** class will create a *job-name.pbs* script and run the command **qsub job-name.pbs**. The *job-name.inp* and *job-name.com* files will be copied to the execution host, where the *job-name.com* script will be executed. After job completion, all output files will be copied back to the submission host. The following constructor argument is optional:

name Name of a valid PBS queue.

This class overrides the following base class method:

submit(...)

This method invokes the PBS **qsub** command to submit the *job-name.pbs* script to a PBS system and returns the exit status of the **qsub** command.

4.3 Examples

The examples included in this section illustrate various uses of the Abaqus environment file.

4.3.1 Example environment file

An example Windows environment file is shown below. This file will work on UNIX systems as well if you change the **scratch** directory setting appropriately. A sample environment file, **abaqusinc.env**, is included in the **site** subdirectory of the Abaqus installation to show the options used at SIMULIA.

```
scratch = "c:/temp"
if applicationName in ('analysis','datacheck','continue'):
    memory = "256 Mb"

def onCaeStartup():
    #Graphics preferences
    session.graphicsOptions.setValues(dragMode=AS_IS, displayLists=ON)
    # Print preferences
    session.printOptions.setValues(vpDecorations=OFF,
```

```

    vpBackground=OFF, rendition=COLOR,
    printCommand='lpr -S marley -P hp3')
session.psOptions.setValues(date=OFF)
# Job preferences
def setJobPreferences(module, userData):
    session.Queue(name='long', hostName='server',
        queueName='large', directory='/tmp')
addImportCallback('job', setJobPreferences)
# Visualization preferences
def setVisPreferences(module, userData):
    session.defaultOdbDisplay.contourOptions.setValues(
        renderStyle=SHADED, visibleEdges=EXTERIOR,
        contourStyle=CONTINUOUS)
addImportCallback('visualization', setVisPreferences)

```

The default compile and link environment variables for your computer have also been inserted in your site environment file.

4.3.2 Notifying users when a job is completed

The following is an example of how environment file commands can be used to notify UNIX system users when their job is finished. The notification method used depends on how the job was run and if the user is logged in. If the job was run interactively, the user will not be notified that the job has finished. If the user is still logged in when the job completes, a message will be output to the screen. If the user has logged out by the time the job completes, a message will be mailed to the user. The syntax of the **mail** command varies from system to system. Please consult your system documentation to determine the appropriate commands.

```

def onJobCompletion():
    import os, re

    userName = os.environ['USER']
    msg = 'Job %s has completed' % id

    # Run "who" command, pipe the output, and read into a list
    whopipe = os.popen('who', 'r')
    output = whopipe.readlines()
    whopipe.close()

```

CUSTOMIZING THE Abaqus ENVIRONMENT

```
# Find out if the user is logged in
loggedIn = 'no'
terminal = [ ]

for line in output:
    columns = re.split('[ ]+', line) # Split into blank separated columns
    name = columns[0] # User name is in the first column
    if name == userName:
        terminal.append(columns[1]) # Terminal at which user is logged in
        loggedIn = 'yes'

# Use "write" command if the user is logged in, use mail otherwise
if loggedIn == 'no':
    logFile = savedir + id + ".log"
    if os.path.exists(logFile):
        os.system('cat %s | Mail -s "%s" %s' % (logFile, msg, userName))
    else:
        os.system('Mail -s "%s" %s' % (msg, userName))
else:
    for termNum in terminal:
        os.system('echo "%s" | write %s %s' % (msg, userName, termNum))
```

4.3.3 Customizing Abaqus/CAE startup

The following example for the **onCaeStartup** parameter will establish viewport preferences and print options (including a print command), set up a remote execution queue for running Abaqus jobs, and set preferences for contour plots in the Visualization module:

```
def onCaeStartup():

    # Print preferences
    session.printOptions.setValues(vpDecorations=OFF,
        vpBackground=OFF, rendition=COLOR,
        printCommand='lpr -S server -P printer')
    session.psOptions.setValues(date=OFF, logo=OFF)

def initQueues(*args):
    session.Queue(name='long', hostName='server',
        queueName='large', directory='/tmp')
    addImportCallback('job', initQueues)
```

```

# Visualization preferences
def setVisPreferences(module, userData):
    import visualization
    session.defaultOdbDisplay.contourOptions.setValues(
        renderStyle=SHADED, visibleEdges=EXTERIOR,
        contourStyle=CONTINUOUS)
    addImportCallback('visualization', setVisPreferences)

```

Indented text must be valid Python commands. For more queue examples, see Chapter 18, “The Job module,” of the Abaqus/CAE User’s Manual.

4.3.4 Using a web browser not supported by Abaqus

Abaqus provides support only for the Firefox web browser on Linux platforms; however, it does provide the ability to use a web browser not supported by Abaqus for viewing HTML documentation. Examples are shown for setting the **browser_type** and **browser_path** system customization parameters in this situation.

The first example illustrates the parameter settings to specify a web browser not supported by Abaqus:

```

browser_type = CUSTOM_BROWSER
browser_path = ['full_path_to_browser', 'argument1', 'argument2', etc.]

```

where the first string contains the full path to the web browser and subsequent strings are arguments to customize the browser behavior. Refer to the specific web browser documentation for valid arguments.

The second example illustrates the parameter settings to specify the Opera browser (not supported by Abaqus) as the web browser:

```

browser_type = CUSTOM_BROWSER
browser_path = ['/usr/local/bin/opera', '-newwindow' '%s']

```

where `/usr/local/bin/opera` is the full path to the browser and the argument `-newwindow` opens a new browser window using the full uniform resource locator (URL) to the Abaqus HTML documentation. Use the argument `-newpage` to open a new page.

The third example illustrates the parameter settings to specify the Konqueror browser (not supported by Abaqus) as the web browser:

```

browser_type = CUSTOM_BROWSER
browser_path = '/opt/kde3/bin/konqueror'

```

where `/opt/kde3/bin/konqueror` is the full path to the browser. By default, a new browser window opens using the URL to the Abaqus HTML documentation.

4.3.5 Using predefined derived classes

The following example illustrates the instantiation of some predefined derived classes and their insertion into the **queues** dictionary:

```
run_mode = BATCH
from driverQueues import *
queues['atq'] = AtQueue()
queues['batchq'] = BatchQueue()
queues['hold'] = HoldQueue()
```

To submit an analysis using one of the queues, specify the queue name as the value for the analysis execution procedure **queue** parameter.

4.3.6 Deriving and using custom queue classes

To derive a custom queue class, the **driverQueues** module must be imported and the class must inherit directly or indirectly from the **Queue** class. Derived queues must provide an implementation for the **submit** method. Derived class methods can raise exceptions as needed. The predefined **QueueError** exception is provided as a general-purpose exception.

The following examples illustrate the derivation and use of custom queue classes:

```
run_mode = BATCH
from driverQueues import *
#-----
class NiceQueue(Queue):
    #-----
    def __repr__(self):
        return 'Executes analysis using UNIX nice command.'
    #-----
    def submit(self, options, env):
        job = options['job']
        after = options.get('after', '')
        verbose = options.get('verbose', 0)
        if options.get('after', ''):
            # a descriptive string must be supplied as data when
            # raising a QueueError exception
            #-----
            raise QueueError, \
                "after" is not a valid argument for this queue.'
```

```

    # run nice under bourne shell to eliminate platform dependencies
    cmd = "/bin/sh -c 'nice %s python ./%s.com 1>./%s.log 2>&1 &' \" \
        % (self.getDriverName(), job, job)
    return self.spawn(cmd, env, verbose)
#-----
class LSF_ResvQueue(LSFQueue):
    # For integration with LSF. This queue class supports cpu, memory,
    # and license reservations.
    #-----
    def __init__(self, name, memReserve=0, cpusReserve=0):
        LSFQueue.__init__(self, name)
        self.memReserve = memReserve
        self.cpusReserve = cpusReserve
    #-----
    def __repr__(self):
        return 'Submits to LSF %s queue (run "bqueues -l %s" for description)' \
            % (self.name, self.name)
    #-----
    def submit(self, options, env):
        job = options['job']
        verbose = options.get('verbose', 0)
        queue = self.name
        cpus = options.get('cpus', '1')
        if self.cpusReserve:
            cpus = self.cpusReserve
        resLst = [ ]
        # license reservation - For the following line to work, LSF
        # must be configured with a static or dynamic resource called
        # "abqtokens".
        #-----
        resLst.append('abqtokens=%d' % self.getNumRequiredTokens(options))
        # memory reservation
        if self.memReserve:
            from math import ceil
            resLst.append('mem=%d' % int(ceil(self.memReserve/float(cpus))))
        resStr = ''
        if resLst:
            import string
            resStr = '-R rusage[%s]' % string.join(resLst, ':')

```

```

        bsub = 'bsub -q %s -J %s -n %s -o %s.log -N %s %s python %s.com' % \
            (queue, job, cpus, job, resStr, self.getDriverName(), job)
        return self.spawn(bsub, env, verbose)

# queue definitions
queues['default']      = NiceQueue()
queues['hold']         = HoldQueue()
queues['benchmark']   = LSF_ResvQueue(name='benchmark', cpusReserve=16)
queues['dedicated']   = LSF_ResvQueue(name='dedicated')
queues['a500M']       = LSF_ResvQueue(name='a500M', memReserve=500)
queues['a1500M']      = LSF_ResvQueue(name='a1500M', memReserve=1500)
queues['a3000M']      = LSF_ResvQueue(name='a3000M', memReserve=3000)
queues['a6000M']      = LSF_ResvQueue(name='a6000M', memReserve=6000)

```

4.3.7 Defining a string-based analysis batch queue

The following example illustrates the use of the environment file parameters for string-based analysis batch queue definition:

```

try:
    queue_name=list(queue_name)
except:
    queue_name = [ ]
queue_name=queue_name + ["aba_short", "aba_long", "hold "]
after_prefix="-a"
queue_prefix="-q"
queue_cmd="qsub -nr -me %P %Q %A %T -x -eo -o %L %S"
aba_short="qsub -nr -me -q short %A %T -x -eo -o %L %S"
aba_long="qsub -nr -me -q long %A %T -x -eo -o %L %S"
hold="printf 'Job %S not submitted\n' "

```

The **qsub** command used in this example is available only on certain computer systems. Other commands, such as **at** and **batch**, can be used to configure a queuing system on most UNIX platforms. Please refer to your system documentation or contact your hardware vendor for information about queuing systems for your platform.

If the queue specified by the **queue** command line option matches one of the queue aliases in the **queue_name** parameter, that **queue** command is used in place of the default command **queue_cmd**. The following are legal command line options for the above example:

```

abacus job=qt queue=normal
abacus job=qt queue=aba_short after=10:00
abacus job=qt queue=hold

```

The first of these three command line options does not match a defined queue, so the **queue_cmd** string is used to submit the job to the **normal** queue. This queue must have been set up by the systems manager prior to submission of the job. The actual command used to send the job to the **normal** queue for execution on UNIX platforms is

```
qsub -nr -me -q normal -x -eo -o qt.log qt.com
```

The value of **%A** is not output if **after=time** is not specified on the command line.

The second option uses the string defined by **aba_short**, which submits the job to the system predefined **short** queue. The command executed by the UNIX platform is

```
qsub -nr -me -q short -a 10:00 -x -eo -o qt.log qt.com
```

The last command line option creates the file **qt.com**, which contains the Abaqus job commands, and saves it in the current directory. The message **Job qt.com not submitted** is then written to the terminal. The job is not submitted to any queue.

4.3.8 Locking out modifications to environment file parameters

In the example below, all Abaqus jobs will run in batch mode by default, and execution of Abaqus/Aqua jobs is not allowed. The inclusion of the **admin** parameter prevents modification of these settings in lower-level environment files. If this parameter is part of the environment file in the installation directory, the values of **run_mode** and **no_aqua** will override any corresponding values in a user's local directory or command line. Therefore, this example constrains all jobs submitted at your site to run in batch mode.

```
run_mode = BATCH
no_aqua = ON
admin = ['run_mode', 'no_aqua']
```

The **no_aqua** parameter would typically be used to provide a “friendly” message to users if Abaqus/Aqua is not licensed at your site.

5. Customizing Abaqus/CAE and Abaqus/Viewer

This chapter describes user interface customization, printer configuration, and graphics card tuning for Abaqus/CAE and Abaqus/Viewer.

Platform dependencies sometimes exist for Abaqus/CAE and Abaqus/Viewer. These dependencies can change between releases; therefore, they are listed on the **Support** page at www.simulia.com, where the latest information is published.

5.1 Customizing the user interface

To customize the Abaqus/CAE and Abaqus/Viewer user interface, you can specify general display properties on Windows platforms. Settings on other platforms, such as Linux, may also affect the appearance of the user interface and some of its functions.

5.1.1 Hardware acceleration (all platforms)

With some graphics devices Abaqus/CAE and Abaqus/Viewer may fail when hardware acceleration is turned on. It is possible to turn off hardware acceleration if completely necessary, although it is not recommended. Disabling hardware acceleration will severely degrade graphics performance in Abaqus/CAE and Abaqus/Viewer. You can disable hardware acceleration using one of the following methods:

Linux platforms

Start Abaqus/CAE or Abaqus/Viewer using the **-mesa** option:

```
abaqus cae -mesa
abaqus viewer -mesa
```

Windows platforms

There are two ways to disable hardware acceleration on Windows platforms:

- Add the parameter **abaqus_no_hardware_acceleration=ON** to the Abaqus environment file; or
- Create a system environment variable using the following command:

```
set ABAQUS_NO_HARDWARE_ACCELERATION=1
```

5.1.2 Common customizations on Windows platforms

The following procedures explain how to specify some commonly desired settings on Windows platforms:

To change the “start-in” location for any Abaqus shortcut:

1. Use **Windows Explorer** to go to the directory where the Abaqus shortcuts are located. The shortcuts contained in this directory affect all users on the computer and may require special permission to change.
2. Click mouse button 3 on the shortcut (**Abaqus CAE**, **Abaqus Command**, or **Abaqus Viewer**) for which you wish to change the start-in location, and select **Properties**; then click the **Shortcut** tab.
3. In the text box labeled **Start in:**, set the full path to the directory you wish to use as the default location for saving the files created by that Abaqus product.

To stop Abaqus/CAE and Abaqus/Viewer windows from being erased when a dialog box is moved:

Menu items may vary depending upon your Windows operating system. For Windows XP operating systems:

1. Select **Start**→**Settings**→**Control Panel**, and double-click **Display**.
2. From the **Display Properties** dialog box that appears, select **Appearance** and click **Effects**.
3. In the **Effects** dialog box, toggle off **Show window contents while dragging**.
4. Click **OK** to save your settings and to close the **Display Properties** dialog box.

To change the colors and fonts used in Abaqus/CAE and Abaqus/Viewer:

You can change the colors and fonts displayed in Abaqus/CAE and Abaqus/Viewer by applying a new color scheme to your session. Color schemes determine the colors and text settings that Windows uses to display each component in an application, such as its menus, dialog boxes, and title bar. For example, when the **Windows Standard** color scheme is selected, Windows displays white text in Tahoma font against a blue background in the title bar; and displays black text in Tahoma font on a white background in application menus.

You can also customize a color scheme by editing the color or text settings for any individual item in the color scheme. This customization enables you to change more specific settings, like increasing the text size in the title bar without changing the text size in other areas of the application.

Changes to the color and font settings affect all applications, not just Abaqus programs. Menu items may vary depending upon your Windows operating system. For Windows XP operating systems:

1. Select **Start**→**Settings**→**Control Panel**, and double-click **Display**.
2. From the **Display Properties** dialog box that appears, select **Appearance**.
3. Choose a preset **Color scheme** from the list.
4. If desired, choose a new **Font size** from the list.

Note: Some color schemes allow only one font size by default. To increase the font size for a particular item, edit the color scheme in the **Advanced Appearance** dialog box.

5. If desired, change the color or font settings for individual items in the selected color scheme:
 - a. Click **Advanced**.

The **Advanced Appearance** dialog box appears.
 - b. Select the **Item** for which you want to change color and font display.

The dialog box displays the current color and, if applicable, the font settings for the selected item. Windows excludes font settings for items that do not display text, like the active window border.
 - c. Choose a new item color from the **Color 1** list. For items like the active title bar that allow a gradient between two colors, you can also adjust the second color in the gradient by choosing a new color from the **Color 2** list.
 - d. Adjust the text settings for the selected item from the options at the bottom of the dialog box. You can choose a new **Font** from the list, click the **Size** arrows to increase or decrease text size, choose a new text **Color** from the list, and toggle the bold or italic formatting for the text in this item.
 - e. Repeat the previous two steps to change color and text display for other individual items in a color scheme.
 - f. Click **OK** to close the **Advanced Appearance** dialog box.
6. Click **OK** to save your settings and to close the **Display Properties** dialog box.

To change the default fonts used in Abaqus/CAE and Abaqus/Viewer:

By default, Windows renders text in the Abaqus/CAE and Abaqus/Viewer user interfaces by referring to the fonts available in your system fonts directory. You can override this default behavior and use other fonts for your session by adding the parameter **hks_font_path** to your Abaqus environment file. Set this parameter to multiple, comma-delimited values if you want to set up multiple font directories for your session.

To display Chinese characters in Abaqus/CAE and Abaqus/Viewer:

You can enable a localized Chinese version of Abaqus/CAE and Abaqus/Viewer. This localized version displays Chinese characters for text in menus, dialog boxes, the Model and Results Trees, and the message area; text in the viewport is not localized.

Menu items may vary depending upon your Windows operating system. For Windows XP operating systems:

1. Set the regional language settings for Windows to enable Chinese characters:
 - a. Select **Start**→**Settings**→**Control Panel**, and double-click **Regional and Language Options**.
The **Regional and Language Options** dialog box appears.
 - b. On the **Languages** tabbed page, ensure that the **Install files for East Asian languages** option is toggled on.
If the East Asian languages are not installed, you must have Administrator privileges to change this option.
 - c. On the **Regional Options** tabbed page, select **Chinese (PRC)** in the **Standards and formats** section.
 - d. Click **OK**.
2. Set the system environment variable **ABAQUS_USE_LOCALIZATION=1**.
 - To set the variable for a single session, enter the following command in a command prompt:

```
set ABAQUS_USE_LOCALIZATION=1
```


You must run Abaqus/CAE or Abaqus/Viewer from the same command prompt.
 - To set the variable permanently, save it in the Windows system properties. Environment variables can be saved using the **Advanced** tabbed page of the **System Properties** dialog box (select **Start**→**Settings**→**Control Panel**, and double-click **System**).

5.1.3 Linux settings that affect Abaqus/CAE and Abaqus/Viewer

Linux operating systems provide you with many options for customization. Because you can alter parts of the operating environment that are held constant on other platforms, your Linux settings may alter some basic interactions within Abaqus/CAE and Abaqus/Viewer. The exact settings available and the methods you use to change them vary according to the version of Linux that you have installed. Three possible settings and their effects are as follows:

Removal of window title bars

Your access to some dialog and toolbox functions may be limited if you have customized your system so that window title bars are not displayed. Without title bars, you may not be able to move a dialog box. Using the [Esc] key is the only means to close a dialog or toolbox that has no title bar or **Cancel** button.

Removal of window borders

Resizing a dialog box requires you to click and drag the border. If your Linux customizations include the removal of application window borders, you may not be able to resize dialog boxes in Abaqus/CAE and Abaqus/Viewer. Instead, use the scroll bars to access data that extend beyond the edges of a dialog box.

Displaying Japanese characters

If your locale setting is Japanese, Abaqus/CAE and Abaqus/Viewer can display Japanese text in the viewport. For example, text annotations and the state and title blocks are displayed in Japanese. To display the Japanese characters correctly, the Japanese TrueType fonts must be installed in the directory `/usr/lib/X11/fonts`.

5.2 Configuring printers

Print commands are not normally needed for Windows platforms. Abaqus/CAE and Abaqus/Viewer automatically detect and list any installed Windows printers. If you encounter problems using the print tool in Abaqus/CAE or Abaqus/Viewer, check the installed Windows printers on your machine and remove any printers that are no longer valid.

This section describes how Abaqus uses print commands. Print commands on Linux platforms should be identical to those used for other applications.

5.2.1 How Abaqus determines and uses the print command

When Abaqus/CAE or Abaqus/Viewer prints to a PostScript printer, it goes through the following steps:

Creating a temporary PostScript file

The temporary PostScript file contains all the PostScript code necessary to describe the page to be printed.

Retrieving the print command specified in the print dialog box

The command specified in the print dialog box can be any command that sends to the printer an unmodified copy of the file whose path is its last argument (some PostScript modifications such as the ones done by PostScript print managers are allowed).

To customize the default print command that is used by Abaqus/CAE or Abaqus/Viewer when it shows the print dialog box, add the following line to the `abaqus_v6.env` file in your home directory or in the site directory of the Abaqus release:

```
def onCaeStartup():
    session.printOptions.setValues(printCommand=
        'print_command_and_arguments_here')
```

Appending the name of the temporary file and invoking the resulting command

The name of the previously created temporary file is appended to the print command and the PostScript file is printed.

Deleting the temporary PostScript file

If the printer that you are using does not support print spooling, the temporary PostScript file may be deleted before the file is printed. To prevent the temporary PostScript files from being deleted,

add the following line to the `abaqus_v6.env` file in your home directory or in the site directory of the Abaqus release:

```
def onCaeStartup():  
    session.printOptions.setValues(deleteTemporaryFiles=False)
```

5.3 Tuning graphics cards

This section contains the information that you need to configure Abaqus/CAE and Abaqus/Viewer for a graphics adapter that is not yet qualified.

5.3.1 Why is tuning necessary?

SIMULIA tunes and qualifies a limited set of graphics adapters prior to each release. Tuning parameters for these graphics adapters are included in Abaqus. However, new graphics adapters and new drivers for existing graphics adapters become available between releases. Tuning may enable you to take advantage of these new adapters and drivers without waiting for a new release of Abaqus/CAE or Abaqus/Viewer.

Abaqus/CAE and Abaqus/Viewer use OpenGL for high-speed graphics rendering. While the OpenGL standard has strict conformance tests, some features are implementation dependent and require tuning to function correctly. Tuning a new graphics adapter or driver ensures that Abaqus/CAE and Abaqus/Viewer graphics are rendered correctly and that maximum rendering performance is obtained for each system.

You can find the latest information on qualified graphics adapters on the **Support** page at www.simulia.com. If you read the information on this page and follow the tuning procedures described in this section, you should be able to render Abaqus/CAE and Abaqus/Viewer graphics correctly and with maximum performance. If you continue to experience problems, you should contact your local technical support office for assistance.

5.3.2 How can I tune the parameters?

Abaqus/CAE and Abaqus/Viewer provide the following two methods for tuning graphics parameters:

- Select **View**→**Graphics Options** from the main menu bar. Abaqus displays the **Graphics Options** dialog box shown in Figure 5–1 from which you can select the desired settings. This approach allows you to select from only the most commonly used tuning parameters. The settings in the **Graphics Options** dialog box are described in Chapter 7, “Configuring graphics display options,” of the Abaqus/CAE User’s Manual.
- Use an Abaqus Scripting Interface command to select the desired settings. You can enter the command in the command line interface (CLI) and modify the values of the tuning parameters. This approach provides complete control of all the tuning parameters and is described in “Using

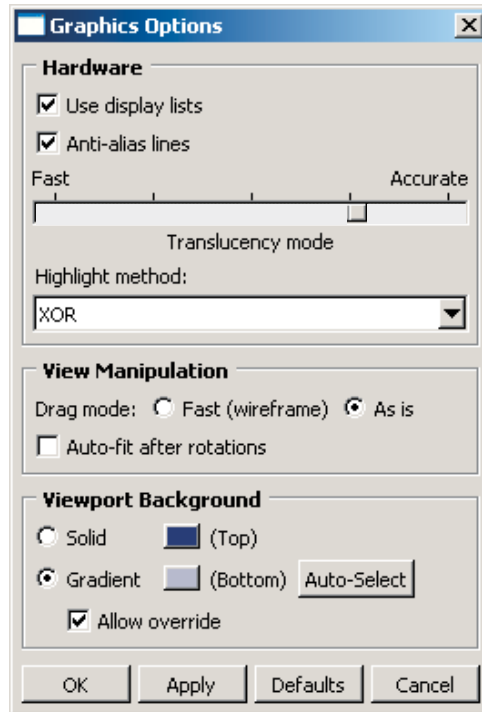


Figure 5–1 The **Graphics Options** dialog box.

the Abaqus Scripting Interface to tune the graphics parameters,” Section 5.3.3. This section also describes how you can use an Abaqus Scripting Interface command to obtain information about the graphics card that you are using.

Table 5–1 shows the tuning parameters that are available from Abaqus/CAE and Abaqus/Viewer. The table also shows the default value of each parameter and whether you can use the **Graphics Options** dialog box to modify it. Certain parameters can be modified only before starting an Abaqus/CAE or Abaqus/Viewer session (see “Making your graphics configuration permanent,” Section 5.3.4, for details on modifying parameters on startup).

Table 5–1 Tuning parameters.

Parameter	Default value	Modify using Graphics Options dialog box	Modify only on startup
<i>displayLists</i>	On	Yes	No
<i>antiAlias</i>	On	Yes	No
<i>translucencyMode</i> ²	More accurate than fast	Yes	No
<i>highlightMethod</i> ³	Hardware	Yes	No
<i>highlightMethodHint</i>	Hardware	Yes	No
<i>dragMode</i>	As-is	Yes	No
<i>autoFitAfterRotate</i>	Off	Yes	No
<i>backgroundColor</i>	#333366	Yes	No
<i>backgroundBottomColor</i>	#acacc1	Yes	No
<i>backgroundStyle</i>	Gradient	Yes	No
<i>backgroundOverride</i>	On	Yes	No
<i>doubleBuffering</i>	On	No	No
<i>polygonOffsetConstant</i>	0.0 to 100.0	No	No
<i>polygonOffsetSlope</i>	0.0 to 100.0	No	No
<i>printPolygonOffsetConstant</i>	0.0 to 100.0	No	No
<i>printPolygonOffsetSlope</i>	0.0 to 100.0	No	No
<i>textureMapping</i>	On	No	No
<i>printTextureMapping</i>	On	No	No
<i>vertexArrays</i>	On	No	No
<i>vertexArraysInDisplayLists</i>	On	No	No
<i>backfaceCulling</i>	On	No	No
<i>directRendering</i>	On	No	Yes
<i>accelerateOffScreen</i>	Off	No	Yes
<i>backingStore</i>	On	No	No
<i>hardwareAcceleration</i> ⁴	On	No	Yes
<i>hardwareOverlay</i>	None	No	Yes
<i>hardwareOverlayAvailable</i> ⁵	None	No	N/A
<i>shadersAvailable</i> ⁵	None	No	N/A

Parameter	Default value	Modify using Graphics Options dialog box	Modify only on startup
<i>viewManipDisplayListThreshold</i> ¹	40	No	No
<p>¹The threshold is only used in the Visualization module of Abaqus/CAE (Abaqus/Viewer) when display lists are enabled.</p> <p>²Translucency mode settings for rendering of translucent objects range from 1 (optimized for performance) to 5 (optimized for accuracy). The default value is 4.</p> <p>³The highlight method is indirectly set by setting the <i>highlightMethodHint</i> parameter. Abaqus uses this value to determine an appropriate setting for <i>highlightMethod</i>.</p> <p>⁴Hardware acceleration is applicable only to Windows platforms.</p> <p>⁵You cannot directly set the <i>hardwareOverlayAvailable</i> parameter or the <i>shadersAvailable</i> parameter. Abaqus automatically sets these parameters by detecting the available hardware on your system.</p>			

5.3.3 Using the Abaqus Scripting Interface to tune the graphics parameters

You can enter Abaqus Scripting Interface commands in the command line interface to tune your graphics parameters and to find out information about the graphics adapter installed on your system. This section explains how to use the Abaqus Scripting Interface to modify the graphics options; the Abaqus Scripting Interface is described in detail in the Abaqus Scripting User’s Manual.

In general, you should use the default values for most of the parameters. However, Abaqus provides the capability to modify parameters to fix the following specific problems:

- The *hardwareAcceleration* parameter controls a number of different graphics tuning parameters and generally should not be modified. Hardware acceleration options are discussed in “Hardware acceleration (all platforms),” Section 5.1.1.
- The *hardwareOverlay* parameter is controlled by the *hardwareOverlayAvailable* parameter. If your system supports hardware overlay planes, Abaqus/CAE and Abaqus/Viewer will use them by default. If your system supports hardware overlay planes but viewports display a solid color and will not display a model, you may need to manually set *hardwareOverlay*=OFF.
- Some graphics adapters do not support the use of textures to generate contour plots properly. If you experience problems displaying contour plots (for example, all contours appear gray or the system aborts), you need to set *textureMapping*=OFF to emulate texture mapping in software. Similarly, if you experience problems printing contour plots, you need to set *printTextureMapping*=OFF.
- Some graphics adapters do not fully support the use of vertex arrays to process information about vertices. Some specific problems indicate that vertex arrays are not fully supported: when you drag

the radius of a circle in the Sketcher, the circle is not visible; when you display an X - Y plot, the axis labels are not visible; and some facets in the shaded display of a mesh are missing. If you experience any of these problems, set *vertexArraysInDisplayLists*=OFF. If this does not resolve the problem, suppress the use of vertex arrays altogether by setting *vertexArrays*=OFF.

- The *backfaceCulling* parameter controls the display of facets that are determined to be facing away from the viewer. If the front sides of elements appear to be missing in the display or if the display is incomplete, set *backfaceCulling*=OFF.
- You can disable direct rendering (set *directRendering*=OFF) for Linux systems that do not behave correctly when accessing the graphics hardware directly.
- You can disable hardware-accelerated off-screen rendering (set *accelerateOffScreen*=OFF) when you want printed images to be rendered without OpenGL hardware acceleration or if you experience problems with the Probe functionality in the Visualization module of Abaqus/CAE (Abaqus/Viewer).
- You can disable the backing store (set *backingStore*=OFF) when you want to conserve memory. When *accelerateOffScreen*=ON, the memory for the backing store is allocated from memory on the graphics card. When OFF, the memory for backing store is allocated from system memory. The backing store is generated by rendering the viewport to an off-screen area. Subsequent viewport refreshes are performed more quickly by copying the off-screen area to the viewport window. Even when *backingStore*=ON, the backing store will not be created if the viewport can be redrawn sufficiently quickly.
- The *translucencyMode* parameter determines whether Abaqus/CAE optimizes the rendering of translucent objects for performance, accuracy, or for a level in between. Lower values provide better performance, while higher values provide greater accuracy.
- The *polygonOffsetConstant* and *polygonOffsetSlope* parameters, which affect onscreen display, require manual tuning for each graphics adapter. On Linux systems the *printPolygonOffsetConstant* and *printPolygonOffsetSlope* parameters can generally be set equal to the same values as the corresponding onscreen display parameters. On Windows systems the *printPolygonOffsetConstant* and *printPolygonOffsetSlope* parameters do not generally need to be adjusted.
- The *viewManipDisplayListThreshold* parameter can be lowered if there is an unacceptable delay when initiating view manipulation operations in the the Visualization module. Increasing this value may increase the delay for large models but should produce improved graphics performance during the view manipulation. If set high with a large model, the delay can be many seconds and in excessive cases may exceed system graphics memory and result in an empty display (no visible model) for the view manipulation.

You can tune the graphics parameters using the following Abaqus Scripting Interface objects:

- **GraphicsOptions**: The members of the GraphicsOptions object determine the current graphics settings. These settings can be modified during a session using the **setValues** method. The arguments to the **setValues** method are described in “setValues,” Section 15.8.1 of the Abaqus Scripting Reference Manual.

You can view the current settings of the graphics parameters by entering the following command in the command line interface:

```
print session.graphicsOptions
```

The following output is typical:

```
{'accelerateOffScreen': OFF,
  'antiAlias': ON,
  'autoFitAfterRotate': OFF,
  'backfaceCulling': ON,
  'backgroundBottomColor': '#acaccl',
  'backgroundColor': '#333366',
  'backgroundOverride': ON,
  'backgroundStyle': GRADIENT,
  'backingStore': ON,
  'directRendering': ON,
  'displayLists': ON,
  'doubleBuffering': ON,
  'dragMode': AS_IS,
  'graphicsDriver': OPEN_GL,
  'hardwareAcceleration': ON,
  'hardwareOverlay': OFF,
  'hardwareOverlayAvailable': False,
  'highlightMethod': SOFTWARE_OVERLAY,
  'highlightMethodHint': (HARDWARE_OVERLAY,
    SOFTWARE_OVERLAY, XOR, BLEND),
  'polygonOffsetConstant': 2.0,
  'polygonOffsetSlope': 0.75,
  'printPolygonOffsetConstant': 1.0,
  'printPolygonOffsetSlope': 0.75,
  'stencil': False,
  'printTextureMapping': ON,
  'shadersAvailable': True,
  'textureMapping': ON,
  'translucencyMode': 3,
  'vertexArrays': ON,
  'vertexArraysInDisplayLists': ON,
  'viewManipDisplayListThreshold': 40})
```

Note: Some of the parameters listed above have been deprecated. For information on accessing deprecated parameters, see “BackwardCompatibility object,” Section 49.4 of the Abaqus Scripting Reference Manual.

The following command uses the `setValues` method to modify some members of the `GraphicsOptions` object:

```
session.graphicsOptions.setValues(autoFitAfterRotate=ON,
    dragMode=AS_IS)
```

You can enter this command in the command line interface.

- **GraphicsInfo:** The members of the `GraphicsInfo` object provide information about the graphics adapter installed on your system. This information may be useful for determining how to tune the graphics adapter. The members are described in “GraphicsInfo object,” Section 15.7 of the Abaqus Scripting Reference Manual. The members are for reference only and cannot be modified.

You can view the graphics information by entering the following command in the command line interface:

```
print session.graphicsInfo
```

The following output is typical on Windows platforms:

```
{('glRenderer': 'Quadro FX 560/PCI/SSE2',
  'glVendor': 'NVIDIA Corporation',
  'glVersion': (2, 0, '.3'),
  'glxClientVendor': None,
  'glxClientVersion': (None, None, None),
  'glxServerVendor': None,
  'glxServerVersion': (5, 1, None)})
```

Tuning the `polygonOffsetConstant` and `polygonOffsetSlope` parameters

If display lists are enabled, you will not see the effect of tuning these parameters; therefore, you must toggle off **Use display lists** before attempting to tune your graphics adapter. Alternatively, you can enter the following command in the command line interface:

```
session.graphicsOptions.setValues(displayLists=OFF)
```

Setting drag mode to `AS_IS` is helpful for fine tuning the parameters. Rotating the view interactively will show you if minor adjustments are necessary.


```
session.graphicsOptions.setValues(dragMode=AS_IS)
```

It is recommended that you tune *polygonOffsetConstant* first, then tune *polygonOffsetSlope*. To tune these parameters, you should first display the part that is generated by the example script in “Creating a part,” Section 3.1 of the Abaqus Scripting User’s Manual. To retrieve the script, use the following command in a command prompt window (operating system shell):

```
abaqus fetch job=modelAExample
```

Select **File**→**Run Script** from the main menu bar, select the example script from the **Run Script** dialog box that appears, and click **OK**. The example script creates a new viewport; however, Abaqus should display only one viewport while you are trying to tune the graphics parameters. Select any old viewports and delete them by selecting **Viewport**→**Delete Current** from the main menu bar.

To tune the `polygonOffsetConstant` parameter:

1. From the **Views** toolbar, apply the bottom view .
2. In the command line interface, enter the following commands:

```
session.graphicsOptions.setValues(polygonOffsetSlope=0.0)
session.graphicsOptions.setValues(polygonOffsetConstant=0.0)
```

3. Display the bottom view again to refresh the display.
4. Examine the model for visible lines. If all lines are not visible, repeat Step 2, increasing the value of the polygon offset constant by a small increment; for example, **0.1**. The normal range for this parameter is between **0.5** and **1.5**, and two decimal places usually provide sufficient precision. You should attempt to find a value as small as possible that produces a correct display. Values that are too large may cause the lines to appear to float above the part. The following figures illustrate the lines that should be visible in your model.

Figure 5–2 illustrates the model with an incorrect value for *polygonOffsetConstant*; some lines are missing between the shaded areas.

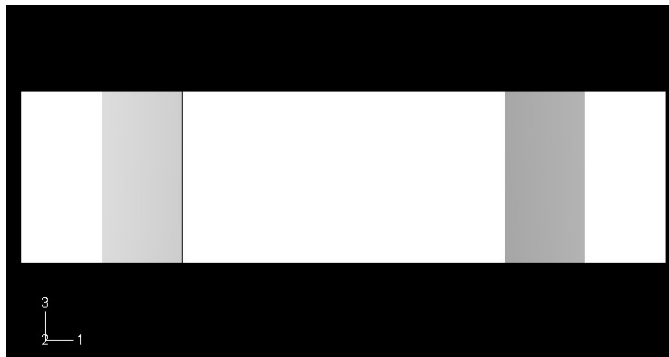


Figure 5–2 Incorrect value for *polygonOffsetConstant*.


Figure 5–3 illustrates the model with a correct value for *polygonOffsetConstant*; all the shaded areas are separated by lines.



Figure 5-3 Correct value for *polygonOffsetConstant*.

After you have derived a value for *polygonOffsetConstant*, you can tune *polygonOffsetSlope*.

To tune the *polygonOffsetSlope* parameter:

1. From the **Views** toolbar, apply the isometric view . This view shows edges at a 45° angle on at least one axis.
2. In the command line interface, enter the following command:

```
session.graphicsOptions.setValues (polygonOffsetSlope=1.0)
```

3. Examine the model for visible lines. If all lines are not visible, repeat Step 2, increasing the value of the polygon offset slope by a small increment; for example, **0.1** or **0.05**. The normal range for this parameter is between **1.0** and **2.0**, and two decimal places usually provide sufficient precision. If the *polygonOffsetConstant* value is too low, it may force the *polygonOffsetSlope* to be high. High values of *polygonOffsetSlope* may cause the edges of hidden polygons to show through where they meet visible polygons. In this case, raise the *polygonOffsetConstant* value to get an acceptable value for *polygonOffsetSlope*.

Figure 5-4 illustrates the model with an incorrect value for *polygonOffsetSlope*; some line segments are missing between the shaded areas, there is a line missing inside the hole, and some lines appear dashed when they should appear solid. Figure 5-5 illustrates the model with a correct value for *polygonOffsetSlope*; all the shaded areas are separated by solid lines.

Test the tuned values of *polygonOffsetConstant* and *polygonOffsetSlope* on several models to make sure the values are satisfactory. When you have finished tuning the graphics parameters, you should return your settings for display lists and drag mode to the original values.

When you are satisfied with the parameter settings, you should modify the environment file as described in “Making your graphics configuration permanent,” Section 5.3.4.

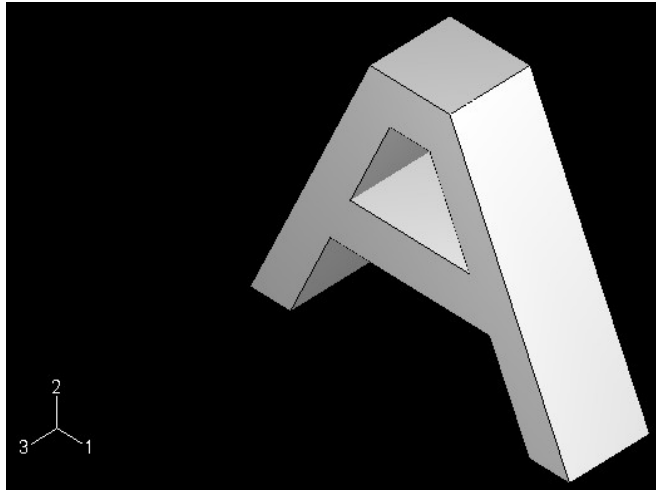


Figure 5-4 Incorrect value for *polygonOffsetSlope*.

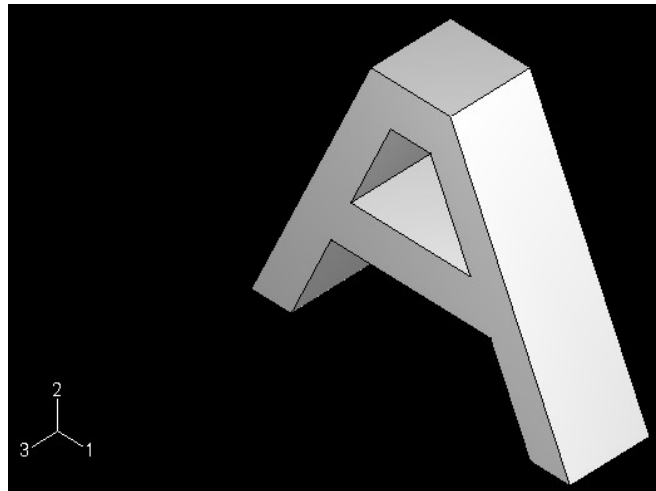


Figure 5-5 Correct value for *polygonOffsetSlope*.

5.3.4 Making your graphics configuration permanent

Once you are satisfied with the values you have specified for the tuning parameters, you can make the changes permanent by including an `onCaeGraphicsStartup` function in your environment file

(**abaqus_v6.env**). To avoid conflicts with other graphics settings, you should add the customized **onCaeGraphicsStartup** function only to the environment file in your home directory (see “Using the Abaqus environment file,” Section 4.1, for details on environment file location and execution).

The members of the **DefaultGraphicsOptions** object determine the default graphics settings that are enabled when you start a session and when you click **Defaults** in the **Graphics Options** dialog box. You can view the default graphics settings by entering the following command in the command line interface:

```
print session.defaultGraphicsOptions
```

You use the **setValues** method in the environment file (**abaqus_v6.env**) to modify the members of the **DefaultGraphicsOptions** object. To set your default graphics options in the environment file, you must use the **session.defaultGraphicsOptions** object instead of the **session.graphicsOptions** object that you modified from the command line interface. The following example environment file configures your Abaqus/CAE and Abaqus/Viewer graphics settings:

```
def onCaeGraphicsStartup():  
    session.defaultGraphicsOptions.setValues(  
            polygonOffsetConstant=1.0,  
            polygonOffsetSlope=1.2)
```

Appendix A: System requirements

A.1 Software

The system configurations on which this release of Abaqus was developed and prepared for shipment by SIMULIA are listed below. The system configurations and requirements may change between releases or maintenance deliveries. The latest system configuration information, including any additional supported platforms and the Abaqus programs available on those platforms, can be found under the **System Information** section of the **Support** page at www.simulia.com. In most cases Abaqus is compatible with later versions of the operating system software; however, it is usually not compatible with earlier versions.

Abaqus/CAE and Abaqus/Viewer are supported on a limited number of platforms. For more information, see the latest system configuration information as described above.

The following conditions apply to the system configurations:

- A FORTRAN compiler is required for compiling user subroutines but is not required for running Abaqus without user subroutines.
- Message Passing Interface (MPI) components are required to use MPI-based parallel execution in Abaqus/Standard, to use domain-level parallelization in Abaqus/Explicit, or to run any job in Abaqus/CFD (regardless of the number of CPUs). See “Visual C++ and MPI Libraries,” Section 2.4.1, and check the systems information web page mentioned above for supported MPI software and hardware interconnects.
- The latest requirements for viewing Abaqus documentation, including supported web browsers (for HTML documentation) and PDF viewers (for PDF documentation), can be found on the systems information web page mentioned above.

Table A–1 through Table A–6 list the requirements for platforms to be supported in Abaqus 6.10.

Table A–1 Windows/x86-32.

Processor	x86-32 ¹
Operating system	Windows XP Professional SP3, Windows Vista, Windows 7
Compilers	Intel Visual Fortran 10.1 and 11.1, Microsoft Visual C++ 2008 SP1 (also known as Visual C++ 9.0)
DVD	Mounted automatically.
¹ The minimum required processor is compatible with Intel Pentium IV or AMD processors supporting the SSE2 instructions set.	

APPENDIX A: SYSTEM REQUIREMENTS

Table A–2 Windows/x86-64¹.

Processor	x86-64
Operating system	Windows XP 64-bit SP2, Windows Vista 64 SP2, Windows 7 Windows Server 2003 SP2 Windows Server 2008 SP2, Windows Server 2008 R2, Windows HPC Server 2008
Compilers	Intel Visual Fortran 10.1 and 11.1, Microsoft Visual C++ 2008 SP1 (also known as Visual C++ 9.0)
DVD	Mounted automatically.
¹ This platform supports 64-bit addressing, which enables memory usage in excess of 2 GB.	

Table A–3 Linux/x86-64¹.

Processor	x86-64 ²
Operating system	SUSE Linux Enterprise 10 ³ , SUSE Linux Enterprise 10 SP2, SUSE Linux Enterprise 11, Red Hat Enterprise Linux 5, Red Hat Enterprise Linux 5.4 ⁴
Compilers	Intel Visual Fortran 10.1 and 11.1, Intel C++ 10.1 and 11.0
DVD	Mounted automatically.
¹ This platform supports 64-bit addressing, which enables memory usage in excess of 2 GB.	
² Linux/x86-64 supports both AMD Opteron and Intel EM64T processors.	
³ SUSE Linux Enterprise 10 includes Server (SLES) and Desktop (SLED) versions.	
⁴ Planned	

Table A–4 Linux/Itanium¹.

Processor	Itanium II
Operating system	SUSE Linux Enterprise 10 ² , Red Hat Enterprise Linux 5
Compilers	Intel Visual Fortran 10.1, Intel C++ 10.1
DVD	Available only by ftp download. Contact your local office or representative.
¹ This platform supports 64-bit addressing, which enables memory usage in excess of 2 GB.	
² SUSE Linux Enterprise 10 includes Server (SLES) and Desktop (SLED) versions.	

Table A-5 HP-UX/Itanium¹.

Processor	Itanium II
Operating system	HP-UX 11.23, HP-UX 11.31
Compilers	HP F90 v2.9.3 or v3.1.2, HP aC++ A.06.10 or A.06.12
DVD	Available only by ftp download. Contact your local office or representative.
¹ This platform supports 64-bit addressing, which enables memory usage in excess of 2 GB.	

Table A-6 AIX/POWER¹.

Processor	Power 4, Power 5, Power 6
Operating system	AIX 5.3 ² , AIX 6.1
Compilers	XL Fortran Compiler 11.01, IBM XL C/C++ Compiler 9.0 ³
DVD	Available only by ftp download. Contact your local office or representative.
¹ This platform supports 64-bit addressing, which enables memory usage in excess of 2 GB.	
² Patch bos.64bit 5.3.0.52 must be installed prior to installing Abaqus on AIX 5.3 systems.	
³ See Answer 4118 in the SIMULIA Online Support System (SOSS).	

A.2 Disk space

The disk space requirements to install Abaqus vary depending upon the platform on which Abaqus is installed. The Abaqus HTML plus PDF documentation installation requires approximately 2.0 GB. The Abaqus product installation requires approximately 700 MB to 1.1 GB, depending on the platform. The installation procedures provide an estimate of the disk space required for each installation option.

This disk space estimate does not include disk requirements for running analyses. Estimates of actual disk requirements are provided during the analysis input file processor stage of an Abaqus analysis. Swap space should typically be set to approximately three times the amount of physical memory installed on the computer.

Abaqus analyses can create large scratch files that are written to the directories specified by environment file parameters. These parameters can be changed from their defaults to point to file systems that are large enough to contain the scratch files. These variables are described in “Memory and disk management parameters,” Section 4.1.1.

A.3 Memory

A minimum of 1 GB of free physical memory is required to run Abaqus; at least 2 GB of free physical memory is recommended for adequate performance. If Abaqus is page faulting excessively, additional memory will reduce the page-fault rate, as well as the execution time. The requirements are highly job

APPENDIX A: SYSTEM REQUIREMENTS

dependent. Abaqus will perform better if more memory is available; to run large problems, significantly more memory than the minimum is required.

A.4 Graphics

The graphics requirements depend on the platforms on which you are running Abaqus. Abaqus/CAE and Abaqus/Viewer can be run on the following platforms:

- Windows/x86-32
- Linux/x86-64
- Windows/x86-64

A color workstation with at least 256 colors is required for Abaqus/CAE and Abaqus/Viewer. A limited set of graphics devices has been tested and qualified; this list is available under the **System Information** section of the **Support** page at www.simulia.com. For information on tuning new graphics devices between Abaqus releases, see “Tuning graphics cards,” Section 5.3.

The latest information on supported platforms and products can also be found under the **System Information** section of the **Support** page at www.simulia.com.

A.4.1 Linux platforms

X-Windows is required on Linux platforms. The OpenGL shared library is required on the computer where Abaqus/CAE and Abaqus/Viewer are run.

A.5 Compilers

A FORTRAN compiler is required to compile and link user subroutines for Abaqus. If you prefer to write user subroutines in languages other than FORTRAN, you must have the FORTRAN compiler specified, a compiler for your language, and the knowledge of how to call routines in your language from FORTRAN. SIMULIA cannot guarantee compatibility of Abaqus with user subroutines written in programming languages other than FORTRAN.

User-written postprocessing programs created with the **abaqus make** utility in Abaqus 6.10 should be compiled using the compiler for the program language and linked using a C++ compiler. Therefore, both a FORTRAN and C++ compiler are required for user-written FORTRAN programs. User-written programs that were compiled and linked with the **abaqus make** utility in a previous release of Abaqus (before Abaqus 6.10) and processing only the results (**.fil**) file can still be used with an Abaqus 6.10 results file without any change. In these instances no C++ compiler is required.

A.6 MPI

Message Passing Interface (MPI) components must be installed to use MPI-based parallel execution in Abaqus/Standard, to use domain-level parallelization in Abaqus/Explicit, or to run any job in

Abaqus/CFD (regardless of the number of CPUs). If your Abaqus users will be running these types of simulations, you must have the required MPI components preinstalled or allow the Abaqus installer to install them for you. MPI is a middleware scheme used to facilitate interprocessor communication. MPI-based parallelization is available for both computer clusters and shared memory architecture platforms. The necessary MPI software is included with Abaqus and is installed during the Abaqus product installation on most platforms; see “Visual C++ and MPI Libraries,” Section 2.4.1.

The latest information on supported platforms, MPI versions, and availability can be found under the **System Information** section of the **Support** page at www.simulia.com. For information about configuring and using MPI-based parallelization with Abaqus, refer to the SIMULIA Online Support System, which is available from the **My Support** page at www.simulia.com.

A.7 3Dconnexion motion controllers

If you are using any of the 3Dconnexion motion controllers with Abaqus/CAE, you must have Version 4.0 or later of the 3DxWare device drivers. For more information, see the 3Dconnexion web site (www.3dconnexion.com).

Appendix B: Abaqus release directories and files

The Abaqus directory structure is designed to facilitate multiple releases of Abaqus, as well as other SIMULIA products. The directory structure for Abaqus 6.10 is different from the structure used in previous releases of Abaqus; for details, see “Directory structure for Abaqus 6.7 and earlier,” Section B.3.

B.1 The SIMULIA parent directory

All files associated with Abaqus licensing and documentation installations are stored in the SIMULIA parent directory. It is recommended that Abaqus products are also installed within the SIMULIA parent directory. The suggested name for the SIMULIA parent directory is **SIMULIA**; throughout this manual, the SIMULIA parent directory is referred to as *simulia_dir* in path listings. The following directories will be created under *simulia_dir*, depending on which of the installation options are chosen:

License	Default directory for all the files associated with licensing (license file, server log file, and license options file).
installation_info	Directory containing the license startup command and log files from the licensing installation procedure and Windows uninstaller (if used).
Documentation	Default directory for all files associated with documentation.
installation_info	Directory containing the command to restart the web server and log files from the documentation installation procedure and Windows uninstaller (if used).
docs	Directory containing release-specific book files for an HTML plus PDF documentation installation.
pdf_docs	Directory containing release-specific book files for a PDF-only documentation installation.

An illustration of the SIMULIA parent directory structure is shown in Figure B–1.

B.2 The Abaqus parent directory

All files associated with an Abaqus product installation are stored in the Abaqus parent directory. It is recommended that the Abaqus parent directory be located within the SIMULIA parent directory, and the suggested path is **SIMULIA/Abaqus** (see Figure B–1). Throughout this manual, the Abaqus parent directory is referred to as *abaqus_dir*. The following directories will be created under *abaqus_dir*, depending on which of the installation options are chosen:

APPENDIX B: Abaqus RELEASE DIRECTORIES AND FILES

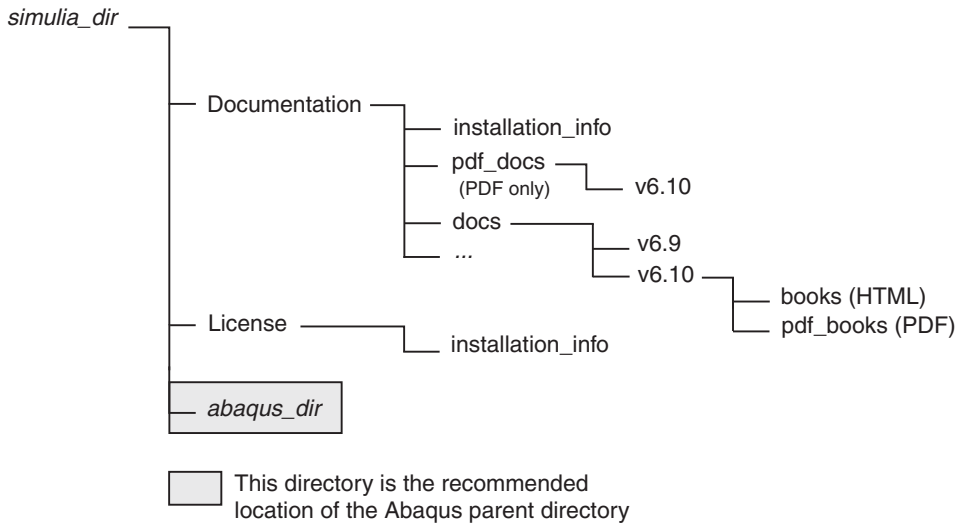


Figure B-1 The SIMULIA parent directory structure.

Commands

Directory containing links (UNIX) to, or batch files (Windows) for, the driver commands for each release.

release

Release-specific parent directory for each installed release. For example, the default directory name for Abaqus 6.10-1 is **6.10-1**.

Configuration

Directory containing configuration files.

exec

Directory containing executables and command procedures.

External

Directory containing third-party executables and libraries.

Python

Directory containing the Python interpreter and modules.

include

Directory containing header files used for building postprocessing programs with the **abaqus make** utility.

info

Directory containing information files pertaining to the Abaqus release.

installation_info	Directory containing product installation and verification log files as well as the Windows uninstaller log file (if used).
samples	Directory containing input files and directories associated with the installed products, including installation verification problems, timing test problems, files for the Introduction to Abaqus seminar, files for the Getting Started tutorial manuals, files for application briefs, files used with the abaqus findkeyword utility, and files from the Example Problems, Benchmarks, and Verification Manuals.
site	Directory containing site-specific files: the Abaqus environment file (abaqus_v6.env), a sample environment file (abaqusinc.env), include files (<i>file_name.inc</i>), a data file containing information used in the verification procedure (chksum.dat), and the platform computer tables.
abaqus_plugins	Directory containing example Python modules for Abaqus/CAE plug-ins. This directory will not be installed on platforms that do not support Abaqus/CAE.
Help	Directory containing files used for context-sensitive help in Abaqus/CAE. This directory will not be installed on platforms that do not support Abaqus/CAE.

An illustration of the Abaqus parent directory structure is shown in Figure B-2.

B.3 Directory structure for Abaqus 6.7 and earlier

Releases of Abaqus prior to Abaqus 6.8 used a directory structure different than the one currently being used. If the installation procedure detects an existing installation of Abaqus 6.7 or earlier, the new release of Abaqus will be installed using the old directory structure.

In the old directory structure, the SIMULIA parent directory does not exist. The **License** and **Documentation** directories discussed above are both installed in the Abaqus parent directory. The default location for the Abaqus parent directory in the old structure is **C:\Abaqus** on Windows platforms and the current working directory at the time of the installation on UNIX and Linux platforms.

APPENDIX B: Abaqus RELEASE DIRECTORIES AND FILES

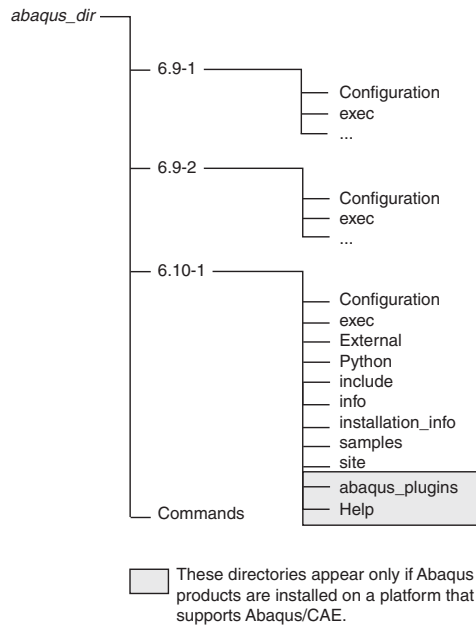


Figure B-2 The Abaqus parent directory structure.

If you are working with an old directory structure, the *simulia_dir* and *abaqus_dir* names used in the path listings throughout this manual should be considered equivalent; both names refer to the Abaqus parent installation directory in the old structure. The remaining subdirectories in the path listings have not changed.

Appendix C: Configuring Abaqus documentation

Abaqus HTML documentation is installed separately from the licensing and product installations and is viewed through a web browser. It is recommended that you install and start the Abaqus web server to serve the HTML (and PDF) files to your users' web browsers. You can install HTML documentation without a web server; however, in this case no search functionality is available. For information on the documentation installation options, see “Abaqus documentation installation details,” Section 2.2.

This appendix contains information related to configuring the web server and Abaqus HTML documentation. You must configure the Abaqus HTML documentation if you use an existing non-Abaqus web server during the HTML documentation installation procedure, if you install Abaqus products before installing Abaqus HTML documentation, or if you do not specify the URL for HTML documentation during the product installation.

For the latest information on supported web servers and configuring existing non-Abaqus web servers, see the Answers in the SIMULIA Online Support System. The SIMULIA Online Support System is accessible through the **My Support** page at www.simulia.com.

Note: When you install the Abaqus HTML documentation, the PDF files are also installed. Both the HTML and PDF versions are available through a web browser, via a link from the HTML collection home page (`index.html`) to the PDF collection index file (`index.pdf`).

C.1 Using an existing non-Abaqus web server

When you install Abaqus documentation and you choose to use an existing non-Abaqus web server, you must configure your server manually to access the HTML (and PDF) documentation. If you want to access the HTML documentation from the **Start** menu (Windows platforms only), you must modify the `documentation.htm` file after you configure the server (see “Setting the documentation URL in the `documentation.htm` file,” Section C.1.2).

C.1.1 Configuring an existing non-Abaqus web server

You can configure an existing non-Abaqus web server to access the Abaqus HTML documentation using the following procedures. For the latest information about non-Abaqus web server configurations, refer to Answer 1433 in the SIMULIA Online Support System (the SIMULIA Online Support System is accessible through the **My Support** page at www.simulia.com).

UNIX and Linux platforms

1. Install the Abaqus HTML documentation in the SIMULIA parent directory (see “Installation procedures,” Section 2.1).
2. Create an alias in your web server's configuration file for the root directory of the HTML documentation. For example, on an Apache web server you must add

```
Alias /v6.10 "simulia_dir/Documentation/docs/v6.10"
```

where *simulia_dir* refers to the HTML documentation installation parent directory.

3. Create an alias in your web server's configuration file for the Taxis executable. For example, on an Apache web server you must add

```
ScriptAlias /taxis/ "simulia_dir/Documentation/scripts/taxis/"
```

4. Restart your web server to reread the web server's configuration file.
5. Access the Abaqus HTML documentation using the following URL:

```
http://computername:port_number/v6.10
```

where *computername* is the fully qualified domain name of your web server and *port_number* is the port number that your web server is running on (*:port_number* needs to be specified only if your web server is not running on the default port of 80).

Windows platforms

1. Install the Abaqus HTML documentation in the SIMULIA parent directory. The recommended location for the SIMULIA parent directory is **C:\SIMULIA**.
2. Create an alias in your web server's configuration file for the root directory of the HTML documentation. For example, on an Apache web server you must add

```
Alias /v6.10 "simulia_dir/Documentation/docs/v6.10"
```

where *simulia_dir* refers to the HTML documentation installation parent directory.

3. Create an alias in your web server's configuration file for the Taxis executable. For example, on an Apache web server you must add

```
ScriptAlias /taxis/ "simulia_dir/Documentation/scripts/  
taxis.exe/"
```

4. Restart your web server to reread the web server's configuration file.
5. Access the Abaqus HTML documentation using the following URL:

```
http://computername:port_number/v6.10
```

where *computername* is the fully qualified domain name of your web server and *port_number* is the port number that your web server is running on (*:port_number* needs to be specified only if your web server is not running on the default port of 80).

6. To access the Abaqus HTML documentation from the **Start** menu, see the instructions in "Setting the documentation URL in the **documentation.htm** file," Section C.1.2.

C.1.2 Setting the documentation URL in the `documentation.htm` file

After you configure an existing non-Abaqus web server on Windows platforms, you can modify the following file to access the Abaqus HTML documentation from the **Start** menu:

```
simulia_dir/Documentation/installation_info/v6.10/documentation.htm
```

where *simulia_dir* refers to the HTML documentation installation parent directory. The line in the `documentation.htm` file should have the following form:

```
<META HTTP-EQUIV="Refresh" CONTENT="0";
  URL="http://computername:port_number/v6.10">
```

where *computername* is the fully qualified domain name of your web server and *port_number* is the port number that your web server is running on (*port_number* needs to be specified only if your web server is not running on the default port of 80). For example, a valid parameter definition is

```
<META HTTP-EQUIV="Refresh" CONTENT="0";
  URL="http://velouria.xyz.com:2080/v6.10">
```

C.2 Setting the documentation URL after installation

If you install Abaqus products before installing Abaqus HTML documentation or if you do not specify the URL for HTML documentation during the product installation, you must add the URL to the `abaqus_v6.env` and `url.js` files.

C.2.1 Setting the `doc_root` parameter in the `abaqus_v6.env` file

The documentation URL is required in the `abaqus_v6.env` file to provide access to the HTML documentation from Abaqus. The `abaqus_v6.env` file is located in the `abaqus_dir/6.10-n/site` directory, where *abaqus_dir* is the Abaqus parent directory and *n* refers to the Abaqus update. The required parameter definition is as follows:

- HTML and PDF documentation installation with web server

```
doc_root="http://computername:port_number/v6.10"
```

where *computername* is the fully qualified domain name of the web server and *port_number* is the port number used by the web server (if required). For example, a valid parameter definition is

```
doc_root="http://willow.xyz.com:2080/v6.10"
```

- HTML and PDF documentation installation with no web server

```
doc_root="file:///simulia_dir/Documentation/docs/
v6.10/index.html"
```

where *simulia_dir* is the HTML documentation installation parent directory. For example, valid parameter definitions are

UNIX and Linux platforms

```
doc_root="file:///usr/local/SIMULIA/Documentation/docs/
v6.10/index.html"
```

Windows platforms

```
doc_root="file:///d:/SIMULIA/Documentation/docs/
v6.10/index.html"
```

- PDF-only documentation installation (no web server). In this case you must set the values of two parameters as follows:

```
doc_root_type="pdf"
doc_root="simulia_dir"
```

where *simulia_dir* is the path to the installation base directory.

C.2.2 Setting the documentation URL in the `url.js` file

The documentation URL is required in the `url.js` file to enable links from help windows in Abaqus/CAE to associated topics in the Abaqus HTML documentation. The `url.js` file is located in the *abaqus_dir*/`6.10-n/Help/redirects` directory, where *abaqus_dir* is the Abaqus parent directory and *n* refers to the Abaqus update. The line in the `url.js` file should have the following form:

- HTML documentation install with web server

```
var baseURL="http://computername:port_number/v6.10/books/";
```

where *computername* is the fully qualified domain name of the web server and *port_number* is the port number used by the web server (if required). For example, a valid definition is

```
var baseURL="http://willow.xyz.com:2080/v6.10/books/";
```

- HTML documentation install with no web server

```
var baseURL="file:///simulia_dir/Documentation/docs/
v6.10/books/";
```

where *simulia_dir* is the HTML documentation installation parent directory. For example, valid definitions are

UNIX and Linux platforms

```
var baseURL="file:///usr/local/SIMULIA/Documentation/docs/
v6.10/books/";
```

Windows platforms

```
var baseUrl="file:///d:/SIMULIA/Documentation/docs/  
v6.10/books/";
```


Appendix D: Accessing remote file systems for installation and execution

This appendix describes accessing remote file systems for Abaqus installation and execution.

D.1 Accessing the installation files from a remotely mounted DVD

This section describes the procedure to access the installation files from a remote DVD device. The DVD device must be properly mounted. If the device does not mount automatically, check the DVD mount command for the platform on which the device exists. These commands can be found in “Software,” Section A.1.

Once you have completed the following steps, you can use the installation procedures to install Abaqus:

UNIX platforms

1. Locate the Abaqus products DVD.
2. Copy the following items from the top level of the DVD to a location that is accessible by the target computer:
 - The directory (and its entire contents) corresponding to the platform on which you are installing
 - The **common** directory (and its entire contents)
 - The **plugins** directory (applicable only for platforms that support interactive products; this directory exists only in the general release, not in maintenance deliveries)
 - The **setup** executable file

The UNIX utilities **cp** and **rcp** are convenient for copying these items. When copying files between different platforms (particularly between UNIX and Windows), special care may be needed to preserve the original file permissions and the file name case. In general, read and execute permissions on all files are sufficient for proper execution.

3. Create an Abaqus parent directory (**SIMULIA/Abaqus**) on the machine where you want to install Abaqus; see “Installation procedures,” Section 2.1, for details.
4. From the Abaqus parent directory, execute *prod_dir/setup* to launch the installation procedure, where *prod_dir* is the location of the copied installation files.

Windows platforms

You can install Abaqus from a remote DVD device if it has been defined as a share. Setting up the DVD device as a share must be performed by a user with Administrator privileges. To install

Abaqus from a shared remote DVD, connect to that drive and perform the installation as if the drive were local (see “Installation procedures,” Section 2.1). Universal Naming Convention (UNC) paths are not supported by the installation procedures.

D.2 Running Abaqus remotely

To run Abaqus remotely on Windows platforms, it is recommended that you use the client product installation described in “Client installation for Windows,” Section 2.4.4. It is also possible to run Abaqus on NFS-mounted file systems or to export the display from a remote computer; however, users may experience performance and reliability problems with these alternatives.

D.2.1 NFS-mounted file systems

There are several scenarios for running Abaqus on NFS-mounted file systems; the impact of using NFS-mounted file systems varies with the method of NFS use. The most common scenarios for running Abaqus on NFS-mounted file systems are as follows:

- Abaqus is installed on a remote file system. The CPU, save directory (see “Job variables,” Section 4.1.9), and scratch directory are local (on the computer where Abaqus will be run).

In this scenario Abaqus executables and shared libraries are loaded into local memory as they are needed across the network from the file system where Abaqus is installed. Processing and output occur locally. If there is sufficient local memory to prevent frequent paging of the code, network traffic will be relatively light. When local memory is insufficient to prevent paging, performance will suffer and reliability may be impacted.

- Abaqus is installed on a remote file system. The save directory and/or scratch directory is/are not local to the CPU where Abaqus is executed.

In this scenario program files and data used during execution are written across the network. Potentially large volumes of data will be transferred across the network, and performance may be impacted adversely. Abaqus performance may be extremely slow, and other users of the network may be affected because the network could become saturated. In addition, Abaqus does not trap file errors arising from NFS, so a failure in accessing NFS-mounted files, even temporary, will cause the Abaqus job to fail. The use of Abaqus in this configuration should be avoided whenever possible. If this configuration is required for an Abaqus/Standard job, the user should move the save directory to the NFS-mounted file system before moving the scratch directory.

D.2.2 Exporting the display

This configuration is relevant only for Abaqus/CAE and Abaqus/Viewer. In this configuration Abaqus is installed on a remote file system and the CPU, save directory, and scratch directory are located on the same remote computer, which is accessed through remote login. All processing occurs on the remote computer, and only output messages or graphics are exported to the local computer and display. This method is known to cause performance problems. Minor incompatibilities between OpenGL and GLX

libraries can introduce significant graphics problems and, in some cases, can prohibit Abaqus/CAE or Abaqus/Viewer from running. The use of Abaqus in this configuration is not supported and should be avoided whenever possible.

D.3 Using a network ODB connector

Users can create a network ODB connector to access an output database on a remote computer (see “Accessing an output database on a remote computer,” Section 9.3 of the Abaqus/CAE User’s Manual). Abaqus/CAE or Abaqus/Viewer can start the server on the remote system and assign port numbers if the following are true:

- The user name on the remote host is the same as the user name on the local system.
- The remote shell command (**rsh**) or the secure shell command (**ssh**) is configured so that it does not prompt the user for a password.

Abaqus checks the security of the connection by passing a key back and forth between the server and the client. For information about how the key is generated, see the SIMULIA Online Support System, which is accessible from the **My Support** page at www.simulia.com. If a file called **.abaqus_net_passwd** is present in the user’s home directory on the remote system, Abaqus uses the password in the file for authentication instead of the key generated by Abaqus. Abaqus checks that the user is the only account with permission to read and write to the password file. In addition, the user must update the file after 30 days, and the password must be at least eight characters long. These files are described in “Network output database file connector,” Section 3.2.19 of the Abaqus Analysis User’s Manual.

If users experience problems establishing communication or if the user names are different, they can start the network ODB server manually from the command line using the **abaqus networkDBConnector** execution procedure on the remote computer. Abaqus uses the password in **.abaqus_net_passwd** to authenticate the connection between the client and the server if the user starts the network ODB server manually. If **.abaqus_net_passwd** does not exist in the user’s directory on the remote system, the user cannot start the network ODB server from the command line.

You can disable network odb connectors by removing **dmbwtr** and **dmbwtrd** from the *abaqus_dir*/*release*/**External**/**dmbwtr** directory, where *abaqus_dir* is the Abaqus parent directory, and *release* is the release of Abaqus you are using.

Appendix E: Verification procedure

The verification procedure checks the installation of all licensed Abaqus products and reports on the success or failure of verification for each product. The verification procedure runs automatically during the Abaqus installation procedure, but only a subset of the products are verified (see the **install** option below for details). The verification procedure can be run also as an Abaqus command option and from the **Start** menu on Windows platforms at any time following the installation.

The procedure runs verification problems for each licensed product and compares the results to reference values. The command line options are not affected by license type; that is, the verification procedure attempts to verify all products named in the command. Before the verification procedure is run, licensing and system requirements are checked for the selected product(s). If a teaching academic license is detected, the verification procedure that is run during installation checks only Abaqus/Standard, Abaqus/Explicit, and Abaqus/CAE.

Product verification is skipped if the product is not licensed or if it fails to meet system requirements. The command line options **-NoLicenseCheck** and **-NoSystemCheck** bypass these checks.

The verification problems for all Abaqus products are extracted automatically from the disk during the installation. When it runs, the verification procedure attempts to create a new directory called **verify** in the current working directory. During the Abaqus installation procedure, the **verify** directory is created in the release **installation_info** directory.

To run the verification procedure from the command line:

Run the procedure by typing the following command:

```
abaqus verify [-adams -all -ams -cae -catiaV4 -catiaV5 -cfD
-contextHelp -cPerf -dcatiaV5 -design -docUrl -exp -foundation -help
-ideas -install -ioPerf -log -make -moldflow -noGui -NoLicenseCheck
-NoSystemCheck -parallel -param -parasolid -proe -retainFiles
-scripting -std -swi -user_exp -user_std -verbose -viewer]
```

Common options

all	Verify all licensed products. All other verification options except log , NoLicenseCheck , and NoSystemCheck are ignored.
help	Print summary of verify usage.
install	Verify the subset of products that are verified during installation. They include user_std (or std), user_exp (or exp), param (std or exp), and cae (or viewer). All other verification options except log , NoLicenseCheck , and NoSystemCheck are ignored.
log	Direct all output to a file named verify.log in the current working directory.

APPENDIX E: VERIFICATION PROCEDURE

Product options

ams	Verify Abaqus/AMS.
cae	Verify Abaqus/CAE.
cfld	Verify Abaqus/CFD.
design	Verify Abaqus/Design.
exp	Verify Abaqus/Explicit.
foundation	Verify Abaqus/Foundation.
param	Verify parametric studies in Abaqus.
std	Verify Abaqus/Standard.
user_exp	Verify Abaqus/Explicit with user subroutines. Required FORTRAN compiler and libraries must be installed.
user_std	Verify Abaqus/Standard with user subroutines. Required FORTRAN compiler and libraries must be installed.
viewer	Verify Abaqus/Viewer.

Translator options

adams	Verify Abaqus Interface for MSC.ADAMS.
catiav4	Verify Geometry Translator for CATIA V4.
catiav5	Verify Geometry Translator for CATIA V5.
dcatiav5	Verify Direct Geometry Import for CATIA V5 (Direct Geometry Import is a component of the CATIA V5 Associative Interface). Verifies geometry import capability in Abaqus/CAE; does not verify installation or functionality of the CATIA V5 Associative Interface plug-in.
ideas	Verify Geometry Translator for I-DEAS.
moldflow	Verify Abaqus Interface for Moldflow.
parasolid	Verify Geometry Translator for Parasolid.
proe	Verify Geometry Translator for Pro/ENGINEER (the Geometry Translator is a component of the Pro/ENGINEER Associative Interface). Verifies geometry import capability in Abaqus/CAE; does not verify installation or functionality of the Pro/ENGINEER Associative Interface plug-in.

swi Verify Geometry Translator for SolidWorks (the Geometry Translator is a component of the SolidWorks Associative Interface). Verifies geometry import capability in Abaqus/CAE; does not verify installation or functionality of the SolidWorks Associative Interface plug-in.

Additional options

contextHelp Verify Abaqus/CAE context-sensitive help functions.

cPerf Verify Abaqus/CAE performance.

docUrl Verify Abaqus HTML documentation URL.

ioPerf Verify I/O performance.

make Verify the **abaqus make** utility.

noGui Verify the **-noGUI** option for Abaqus/CAE and Abaqus/Viewer.

NoLicenseCheck Requires **all**, **install**, or a list of product options. Run the verification procedure for the specified products but bypass all licensing checks. The procedure attempts to verify all selected products regardless of licensing.

NoSystemCheck Requires **all**, **install**, or a list of product options. Run the verification procedure for the specified products but bypass all system requirement checks. The procedure attempts to verify all selected products regardless of system requirements.

parallel Verify Abaqus analysis jobs using parallelization.

scripting Verify the Abaqus scripting interface.

retainFiles Retain all verification files in the **verify** directory (by default, the files are deleted after a successful verification).

verbose Include additional details for debugging purposes.

If a product selected for verification is not licensed, an Abaqus warning is displayed for the selected product and the verification of other selected products continues.

To run the verification procedure on Windows platforms:

1. From the taskbar, click **Start**.
2. From the menu, select **Programs**→**Abaqus release**→**Verification**, where *release* corresponds to the release you wish to verify.

This procedure attempts to verify all licensed products. Use the command line option to verify individual products.

Reviewing and resolving verification procedure failures

If the verification procedure finishes successfully, all files and the **verify** directory are removed (unless you use the **retainFiles** option). Error diagnostics for all products that fail verification remain in the **verify** directory. It is very important that you review these error messages.

The following suggestions may help you to correct common installation errors that cause the verification procedure to fail:

- Make sure that the license file was installed properly. If there are problems with the license file, error messages will be written to standard output.
- Make sure that you have not tried to execute Abaqus products for which you are not licensed.
- Make sure that the operating system and compiler level are consistent with those specified for this release under **System Information** on the **Support** page at www.simulia.com.

If the error messages and these suggestions are insufficient to verify the installation, check for installation information available on www.simulia.com. Installation problems and resolutions are posted under **System Information** on the **Support** page. You can also consult the SIMULIA Online Support System for more information on installation and verification.

If you are still unable to resolve the problem, contact your local office or representative for help. Refer to the **Support Overview** page at www.simulia.com.

Appendix F: Troubleshooting Abaqus network licensing

This appendix addresses questions about network licensing. You can also consult the SIMULIA Answers database in the SIMULIA Online Support System for more information on troubleshooting network licensing. The SIMULIA Online Support System is accessible through the **My Support** page at www.simulia.com.

F.1 General questions

This section addresses some general questions about Abaqus licensing.

F.1.1 Are there limits to network size for network licensing to work?

The size of a network is limited by the requirement that all computers running Abaqus must be able to connect directly and regularly with the license server. If communication with the server takes too long, the client will spend more time communicating with the server than it does processing. Failure to communicate regularly will result in clients losing tokens and waiting while they are reacquired. The most basic test of network communication is the **ping** command. If a client and server cannot ping each other by name and IP address, network licensing will not work between the two computers. (However, the fact that the **ping** command functions properly does not necessarily guarantee that network licensing will work.)

F.1.2 Must there be a name server running for network licensing to work?

Some form of host name resolution is required to set up network licensing. If there is no NIS or DNS name server functionality on your network, then host files must be manually configured on each machine. For more information, refer to Answer 623 in the SIMULIA Online Support System, which is available from the **My Support** page at www.simulia.com.

F.1.3 How can I tell if a license file is installed?

The license file for Abaqus is usually installed in a directory called *simulia_dir*/**License** and is typically named **abaquslm.lic**. The **License** directory can be placed anywhere, but it usually resides within the SIMULIA parent directory.

F.1.4 What is the license server log file, and where should it be stored?

The server log file records all the license requests for Abaqus jobs and sessions (clients). It is stored in the Abaqus **License** directory and can be given any name you like. Usually the file is called **abaquslm.log**.

F.1.5 What is the license options file, and where should it be stored?

The options file allows the license administrator to control various operating parameters of FLEXnet Licensing, such as reserving a number of tokens for certain users, computers, or domains on a network or restricting access to Abaqus license tokens. Although it can be stored anywhere and can be given any name you like, the recommended location and name for the options file is *simulia_dir/License/ABAQUSLM.opt*. If you do not want to restrict access to Abaqus license tokens, you do not need a license options file.

After creating or modifying a license options file, you must use the **lmreread** tool to reread the license file and implement the new options (see “lmreread,” Section 3.6.6). The options file is also read when the license server is started. Some changes to the options file will not take immediate effect unless the license server is restarted.

A sample options file is shown in “Using the options file,” Section 3.7, and instructions for creating an options file are included in Chapter 5 of the FLEXnet Licensing End User Guide Version 10.8, listed under **Licensing** on the **Support** page at www.simulia.com.

F.2 License server operation

This section addresses questions that relate to the operation of the license server. The answers to these questions usually involve running one of the licensing administration tools.

F.2.1 How do I update the license file for our network license server?

To update a license file, you should terminate your license server, replace the old license file with the new license file, and restart the license server. If necessary, you can also update the license file without stopping the license server so that there will be no interruption to Abaqus users. Both techniques are described in detail in “Updating an Abaqus license file,” Section 3.2.

F.2.2 How can I tell if the license server is running?

The easiest way to see if a license server is running is to execute the command

```
abaqus licensing lmstat
```

This command will list the license servers that are running for a general broadcast. To see if a license server is running on a specific computer, you can enter the command

```
abaqus licensing lmstat -c [port]@license_server_host
```

where *license_server_host* is the hostname of the license server that you want to query. If the license server is using a port that is not between 27000 and 27009, you must specify the port number as well.

F.2.3 How can I restart the license server if it is not running?

You must login on the license server host to restart the server (on Windows platforms you must login as Administrator) and then run the **lmgrd** program (see “FLEXnet license server manager lmgrd,” Section 3.5). For an alternate method, see “Starting the server using **LMTTOOLS**,” Section 3.6.9.

F.2.4 How can I terminate an Abaqus license server?

You use the FLEXnet Licensing administration tool **lmdown** to terminate an Abaqus license server. If you use **lmdown** to terminate a license server that is part of a redundant license server configuration, all three license servers in the redundant configuration will be terminated. If you do not want to terminate all three servers, you must use an operating system utility, such as the **ps** and **kill** commands (UNIX platforms) or the Windows **Task Manager**, to terminate the license server. Do not use the **-9** option when using the **kill** command. For more information, see “lmdown,” Section 3.6.2.

F.2.5 What if the job or session loses contact with the license server before it completes?

The client enters a state of “suspended animation.” The client will attempt to reestablish contact with the license server until a user-defined expiration time is exceeded (the default is to continue checking perpetually until contact is reestablished). The minimum expiration time is 60 minutes. If the predefined expiration time has not been set, the client may be terminated by using the standard **kill** command (UNIX platforms) for the process. On Windows platforms the client can be terminated by entering **<Ctrl>-C** in the DOS window where the client was started if the job was run in interactive mode; otherwise, use the **Task Manager**.

F.2.6 What happens to licenses if the client ends abnormally?

Tokens are normally returned immediately. If the tokens are not returned immediately, ensure that the job is not running and stop the executable for the analysis by using **abaqus terminate** or the appropriate operating system utility, such as the **ps** and **kill** commands (UNIX platforms) or the Windows **Task Manager**.

Do not use the **lmremove** tool to free license tokens used for analysis jobs, as it will not have the intended effect. You can use **lmremove** to free Abaqus/CAE or Abaqus/Viewer license tokens; you must first obtain token information using the **lmstat** tool (for an example, see “lmstat,” Section 3.6.7).

F.2.7 Can I see where my job is in the license queue?

No, only log file information is available. When an analysis job is in the license queue, information about the queue status is written to the log or status file for the job every 30 seconds. Abaqus/CAE and Abaqus/Viewer interactive sessions are not queued.

F.2.8 Can I see which users are using the Abaqus licenses?

The command `simulia_dir/License/lmstat -a`, where `simulia_dir` is the SIMULIA parent directory, will list users and the license features that they are using. For more information, see “lmstat,” Section 3.6.7.

F.2.9 Do I have to install new license keys every time I install a new release of Abaqus?

Yes, you must obtain new license keys from your local office or representative when you upgrade to a new general release of Abaqus (for example, from Abaqus 6.9-3 to Abaqus 6.10-1). The new license keys will run previous releases of Abaqus that used FLEX lm or FLEXnet Licensing. Maintenance deliveries do not require new license keys within the same general release. For example, a license key obtained for use with Abaqus 6.10-2 will work with later maintenance deliveries (such as Abaqus 6.10-3, Abaqus 6.10-4, etc.) and previous releases (Abaqus 6.9-1). This license key will not work with Abaqus 6.11-1 or later.

F.2.10 Do I have to update the network license server when I install a new release of Abaqus?

Yes, if the new release of Abaqus uses a later version of FLEXnet Licensing than your existing license server. Abaqus 6.10 uses FLEXnet Licensing Version 10.8.7. If you have a version of FLEXnet Licensing earlier than Version 10.8.7 installed on your system, you should use the licensing installation procedure to automatically upgrade the license server; you must upgrade a license server that was installed with a previous Abaqus release. If you have an existing license server that uses a version of FLEXnet Licensing higher than Version 10.8.7, you should use the procedure described in “Updating an Abaqus license file,” Section 3.2, to update your license file without overwriting the existing license server. For more information, see “Licensing installation procedure,” Section 2.1.2. If your Abaqus license file is integrated with FLEXnet license files for other products using a combined license file, you must ensure that the `lmgrd` version meets the FLEXnet version compatibility rules (listed in Appendix I of the FLEXnet Licensing End User Guide Version 10.8).

You can check under **Licensing** on the **Support** page at www.simulia.com for the FLEXnet Licensing End User Guide and to see if a maintenance delivery includes a later version of FLEXnet Licensing.

F.3 Special situations

This section addresses questions concerning special operations that usually occur infrequently.

F.3.1 How do we add redundant servers if our network has a single license server?

You need to decide which other computers you wish to use as redundant license servers. The total number of redundant license servers is three, and reliable communication must be maintained at all times between the master server and at least one other backup server (shadow) for licensing to operate. The three server hosts must be on the same subnet.

1. Send the lmhostid for all three license server hosts to your local office or representative, and request a license file for redundant servers.
2. Install the redundant license file and start the Abaqus license server manually as described in “Redundant license server configurations,” Section 3.4.
3. For each installation of Abaqus, edit the Abaqus environment file (`abaqus_v6.env`) so that the line

```
abaquslm_license_file="[port]@license_server_host"
```

becomes

```
abaquslm_license_file="[port]@serverhost1: [port]@serverhost2:
[port]@serverhost3"
```

where *serverhost1*, *serverhost2*, and *serverhost3* are the names of the computers where the license servers are running (quotes are required). If a license server is using a port that is not between 27000 and 27009, you must include the port number with the server host name. Items in the list are colon-separated on UNIX platforms and semicolon-separated on Windows platforms. Each entry must begin with an “@” symbol or a port number and an “@” symbol. Make sure that users make similar changes to any environment files in their home and local directories.

F.3.2 How can I tell if I am using a redundant server configuration?

View your Abaqus license file (`abaquslm.lic`) located in the directory *simulia_dir/License*. If you are using a redundant server configuration, there will be three **SERVER** lines in the license file. The server names should be different for each line, although the port number may be the same for all three. For example:

```
SERVER bifrost.simulia.com 69084992 27003
SERVER tank.simulia.com 69094954 27003
SERVER willow.simulia.com 69094344 27003
VENDOR ABAQUSLM
```

F.3.3 What if I run the analysis job in a batch queue?

Licenses are not checked out until the analysis actually begins, so tokens will not be held by a job in a batch queue. However, if a job executes in a batch queue and sufficient tokens are not available, the job will be placed in the license queue until the tokens are free, which will prevent the rest of the jobs in the queue from executing. To prevent wasted time in the batch queue, it is recommended that tokens be registered as a resource in third-party batch queueing systems. See the LSF example for token resource in “Deriving and using custom queue classes,” Section 4.3.6.

F.3.4 What is the difference between the license queue and a batch queue?

The batch queue controls how and when jobs are run on a certain computer (see “Defining analysis batch queues,” Section 4.2). The license queue is created by the license server when Abaqus jobs have requested more tokens than are currently available. A job will not request tokens until it reaches the front of the batch queue and begins execution.

F.3.5 If we have multiple, nonredundant network license servers, how do I connect to a specific server?

The situation with multiple, nonredundant license servers is significantly different from the one with redundant servers. With redundant servers one acts as the master and the others are backups (shadows) in case the master goes down. With multiple, nonredundant servers each one is independent of the others. To connect to a specific server, you must set the **abaquslm_license_file** parameter in a local Abaqus environment file (**abaqus_v6.env**) either in your home directory or in the current working directory. The proper format for the parameter definition is as follows:

```
abaquslm_license_file="[port]@license_server_host"
```

where *license_server_host* is the name of the computer on which the license server is running (the quotation marks should be included). If a license server is using a port that is not between 27000 and 27009, you must include the port number with the server host name. Multiple servers may be listed in the **abaquslm_license_file** parameter as follows:

```
abaquslm_license_file="[port]@server1:[port]@server2:  
[port]@server3:[port]@server_n"
```

Abaqus jobs will try to connect to the servers in the order that they are listed.

F.3.6 Are there any special requirements for using FLEXnet Licensing within a DHCP-configured network?

The hostname on the server line of the license file must be the computer name, not the IP address. The client must refer to the license server by hostname in the **abaquslm_license_file** parameter in an Abaqus environment file (**abaqus_v6.env**).

F.4 Other licensing questions and problems

If you are still experiencing problems with Abaqus network licensing and you have checked all the foregoing FAQs and searched the Answers in the SIMULIA Online Support System, please refer to “Reporting Installation or Licensing Problems for Abaqus Version 6” to submit a request for systems support. This document is listed under **Status Reports** on the **Support** page at www.simulia.com. If you do not have access to www.simulia.com, you can contact your local support office to obtain a copy of the systems support request form.

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.simulia.com.

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 (DASTY). For more information, visit www.3ds.com.

Abaqus, the 3DS logo, SIMULIA, CATIA, SolidWorks, DELMIA, ENOVIA, 3DVIA, and Unified FEA are trademarks or registered trademarks of Dassault Systèmes or its subsidiaries in the US and/or other countries. Other company, product, and service names may be trademarks or service marks of their respective owners.

© Dassault Systèmes, 2010

