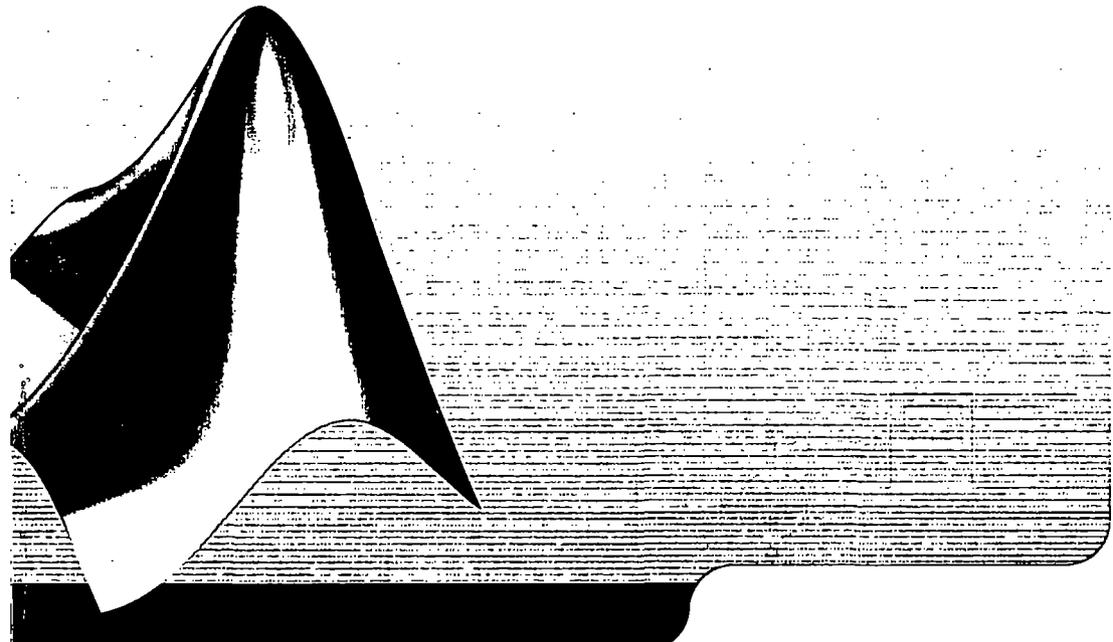


MATLAB® & SIMULINK®

Product Families

R2006b

- Computation
- Visualization
- Programming
- Modeling
- Simulation
- Implementation



Installation Guide
for Windows®

 The MathWorks

MATLAB® & SIMULINK®

Product Families

R2006b

- Computation
- Visualization
- Programming
- Modeling
- Simulation
- Implementation

Installation Guide
for Windows®

 The MathWorks

How to Contact The MathWorks:



www.mathworks.com
comp.soft-sys.matlab
www.mathworks.com/contact_TS.html

Web
Newsgroup
Technical Support



suggest@mathworks.com
bugs@mathworks.com
doc@mathworks.com
service@mathworks.com
info@mathworks.com

Product enhancement suggestions
Bug reports
Documentation error reports
Order status, license renewals, passcodes
Sales, pricing, and general information



508-647-7000 (Phone)



508-647-7001 (Fax)



The MathWorks, Inc.
3 Apple Hill Drive
Natick, MA 01760-2098

For contact information about worldwide offices, see the MathWorks Web site.

Installation Guide for Windows

© COPYRIGHT 1996–2006 by The MathWorks, Inc.

The software described in this document is furnished under a license agreement. The software may be used or copied only under the terms of the license agreement. No part of this manual may be photocopied or reproduced in any form without prior written consent from The MathWorks, Inc.

FEDERAL ACQUISITION: This provision applies to all acquisitions of the Program and Documentation by, for, or through the federal government of the United States. By accepting delivery of the Program or Documentation, the government hereby agrees that this software or documentation qualifies as commercial computer software or commercial computer software documentation as such terms are used or defined in FAR 12.212, DFARS Part 227.72, and DFARS 252.227-7014. Accordingly, the terms and conditions of this Agreement and only those rights specified in this Agreement, shall pertain to and govern the use, modification, reproduction, release, performance, display, and disclosure of the Program and Documentation by the federal government (or other entity acquiring for or through the federal government) and shall supersede any conflicting contractual terms or conditions. If this License fails to meet the government's needs or is inconsistent in any respect with federal procurement law, the government agrees to return the Program and Documentation, unused, to The MathWorks, Inc.

Trademarks

MATLAB, Simulink, Stateflow, Handle Graphics, Real-Time Workshop, and xPC TargetBox are registered trademarks of The MathWorks, Inc.

Other product or brand names are trademarks or registered trademarks of their respective holders.

Patents

The MathWorks products are protected by one or more U.S. patents. Please see www.mathworks.com/patents for more information.

Revision History

December 1996	First printing	New for MATLAB 5.0 (Release 8)
May 1997	Second printing	Revised for MATLAB 5.1 (Release 9)
March 1998	Third printing	Revised for MATLAB 5.2 (Release 10)
January 1999	Fourth printing	Revised for MATLAB 5.3 (Release 11)
November 1999	Fifth printing	Revised for MATLAB 5.3.1 (Release 11.1)
November 2000	Sixth printing	Revised for MATLAB 6.0 (Release 12)
June 2001	Seventh printing	Revised for MATLAB 6.1 (Release 12.1)
July 2002	Eighth printing	Revised for MATLAB 6.5 (Release 13) Renamed as <i>Installation Guide for Windows</i>
June 2004	Ninth printing	Revised for MATLAB 7.0 (Release 14)
October 2004	Tenth printing	Revised for MATLAB 7.0.1 (Release 14SP1)
March 2005	Eleventh printing	Revised for MATLAB 7.0.4 (Release 14SP2)
September 2005	Twelfth printing	Revised for MATLAB 7.1 (Release 14SP3)
March 2006	Thirteenth printing	Revised for MATLAB 7.2 (Release 2006a)
September 2006	Fourteenth printing	Revised for MATLAB 7.3 (Release 2006b)

Standard Installation Procedure

1

Overview of Steps in Standard Installation

Procedure	1-2
Step 1: Before You Install	1-3
Step 2: Start the Installer	1-5
Step 3: Enter Identification and License Information	1-6
Step 4: Review the Software Licensing Agreement	1-8
Step 5: Choose Installation Type	1-9
Step 6: Specify Installation Folder	1-11
Step 7: Specify the Products to Install (Custom Only)	1-12
Step 8: Specify Installation Options (Custom Only)	1-14
Step 9: Confirm Your Choices	1-15
Step 10: Read Product Configuration Notes	1-17
Step 11: Complete the Installation	1-18
Step 12: After You Install	1-20
Updating a License	1-23
Installing Additional Products	1-26
Obtaining Your Licenses	1-28
Finding the Host ID on a Windows System	1-29
System Requirements	1-30
General Requirements	1-30
Supported Platforms	1-30
Graphics	1-31
Other Requirements and Recommendations	1-31
Compilers (for MEX-Files)	1-31

Network Installation Procedure

2

Setting Up a Network Installation	2-2
Network Configuration Options	2-2
Overview of Steps in Network Installation	
Procedure	2-5
Step 1: Before You Install	2-6
Step 2: Start the Installer	2-8
Step 3: Enter Identification and License Information	2-9
Step 4: Specify Location of License File	2-11
Step 5: Review the Software Licensing Agreement	2-12
Step 6: Specify Type of Installation	2-13
Step 7: Specify Installation Folder	2-15
Step 8: Specify Products to Install	2-15
Step 9: Configure the License Manager Service	2-18
Step 10: Specify Installation Options	2-19
Step 11: Confirm Your Choices	2-20
Step 12: Read Product Configuration Notes	2-22
Step 13: Complete the Installation	2-23
Step 14: After You Install	2-25
Setting Up Network Named User Licensing	2-25
Allowing Access to License Manager Daemons	2-27
Installing MATLAB on a Client System	2-28
License File Location	2-29

License Management

3

Licensing Components	3-2
License Management Daemons	3-3
Applications	3-3
Personal License Passwords (PLPs) and License Files	3-4
License File Processing on a License Server	3-6
License File Processing on Client Systems	3-8
License File Search Path	3-8

FLEXnet License Management Utilities	3-10
Using the FLEXnet Command Line Utilities (lmutil.exe) ..	3-10
Using the FLEXnet Utilities Graphical User Interface (LMTOOLS)	3-12
Managing the License Manager Service	3-14
Installing the License Manager as a Service	3-14
Removing the License Manager Service	3-15
Starting and Stopping the License Manager	3-15
Viewing the License Manager Log File	3-17
Using Time-Outs to Free Idle License Keys	3-18
Enabling License Time-Outs	3-18
Experiencing a License Time-Out	3-20
Reacquiring License Keys After a Time-Out	3-20
Creating Advanced Configurations	3-21
Running Multiple Versions of MATLAB	3-21
Using Products in a Concurrent Installation from a Stand-Alone Installation	3-22
Running MATLAB with Other FLEXnet Applications	3-24
Configuring Redundant License Servers	3-24
Running the Installer Noninteractively	3-28
Using an Installer Initialization File	3-28
Removing (Uninstalling) an Installation	3-30
Using the Uninstaller	3-30
Running the Uninstaller Noninteractively	3-32
Removing Installations That Use Dongles	3-34

Troubleshooting

4

Problems During Installation	4-2
Installer Does Not Start Automatically	4-2
Invalid Personal License Password	4-3

License File Could Not Be Updated	4-3
Failure Overwriting an Existing Toolbox	4-4
Product Dependencies	4-4
Problems After Installation	4-5
Getting Help with License Manager Problems	4-5
License Manager Error Messages	4-6
Troubleshooting an Installation That Uses Dongles ...	4-18

Index

Standard Installation Procedure

This chapter describes how to install the MathWorks Release 2006b (R2006b) family of products on a 32-bit or 64-bit computer running the Microsoft Windows operating system. Use this procedure if you have an Individual license. With an Individual license, you can install MATLAB® on any system to which you have access, such as your office, home, and laboratory systems, or your laptop. For information about setting up the network license options, see Chapter 2, “Network Installation Procedure”.

Overview of Steps in Standard Installation Procedure (p. 1-2)

Detailed, step-by-step installation instructions

Updating a License (p. 1-23)

Update your license from a demo license without having to reinstall MathWorks products

Installing Additional Products (p. 1-26)

Add products to an existing installation

Obtaining Your Licenses (p. 1-28)

Getting license information needed for installation, if you don't already have it

System Requirements (p. 1-30)

Minimum system requirements to run MathWorks products

Overview of Steps in Standard Installation Procedure

The following table summarizes the steps required to install MathWorks products on your computer. The sections that follow provide more detailed information about each step. If you encounter any problems during installation, see Chapter 4, “Troubleshooting” for possible solutions.

Instead of performing this procedure and answering the questions posed by each dialog box, you can run the installer noninteractively. In this mode, all the information that you would normally provide using the installer dialog boxes you enter in an installer initialization file. For more information, see “Running the Installer Noninteractively” on page 3-28.

Installation Steps

Step 1: Before you install

Step 2: Start the installer

Step 3: Enter identification and license information

Step 4: Review the software licensing agreement

Step 5: Choose the installation type (Typical or Custom)

Step 6: Specify installation folder

Step 7: Specify the products to install (Custom installations only)

Step 8: Specify installation options (Custom installations only)

Step 9: Confirm your choices

Step 10: Read product configuration notes

Step 11: Complete the installation

Step 12: After you install

Step 1: Before You Install

Before you run the MathWorks Installer program:

- Have your Personal License Password (PLP) handy.
A PLP is a series of five-digit numbers prefixed with a two-digit code, such as 16-12345-12345-12345-12345. Your PLP identifies the products that you are licensed to install. When you purchase products, The MathWorks sends you a PLP by e-mail or fax. If you don't have your PLP, you can get it by clicking the **Get my PLP** button on the License Information dialog box, or by contacting The MathWorks. See "Obtaining Your Licenses" on page 1-28 for more information.
- Exit any existing copies of MATLAB you have running.
- Make sure your system satisfies the requirements of the software you intend to install. For more information, see "System Requirements" on page 1-30.
- Get administrator privileges.
- Consider disabling virus checking software and internet security applications on your system for the duration of the installation process. These applications can slow down or hang the installation process.

64-Bit Availability

The MathWorks offers both 32-bit and 64-bit versions of MATLAB. To determine whether your system is a 32-bit or 64-bit system, right-click on the **My Computer** icon on your desktop, select **Properties**, and look at the System information on the **General** tab. To download the 64-bit version of MATLAB, go to the downloads section of the MathWorks Web site, www.mathworks.com/web_downloads.

Upgrade Installations

If you are upgrading your MATLAB installation to the latest release, The MathWorks recommends that you install the new version in a new installation folder. You do not need to delete an existing MATLAB installation before installing this release. You can run multiple versions of MATLAB on the same system—see “Running Multiple Versions of MATLAB” on page 3-21. For information about adding an additional toolbox to an existing MATLAB installation, see “Installing Additional Products” on page 1-26.

Note If you choose to overwrite an existing installation from a prior release, the installer deletes all the products in your installation folder and installs only those products included in your current license, to avoid potential product incompatibilities. For example, if your existing installation includes 10 products, but your current PLP only includes 9 of those products, your MATLAB installation will only include the 9 products included in your current PLP.

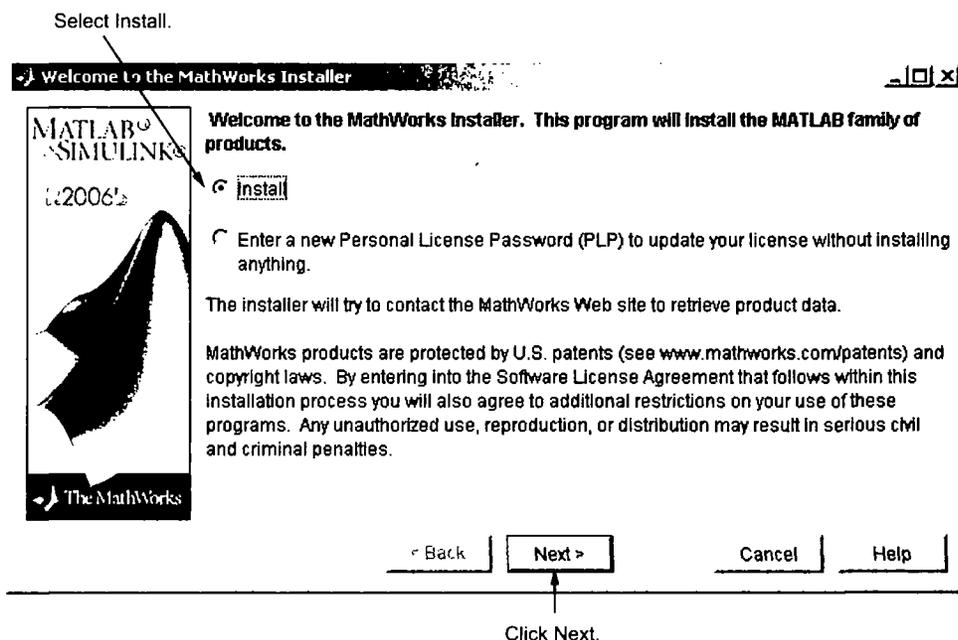
Saving MATLAB Preferences Files

The installer does not overwrite existing MATLAB preference files on your system. Note, however, that if you plan to remove a Release 12 or earlier installation, you might want to first save your MATLAB preference files. Before Release 12.1, the uninstaller automatically deleted MATLAB preferences files. See “Using the Uninstaller” on page 3-30 for more information.

Step 2: Start the Installer

Insert CD 1 into the CD-ROM drive connected to your system or double-click the installer file you downloaded from the MathWorks Web site. The MathWorks Installer starts automatically, displaying the Welcome to the MathWorks Installer dialog box. The installer will prompt you if you need to insert other CDs during installation—see “Installing Products from the Other CDs” on page 1-16.

Select the **Install** option and click **Next**. (The update license option is useful when you purchase a product after a trial and just want to update your license without reinstalling products—see “Updating a License” on page 1-23.)



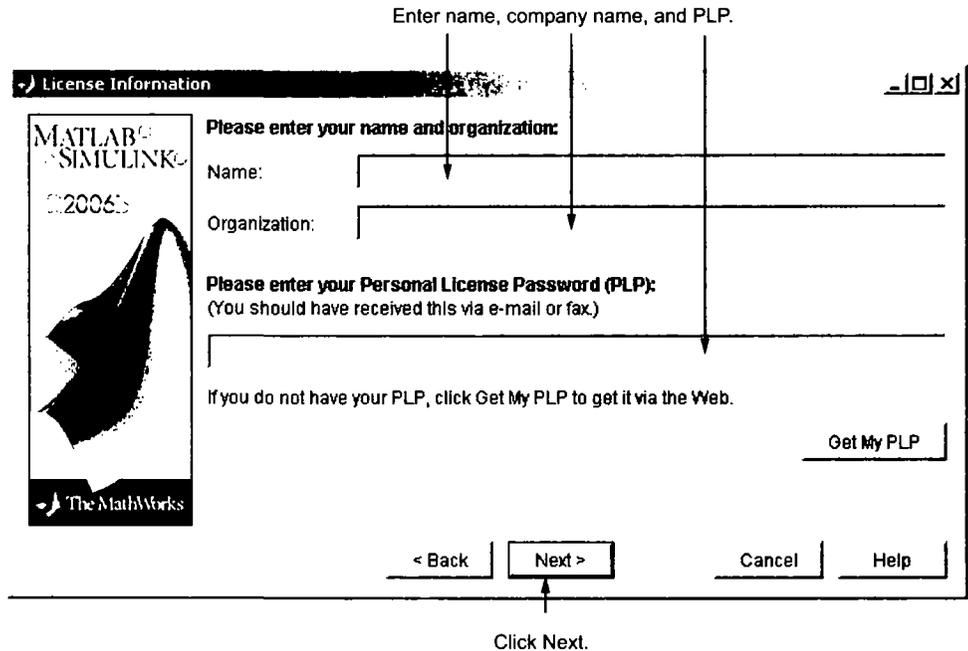
Note During installation, the installer attempts to contact the MathWorks Web site to determine if newer versions of products are available. The installer does not send any personal information from your computer to The MathWorks.

Step 3: Enter Identification and License Information

Enter your name, company name, and Personal License Password (PLP) in the License Information dialog box and click **Next**. To avoid typing errors, copy the PLP from the license e-mail and paste it into this dialog box. If your e-mail program has caused the PLP to wrap onto two lines, make sure you copy the entire PLP.

Note Do not confuse your PLP with your License Number. A PLP is a series of five-digit numbers prefixed with a two-digit code, such as 15-12345-12345-12345-12345-12345.

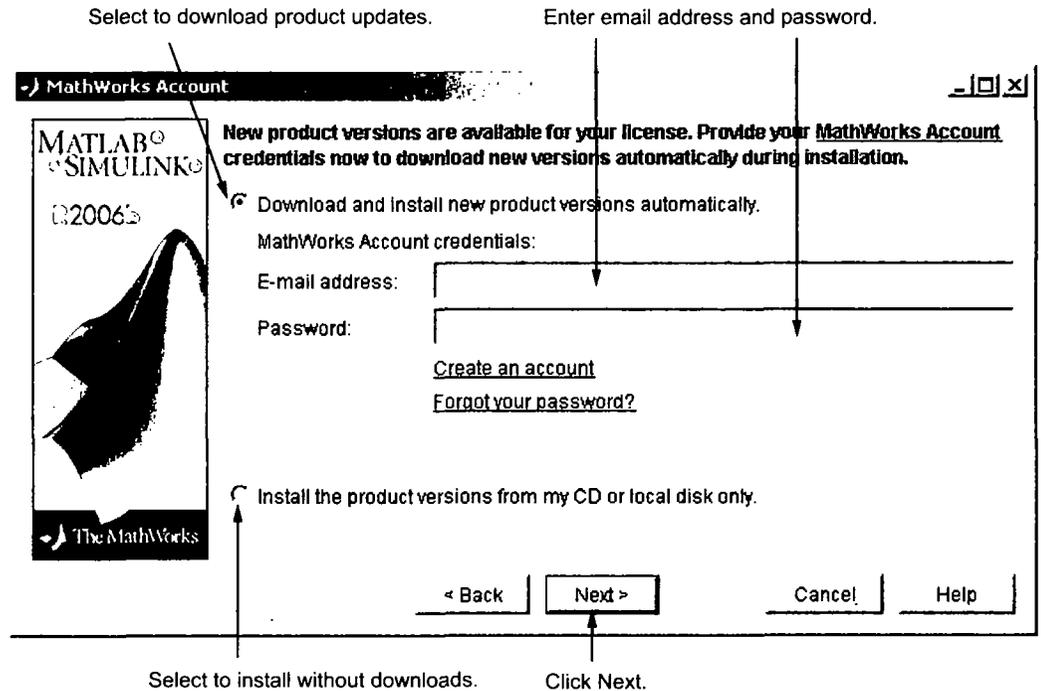
If you do not have your PLP, you can click the **Get my PLP** button to obtain it. To get your PLP this way, you must have an Internet connection and a MathWorks Account. If you do not already have an account, you can create an account at the MathWorks Web site.



Checking for Product Updates

After you enter your PLP, the installer contacts The MathWorks to see if newer versions of the products identified in your PLP are available for download. If it finds that newer versions are available, the installer displays the MathWorks Account dialog box. If you want the installer to download the latest product updates during installation, enter your e-mail address and the password for your MathWorks Account. If you do not have an account and want to learn more about setting one up, click **Create an account**.

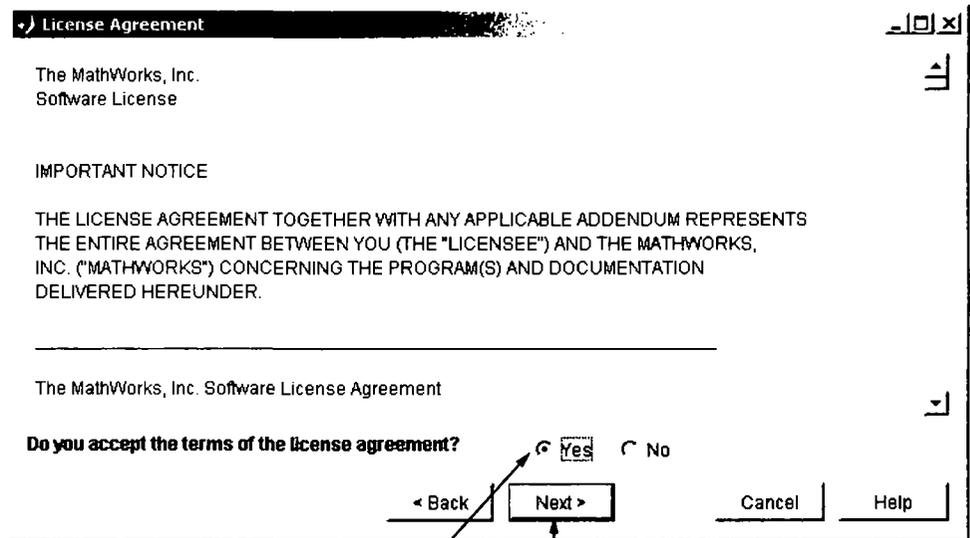
If you do not want to download these new versions and instead install only the products available on the product CD, select the **Install the product versions from my CD...** option.



Step 4: Review the Software Licensing Agreement

Review the software licensing agreement and, if you agree with the terms, select **Yes** and click **Next**.

After the installation is complete, you can view or print the license agreement using the file `license.txt` located in the top-level **MATLAB** installation folder.

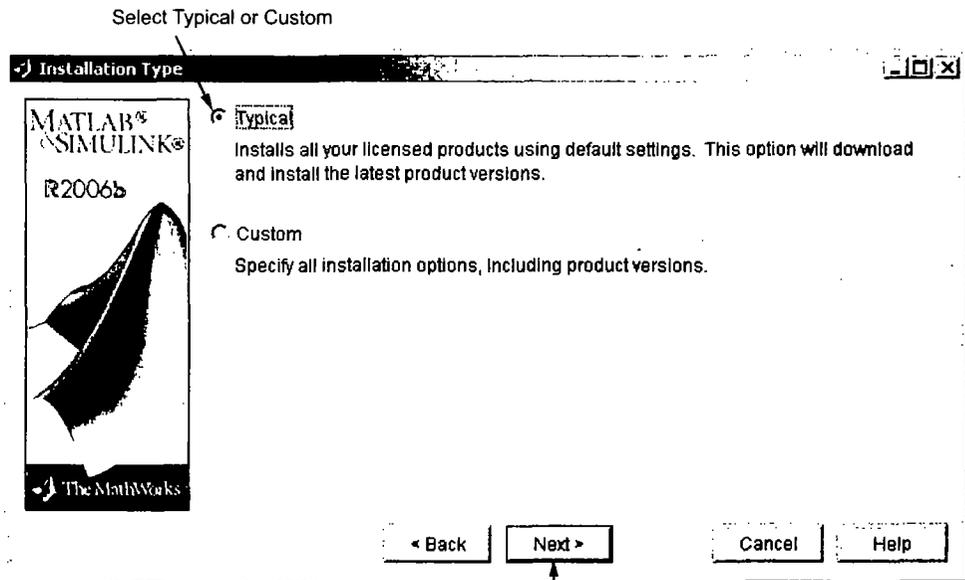


Select Yes. Click next.

Step 5: Choose Installation Type

In the Installation Type dialog box, specify whether you want to perform a Typical or Custom installation and click **Next**.

- **Typical** — Designed to simplify the installation, choose a Typical installation if you:
 - Have an Individual license.
 - Do not need to specify which products you want to install.
 - Do not need to access any installation options, such as setting the access permissions of installed files.
- **Custom** — Choose a Custom installation if you:
 - Need to specify which products to install.
 - Need access to installation options.
 - Need to install the license manager.

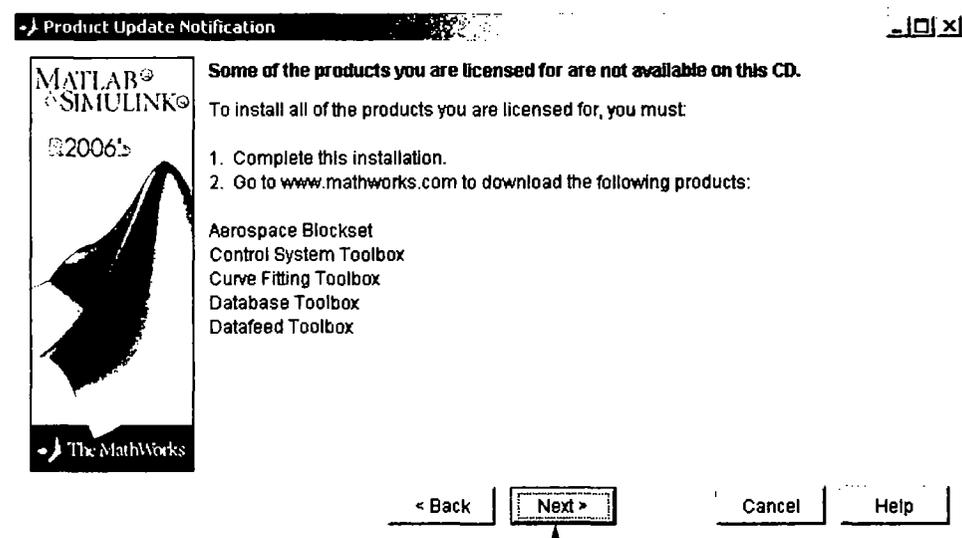


Click Next.

Product Update Information

If your PLP identifies products that are not on any of the CDs, and you are not currently connected to the Internet or you chose not to download product updates, the installer displays the Product Update Notification dialog box. This dialog box lists the names of the new or updated products that are available at the MathWorks Web site. The installer also includes this information in the Product Configuration Notes dialog box. To download these products, go to the MathWorks Web site after installation is complete.

Click **Next** to continue with the installation.

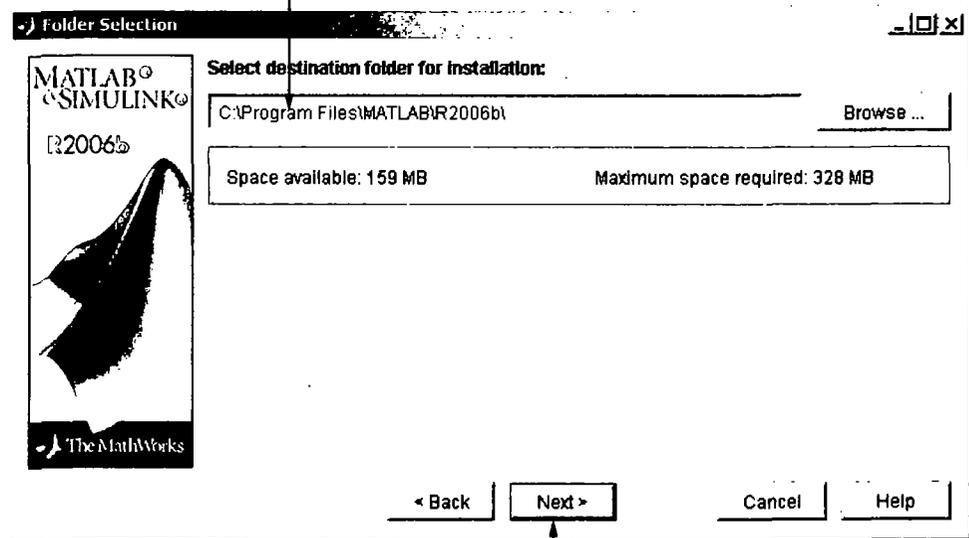


Click next..

Step 6: Specify Installation Folder

Specify the name of the folder where you want to install MATLAB in the Folder Selection dialog box. You can accept the default destination folder or specify the name of a different installation folder. If the folder doesn't exist, the installer creates it. Do not specify a folder name that contains an at (@) sign or a dollar (\$) sign, and the full path of the installation folder must not include a folder named private. Click **Next** to continue with the installation.

Specify name of installation folder.



Click Next.

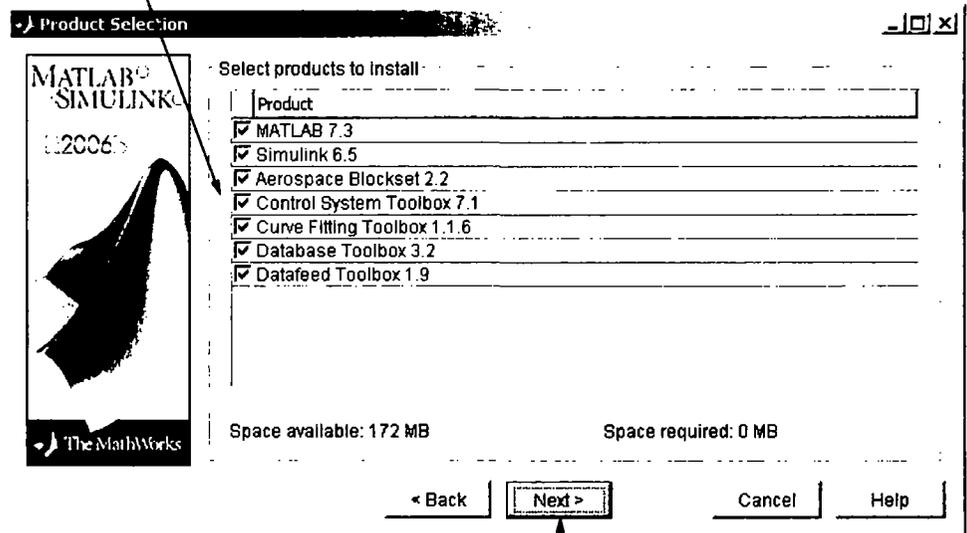
Step 7: Specify the Products to Install (Custom Only)

If you are performing a custom installation, you can specify which products you want to install in the Product Selection dialog box. This dialog box lists all the products you are licensed to install as identified by your PLP. Each product includes the online documentation, in compressed form. PDF documentation is only available at the MathWorks Web site. If you chose to download product updates, see “Product Selection When Updates Are Available” on page 1-13.

In the dialog box, all the products are preselected for installation. If you do not want to install a particular product, clear the check box next to its name. You can always install a product later—see “Installing Additional Products” on page 1-26.

After selecting the products you want to install, click **Next** to continue with the installation. After clicking **Next**, you might receive a message telling you that one or more of the products you selected depend on other products. See “Product Dependencies” on page 4-4 for more information.

Select the products you want to install.



Click Next.

Product Selection When Updates Are Available

If you chose to download product updates, the installer displays both the version that exists on the CD and the new version of the product in the product list. The updates in the product list are identified with the label **(new)**. The Product Selection dialog box also includes a column, called **Source**, that identifies whether the product is going to be downloaded during installation and provides information about the size of the download file. (At step 9, the Confirmation dialog box gives the total download size.)

By default, the latest version is preselected for installation, but you can choose which products you want to install. If you do not want to download any product updates, click the **Select local versions only** button; this deselects all the product downloads in the product list. If you want to install only the latest versions of all products, click **Select newest versions**.

After selecting the products you want to install, click **Next** to continue with the installation.

List includes both product version on CD and version available on Web.

Identifies products to be downloaded with size information.

Product	Source
<input checked="" type="checkbox"/> MATLAB 7.3	local disk
<input checked="" type="checkbox"/> Simulink® 6.5 (new)	download (78.3MB)
<input type="checkbox"/> Simulink 6.4	local disk
<input checked="" type="checkbox"/> Aerospace Blockset 2.2 (new)	download (7.7MB)
<input checked="" type="checkbox"/> Control System Toolbox 7.1 (new)	download (11 MB)
<input checked="" type="checkbox"/> Curve Fitting Toolbox 1.1.6 (new)	download (4.6MB)
<input checked="" type="checkbox"/> Database Toolbox 3.2 (new)	download (2.9MB)
<input checked="" type="checkbox"/> Datafeed Toolbox 1.9 (new)	download (1.6MB)

Space available: 483 MB Space required: 328 MB Download Size: 107 MB

Use buttons to install the latest version of all products, or only products on CD (no downloads).

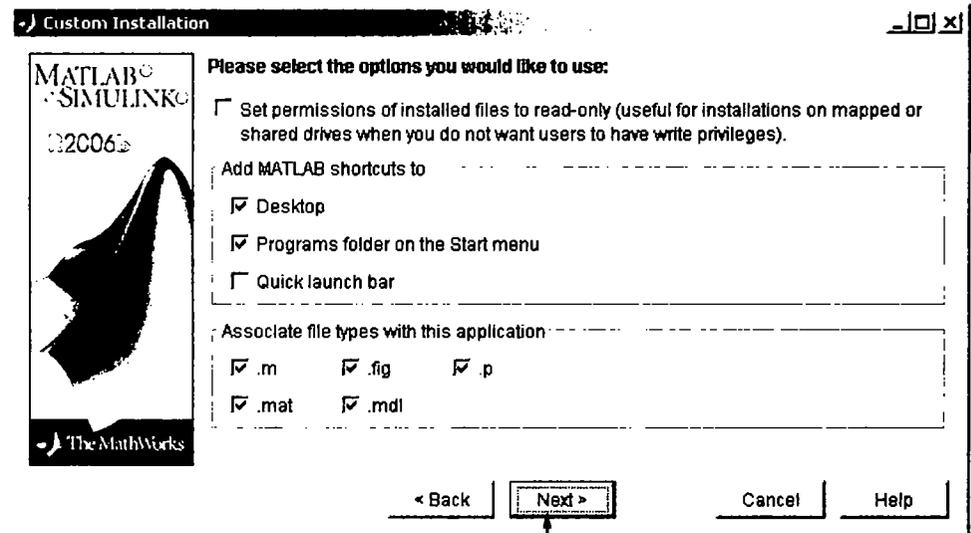
Click Next.

Step 8: Specify Installation Options (Custom Only)

For Custom installations, you can specify several installation options in the Custom Installation dialog box, including

- Setting the permissions of all installed files to read only. This option is more commonly used in network installations— see “Step 10: Read Product Configuration Notes” on page 1-17
- Determining whether the MathWorks Installer puts shortcuts for MATLAB in the **Start** menu, the **Quick launch bar**, and on the desktop
- Specifying which files the Windows operating system associates with MATLAB, based on their file extension. For example, if you associate files with the **.m** file extension with MATLAB, Windows displays MATLAB M-file as the type in the Window Explorer window.

After selecting installation options, click **Next** to proceed with the installation.

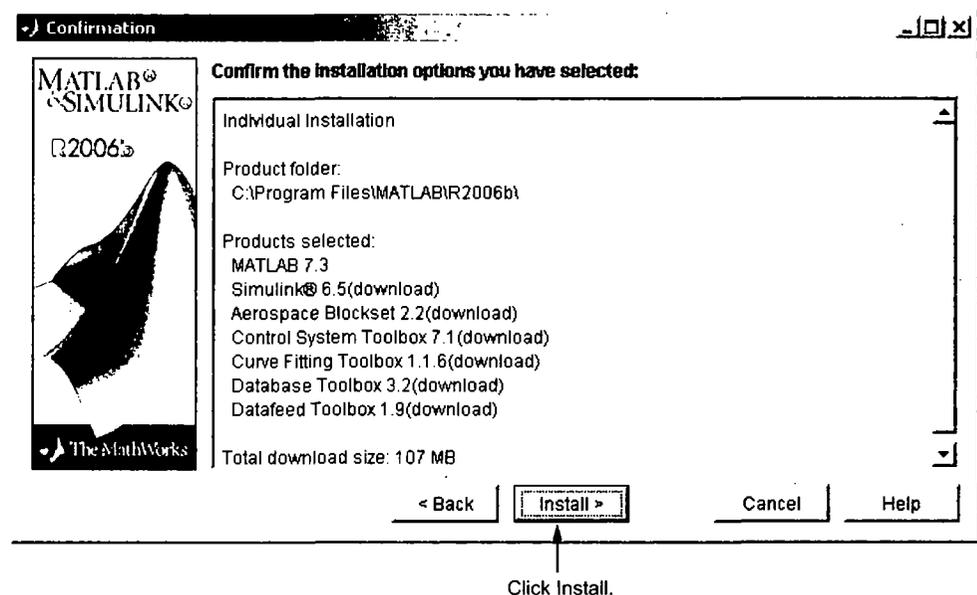


Click Next.

Step 9: Confirm Your Choices

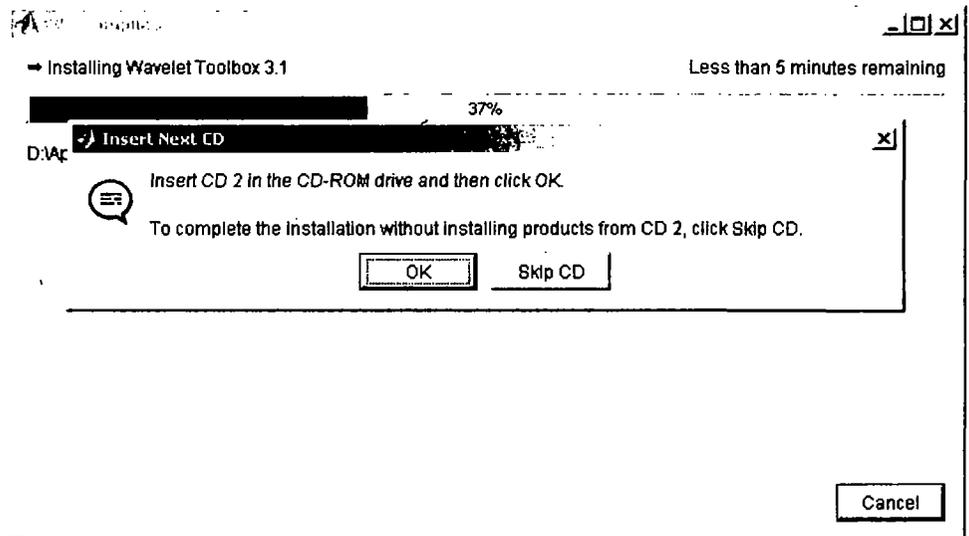
Before copying files to your hard disk, the MathWorks Installer displays a summary of the installation. To change a setting, click the **Back** button. The following example shows the confirmation dialog box for an installation that includes downloaded versions.

To proceed with the installation, click **Install**. As it copies files to your hard drive, the installer displays a status dialog box to show the progress of the installation.



Installing Products from the Other CDs

You must use CD 1 to start an installation. If you are installing products that are not on CD 1, the installer prompts you to insert another CD in your CD-ROM drive. (Depending on which products you are licensed to install, you may not need to use all of the product CDs during the installation.) After switching the CDs, click **OK** to continue with the installation. If you do not want to install these products, click **Skip CD**. You can always install these products later—see “Installing Additional Products” on page 1-26.

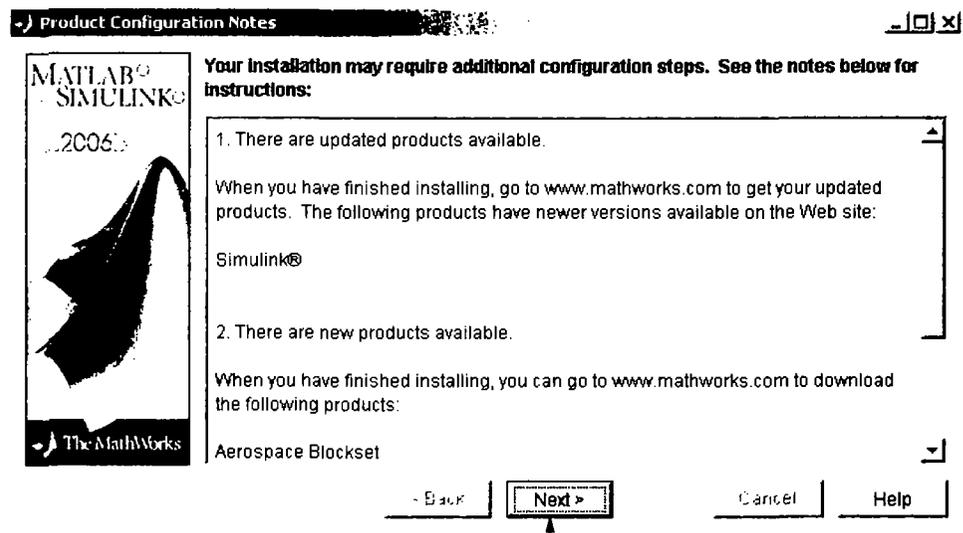


Step 10: Read Product Configuration Notes

Depending upon which products you install, the MathWorks Installer might display a dialog box containing two types of information:

- **Product configuration information** — Certain products require additional configuration. If you installed one of these products, the installer lists the configuration commands in this dialog box. You can copy this command to your system clipboard for use after the installation is completed—see “Configuring Certain Products” on page 1-21.
- **Available product updates** — If your PLP identifies products that are not on any of the CDs, and you are not currently connected to the Internet or you chose not to download product updates, the installer lists them in this dialog box. After installation is complete, you must go to the MathWorks Web site to download these products.

Click **Next** to proceed with the installation.



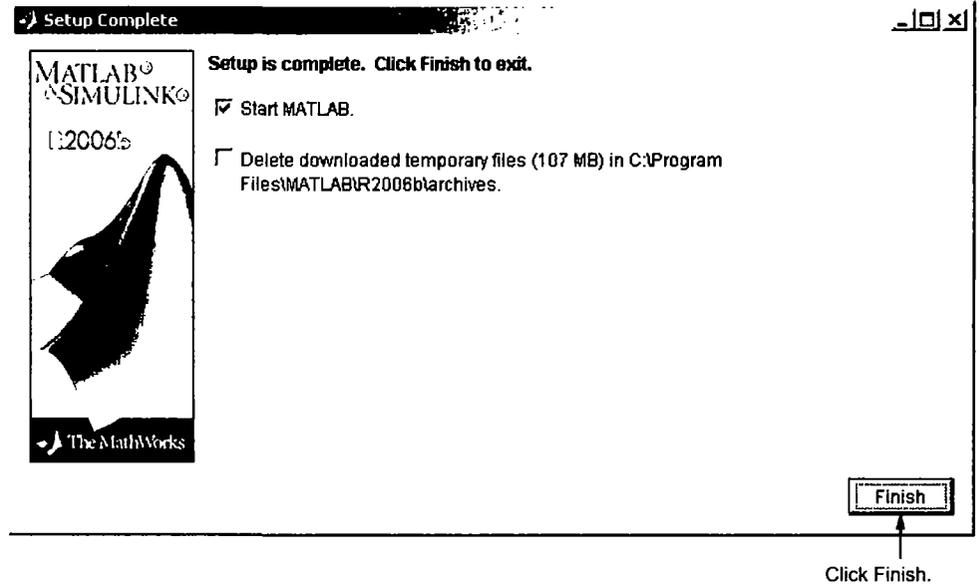
Click Next.

Step 11: Complete the Installation

When the MathWorks Installer finishes, it displays the Setup Complete dialog box. In this dialog box, you can choose to start MATLAB as you exit the installer. If you do not want to start MATLAB, clear this check box. For information about how to start MATLAB after exiting the installer, see “Starting MATLAB” on page 1-20

If the installation included product downloads, this dialog box contains a second option that lets you delete the archive files that were used for the download operation.

Select your option (or options) and click **Finish** to exit the installer.



MATLAB Top-Level Folder Structure

After a successful installation, your MATLAB installation folder contains a folder for this release. This folder contains the folders listed in the following table. (Your installation might include additional folders, depending on which products you installed.)

Folder	Description
\bin\win32	The MATLAB executable and associated files
\demos	MATLAB demonstration programs
\extern	Tools for building external interfaces to MATLAB
\help	Help system
\ja	MATLAB internationalization files
\java	MATLAB Java JAR files
\notebook	MATLAB Notebook
\sys	Tools and operating system libraries that MATLAB depends on
\toolbox	The MathWorks toolboxes
\uninstall	The MATLAB Uninstaller program
\work	Default initial current folder

Step 12: After You Install

The following are some optional tasks you can perform after a successful installation:

- “Starting MATLAB” on page 1-20
- “Setting Your Initial Current Folder” on page 1-21
- “Setting Up MATLAB Environment Options” on page 1-21
- “Configuring Certain Products” on page 1-21
- “Finding Information About MATLAB” on page 1-22

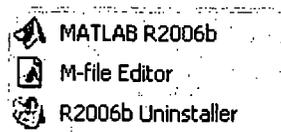
Starting MATLAB

To start MATLAB, you can use any of these methods.

- Double-click the MATLAB icon (called a *shortcut*) that the MathWorks Installer creates on your desktop.



- Click the **Start** button, select **Programs**, and move the cursor over the MATLAB entry. Select the version of MATLAB you want to start and, on the application menu that appears, click the MATLAB entry.



- Using Windows Explorer, navigate to your MATLAB installation folder. Expand the MATLAB folder, expand the R2006b folder, and then double-click the shortcut to the MATLAB executable, MATLAB R2006b.

Setting Your Initial Current Folder

By default, when you start MATLAB using the shortcut the installer puts on your desktop, the initial current folder is the *matlabroot*\work folder, where *matlabroot* represents the name of your installation folder. The \work folder is a good place to store the M-files you modify and create because the MATLAB uninstaller does not delete the \work folder if it contains files.

You can, however, use any folder as your MATLAB initial current folder. To specify another folder as your initial current folder, right-click the MATLAB shortcut that the installer creates on your desktop and select the **Properties** option. Specify the name of the folder in the **Start in** field.

Setting Up MATLAB Environment Options

To include welcome messages, default definitions, or any MATLAB expressions that you want executed every time MATLAB is invoked, create a file named *startup.m* in the *matlab*\toolbox\local folder. Every time you start MATLAB, it executes the commands in the *startup.m* file. A sample startup file, named *startupsav.m*, that you can rename and use as a starting point is included in this folder.

Configuring Certain Products

Certain products require additional configuration. The following table lists some of these products and the commands used to configure them. If you installed any of these products, see the documentation for that product for detailed configuration information.

Product	Command
MATLAB Notebook	<code>notebook -setup</code>
Real-Time Windows Target	<code>rtwintgt -install</code>

Finding Information About MATLAB

After successfully installing MATLAB, you are probably eager to get started using it. The following list provides pointers to sources of information and other features you may find helpful in getting started with MATLAB.

Task	Description
To get an overview of MATLAB and its capabilities	Read the MATLAB Getting Started documentation.
To find out what's new in this release	Read the Release Notes documentation.
To start a product or access product demonstrations programs	Use the Start button in the MATLAB desktop.
To get information about specific MATLAB features	Choose the Help item in the MATLAB menu bar to view reference and tutorial information in hyperlinked HTML form.
To get help with specific questions you can't find answered in the documentation	Go to the MathWorks Web site (www.mathworks.com), click Support , and use the Technical Support solution search area to find more information.

Updating a License

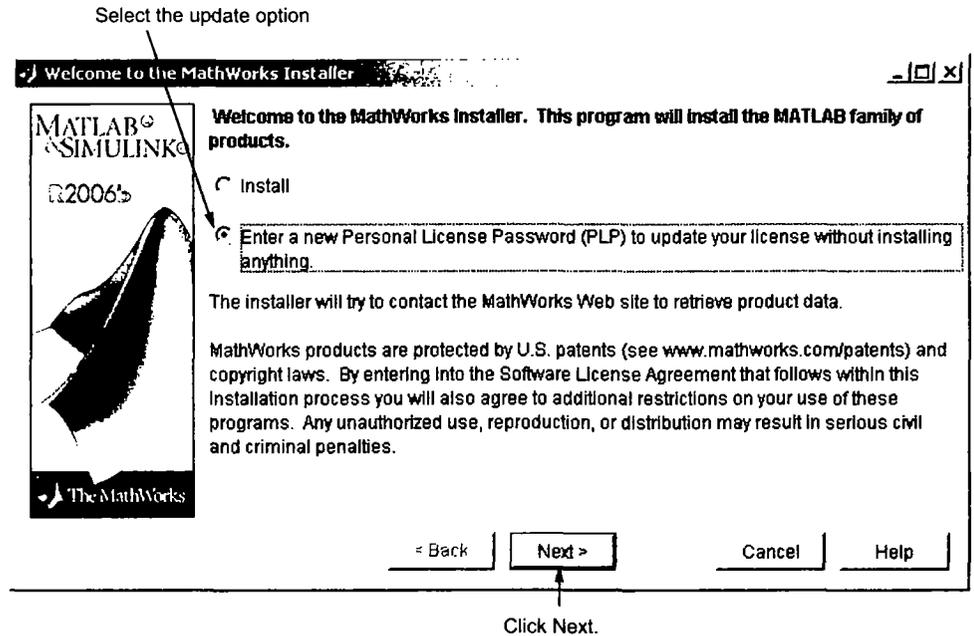
When you extend the expiration date of a demo license or convert a demo license to a standard license, The MathWorks sends you a new PLP. You must use the MathWorks Installer to update your existing installation with this new license. The installer lets you update your license without having to reinstall any products.

This section briefly outlines the procedure to update a license. For more detailed information about any step, refer to the corresponding step in “Overview of Steps in Standard Installation Procedure” on page 1-2.

1 Before you install

- Have your new PLP handy.
- Exit any existing copies of MATLAB you might have running.

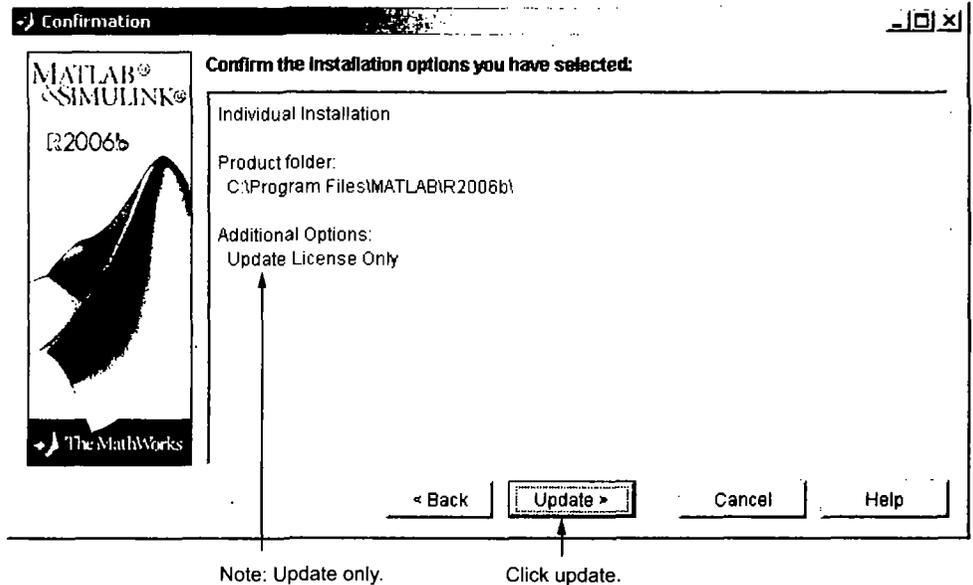
2 Insert the MathWorks CD 1 into the CD-ROM drive connected to your system or double-click the installer file you downloaded from the MathWorks Web site. The MathWorks Installer starts automatically, displaying the Welcome to the MathWorks Installer dialog box. To update a license, select the **Enter a new Personal License Password...** check box and click **Next**.



- 3 Enter your new Personal License Password (PLP) in the License Information dialog box and click **Next**. You cannot edit the **Name** and **Company** fields when updating a license.
- 4 Review the software licensing agreement and, if you agree with the terms, select **Yes** and click **Next**.
- 5 Specify the location of your existing MATLAB installation in the Folder Selection dialog box and click **Next**.

- 6 Verify your choices in the Confirmation dialog box. Note that the **Additional Options** section specifies **Update License Only**. To proceed with the license update, click **Update**.

Note When you click **Update**, the installer updates your license, but does not install any products.



- 7 When the installer finishes, it displays the Setup Complete dialog box. In this dialog box, you can choose to start MATLAB as you exit the installer. Click **Finish** to exit the installer.

Installing Additional Products

When you order additional products, The MathWorks sends you a new Personal License Password (PLP). You must use this new PLP when you install your new products.

This section briefly outlines the procedure to install an additional product. For more detailed information about any step, see “Overview of Steps in Standard Installation Procedure” on page 1-2.

1 Before you install:

- Have your new PLP handy.
- Exit any existing copies of MATLAB you might have running.

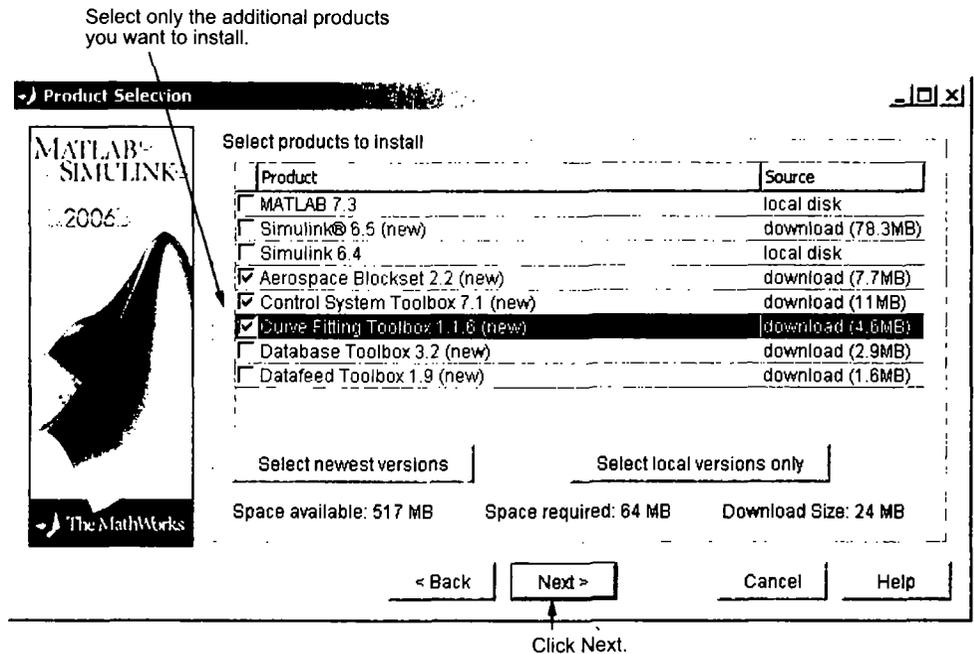
2 Insert the MathWorks CD 1 into your CD-ROM drive or double-click the file you downloaded from the MathWorks Web site. The installer starts automatically, displaying the Welcome to the MathWorks Installer dialog box. Select the **Install** option and click **Next**.

3 Enter your new Personal License Password (PLP) in the License Information dialog box and click **Next**.

4 Review the software licensing agreement and, if you agree with the terms, select **Yes** and click **Next**.

5 Select the **Custom** option on the Installation Type dialog box. If you want to install only your new products, you must choose the **Custom** option because it enables you to select which products you want to install. Click **Next** to continue.

6 Specify the location of your existing MATLAB installation in the Folder Selection dialog box. By default, the MathWorks Installer lists all the products you are licensed to install and preselects them for installation. To only install the new products, clear the check boxes next to the names of the products that you do not want to install.



7 If you are not installing MATLAB, you can ignore the Custom Installation dialog box. Click **Next** to proceed.

8 Verify your choices in the **Confirmation** dialog box. To proceed with the installation, click **Install**.

As it copies files to your hard drive, the installer displays a status dialog box to show the progress of the installation. If you are installing products that are not on CD 1, the installer prompts you to insert another CD in your CD-ROM drive.

9 Depending on which products you installed, the installer might display the Product Configuration Notes dialog box. Copy any configuration commands for later use. See “Configuring Certain Products” on page 1-21 for more information.

10 Click **Finish** in the Setup Complete dialog box to exit the installer.

Obtaining Your Licenses

If you need to

- Obtain your licenses
- Transfer a license to another system
- Combine licenses
- Move toolboxes from one license to another
- Change a license registration
- Update your contact information

go to the MathWorks Web site at www.mathworks.com and click **Support**. On the support page, under **Installation and Licensing**, click the **Get My Passcode (PLP)/License File** link. To get other kinds of licensing help, click **Customer Service Options**.

You can also obtain your PLP by contacting The MathWorks via telephone at 508-647-7000. Customers outside North America should contact their local office. To find the phone number of one of our international offices, go to the MathWorks Web site at www.mathworks.com and click **Contact Us** at the top of the page.

Depending on your request, please have ready, or include in your e-mail or fax, these items:

- Your License Number.

If you have not previously installed MATLAB at your site, you can find your License Number on the upper-right corner of the packing slip.

If you have a previous version of MATLAB running at your site, you can obtain your License Number by entering the license command or the ver command at the MATLAB command prompt.

- Your e-mail address.
- The host ID of the server on which you intend to run the license manager (network installations only). To learn how to get a host ID, see “Finding the Host ID on a Windows System” on page 1-29.

Finding the Host ID on a Windows System

For network installations, you need to supply the host ID of the server on which you intend to run the license manager. On PCs running the Windows operating system, you can use the Ethernet address (Physical Address) or the IP address as the host ID. If you use the Ethernet address remove the dashes (-).

To get this information, open a command prompt window and run the `ipconfig` command.

```
ipconfig /all
```

Note Be sure to get the host ID of the system on which you plan to run the license manager, not the host ID of a client workstation.

System Requirements

Note For the most up-to-date information about system requirements, go to the MathWorks Web site (www.mathworks.com) and click **Support**. On the Support page, click **System Requirements**.

General Requirements

- CD-ROM or DVD drive (for installation)
- Some license types require a license server running FLEXnet 10.8.0.1, which is provided by the MathWorks Installer.
- TCP/IP is required on all platforms when using a license server.
- An available USB port, for licenses that require a hardware lock.

Supported Platforms

32-bit Windows

Operating System	Processors	Disk Space	RAM
Windows XP (Service Pack 1 or 2)	Intel: Pentium III	460 MB (MATLAB only with Help)	512 MB (1024 recommended)
Windows 2000 (Service Pack 3 or 4)	Pentium IV		
Windows Server 2003	Pentium Xeon Pentium M AMD Athlon AMD Athlon MP AMD Athlon XP AMD Opteron		

64-bit Windows

Operating System	Processors	Disk Space	RAM
Windows XP x64	Intel: EM64T processors AMD: AMD64 processors	460 MB (MATLAB only with Help)	512 MB (1024 recommended)

Disk space requirements vary depending on size of partition. The MATLAB installer will inform you of the hard disk space requirement for your particular partition. Installation size is determined by the installer and can vary for NTFS and FAT formats.

Graphics

- 16-, 24-, or 32-bit OpenGL capable graphics adapter

Other Requirements and Recommendations

- Microsoft Windows supported graphics accelerator card, printer, and sound card
- Microsoft Word 2000, 2002, or 2003 is required to run MATLAB Notebook
- Microsoft Excel 2000, 2002, or 2003 is required to run MATLAB® Builder for Excel® and Excel® Link.

Compilers (for MEX-Files)

For the most up-to-date information about compilers, go to the MathWorks Web site (www.mathworks.com) and click **Support**.

Network Installation Procedure

This chapter describes how to install the MathWorks Release 2006b family of products on a 32-bit or 64-bit computer running the Microsoft Windows operating system. Use this procedure for license options that can be installed in a network configuration. For information about performing stand-alone installations of MathWorks products for individual licenses, see Chapter 1, “Standard Installation Procedure”.

Setting Up a Network Installation
(p. 2-2)

Different types of configurations you
can set up with network licenses

Overview of Steps in Network
Installation Procedure (p. 2-5)

Detailed, step-by-step installation
instructions

Installing MATLAB on a Client
System (p. 2-28)

Differences between installing
MATLAB on a client system and on
a server

Setting Up a Network Installation

For network installations, The MathWorks uses a license manager program, FLEXnet Publisher Licensing Module, a product of Macrovision Corporation. To set up a network installation, you must install the license manager as well as the MathWorks products you purchased. You use the MathWorks Installer to perform both installations.

Before you install MATLAB in a network installation, you must decide what type of configuration you want to set up. This decision determines how many installations you need to perform. This section describes these options.

Using the MathWorks Installer, you can set up a network installation with little knowledge of license management software. However, if you want to integrate the MathWorks products into an already established license management scheme, you may need a more thorough understanding of product licensing. For an overview of how The MathWorks uses FLEXnet, see Chapter 3, “License Management”.

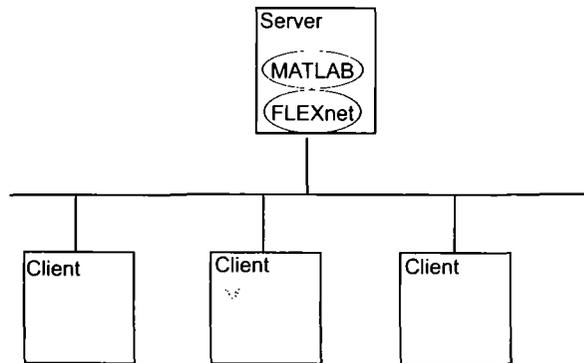
Network Configuration Options

You can set up a MATLAB network installation in two configurations:

- “Remote Client Access Configuration” on page 2-3
- “Local Client Access Configuration” on page 2-4

Remote Client Access Configuration

In a *remote client access* configuration, you install MATLAB and the license manager on a central file server. (These instructions refer to this system as your *license server*.) Users on client systems access MATLAB over the network. This figure shows one possible configuration. The license manager (FLEXnet) and MATLAB can be installed on the same server or on separate servers.

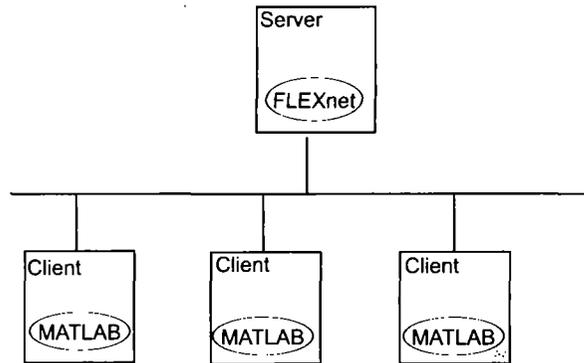


If your license server is a PC, follow the instructions in “Overview of Steps in Network Installation Procedure” on page 2-5. If your license server is a UNIX system, follow the instructions in the installation documentation for UNIX systems. A UNIX license server can serve PC clients and a PC server can serve UNIX clients.

In remote client access configurations, it can be helpful to give clients a script that sets up a shortcut to the network copy of MATLAB and specifies a default startup folder for MATLAB. See “Licensing Components” on page 3-2 for more setup suggestions.

Local Client Access Configuration

In a *local client access* configuration, you install the license manager on a server and you install MATLAB on each client system. When users on client systems start MATLAB, it contacts the license manager over the network to check out a license key. This figure shows one possible configuration.



You use the same basic network installation procedure to install MathWorks products on a client system. However, there are some differences. See "Installing MATLAB on a Client System" on page 2-28 for complete information.

Overview of Steps in Network Installation Procedure

The following table summarizes the steps in the network installation procedure. The sections that follow provide more detailed information about each step. If you encounter any trouble during installation, see Chapter 4, “Troubleshooting” for answers to some common installation problems.

You use this procedure for installations in both remote access and local access configurations. For either configuration, you must first install the license manager on your license server. During this installation, the MathWorks Installer performs some necessary processing of your License File. You will use this processed License File for all subsequent client installations. See “Installing MATLAB on a Client System” on page 2-28 for more information. For remote access configurations, you must also install MATLAB on your license server. You can install MATLAB and the license manager at the same time.

Instead of performing this procedure, and answering the questions posed by each dialog box, you can run the installer noninteractively. In this mode, all the information you would normally provide using the installer dialog boxes you enter in an initialization file. For more information, see “Running the Installer Noninteractively” on page 3-28.

Network Installation Procedure

- Step 1: Before you install
- Step 2: Start the installer
- Step 3: Enter identification and license information
- Step 4: Specify location of License File
- Step 5: Review the software licensing agreement
- Step 6: Specify type of installation (must choose Custom)
- Step 7: Specify installation folder
- Step 8: Specify the products you want to install
- Step 9: Configure the license manager as a service
- Step 10: Specify installation options

Network Installation Procedure

Step 11: Confirm your choices

Step 12: Read product configuration notes

Step 13: Complete the installation

Step 14: After you install

Step 1: Before You Install

Before you run the installation program:

- Have your Personal License Password (PLP) and your License File handy. A PLP identifies the products that you are licensed to install; a License File identifies the products you can run. The licensing information you receive varies depending on the type of license option you choose.

A PLP is a series of five-digit numbers prefixed with a two-digit code, such as 15-12345-12345-12345-12345. Depending on your license option, you might receive a PLP for stand-alone installations as well as a PLP for network installations.

A License File is a specially formatted text file that contains an INCREMENT line for each licensed product. See “Licensing Components” on page 3-2 for an example.

If you don't have your licensing information, see “Obtaining Your Licenses” on page 1-28.

- Exit any existing copies of MATLAB you may have running.
- Shut down the license manager, if it is running. (See “Starting and Stopping the License Manager” on page 3-15 for instructions.)
- Make sure your system satisfies the requirements of the software you intend to install. For more information, see “System Requirements” on page 1-30.
- Get administrator privileges for the system on which you plan to install MATLAB.

- Consider disabling virus checking software and internet security applications on your system for the duration of the installation process. These applications can slow down or hang the installation process.

64-Bit Availability

The MathWorks offers both 32-bit and 64-bit versions of MATLAB. To determine whether your system is a 32-bit or 64-bit system, right-click on the **My Computer** icon on your desktop, select **Properties**, and look at the System information on the **General** tab. To download the 64-bit version of MATLAB, go to the downloads section of the MathWorks Web site, www.mathworks.com/web_downloads.

Upgrade Installations

If you are upgrading your MATLAB installation to the latest release, The MathWorks recommends that you install the new version in a new installation folder. You do not need to delete an existing MATLAB installation before installing this release. You can run multiple versions of MATLAB on the same system — see “Running Multiple Versions of MATLAB” on page 3-21. For information about adding an additional toolbox to an existing MATLAB installation, see “Installing Additional Products” on page 1-26.

Note If you choose to overwrite an existing installation from a prior release, the installer deletes all the products in your installation folder and installs only those products included in your current license, to avoid potential product incompatibilities. For example, if your existing installation includes 10 products, but your current PLP only includes 9 of those products, your MATLAB installation will only include the 9 products included in your current PLP.

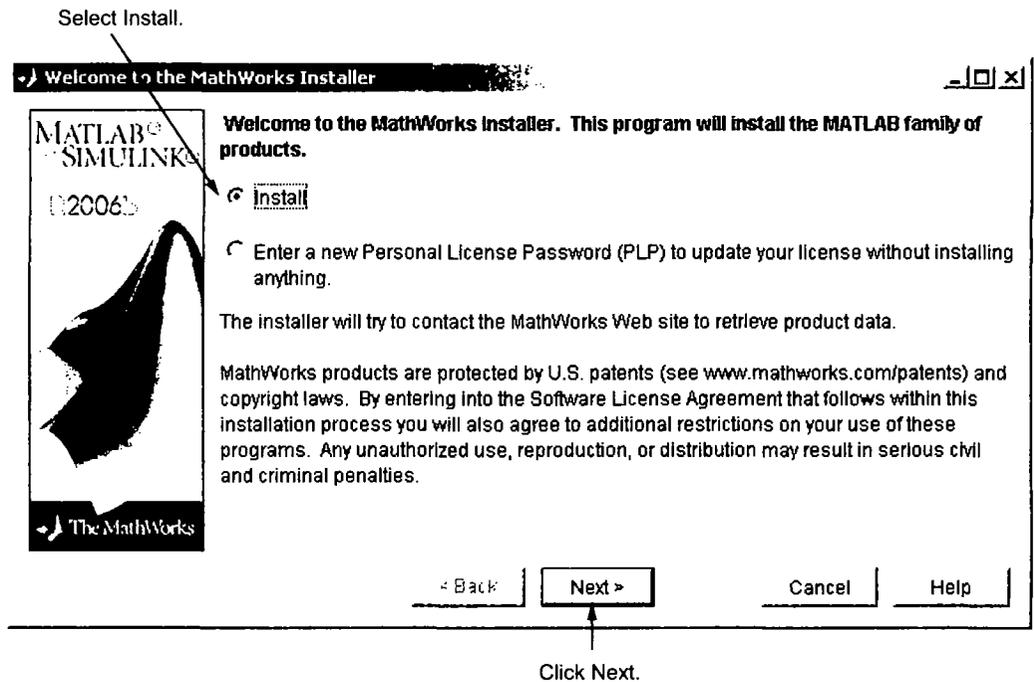
Saving MATLAB Preferences Files

The installer does not overwrite existing MATLAB preference files on your system. Note, however, that if you plan to remove a Release 12 or earlier installation, you might want to first save your MATLAB preference files. Before Release 12.1, the uninstaller provided with MATLAB automatically deleted MATLAB preferences files. See “Using the Uninstaller” on page 3-30 for more information.

Step 2: Start the Installer

Insert the MathWorks product CD into the CD-ROM drive or double-click the installer file you downloaded from the MathWorks Web site. The MathWorks installer starts automatically, displaying the Welcome to the MathWorks Installer dialog box.

Select the **Install** option and click **Next**. (The **Update license** option is useful when you purchase a product after a trial and just want to update your license without reinstalling products. See “Updating a License” on page 1-23.)



Note During installation, the installer attempts to contact the MathWorks Web site to determine if newer versions of products are available. The installer does not send any personal information from your computer to The MathWorks.

Step 3: Enter Identification and License Information

Enter your name, company name, and Personal License Password (PLP) in the License Information dialog box and click **Next**. Your PLP identifies the products you are licensed to install. To avoid typing errors, copy the PLP from the license e-mail and paste it into this dialog box. Take care to copy the entire PLP if your e-mail program has caused it to wrap onto two lines.

Note Do not confuse your PLP with your License Number. A PLP is a series of five-digit numbers prefixed with a two-digit code, such as 16- 12345-12345-12345-12345-12345.

If you do not have your PLP, you can click the **Get my PLP** button to obtain it. To get your PLP this way, you must have an Internet connection and a MathWorks account. If you do not already have an account, you can create an account at the MathWorks Web site.

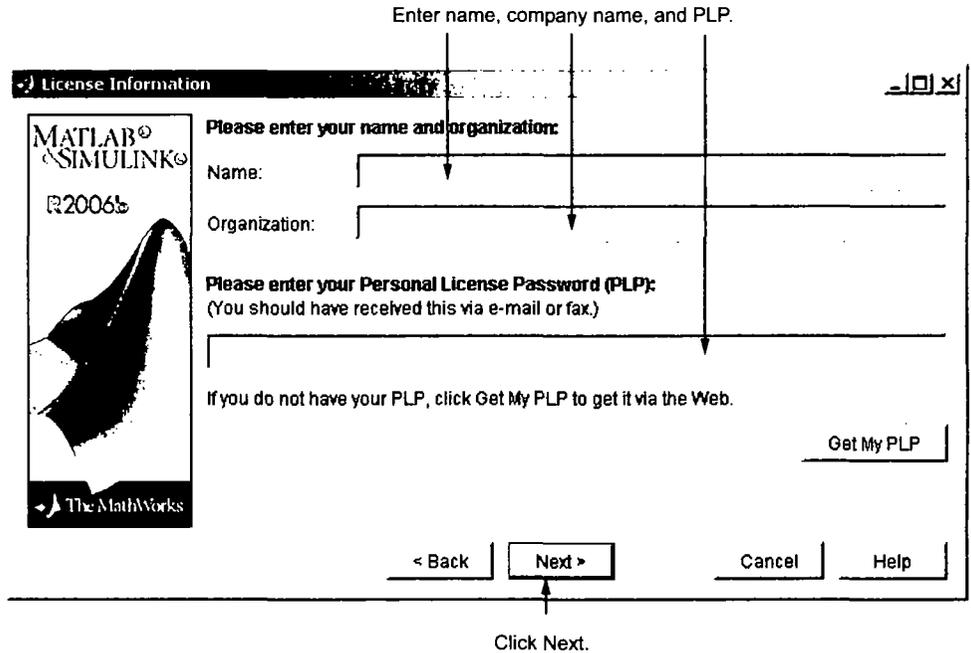
Enter name, company name, and PLP.

Click Next.

Checking for Product Updates

After you enter your PLP, the installer checks if newer versions of the products identified in your PLP are available for download at The MathWorks Web site. If it finds newer versions, the installer displays the MathWorks Account dialog box. If you want the installer to download the latest product updates during installation, enter your e-mail address and password for your MathWorks Account. If you do not have an account and want to learn more about setting one up, click **Create an account**

If you do not want to download these new versions and instead install only the products available on the product CD, select the **Install the product versions from my CD...** option.

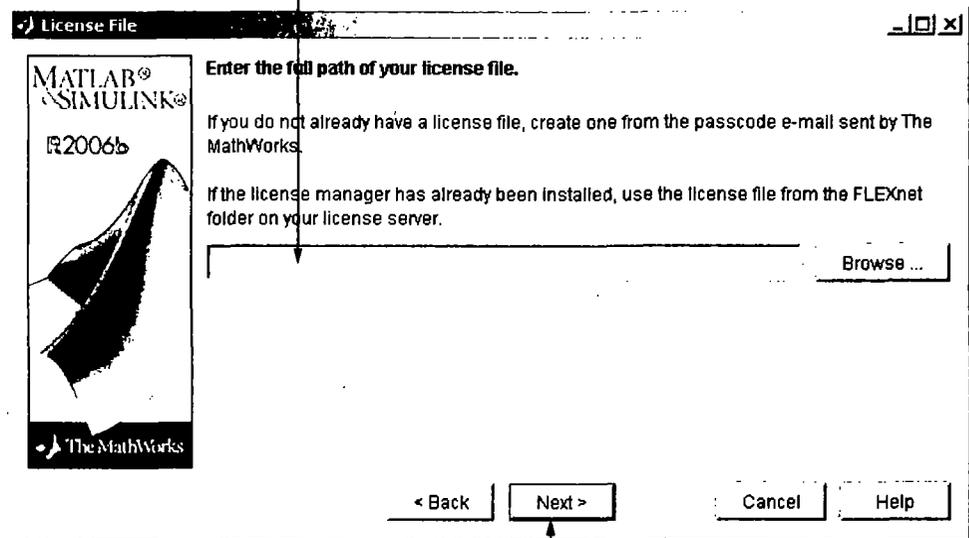


Step 4: Specify Location of License File

In the License File dialog box, specify the full path of your License File and click **Next**. Your License File identifies the products you are licensed to run.

If you are installing the license manager, use the License File you created from the license passcode e-mail. See “Personal License Passwords (PLPs) and License Files” on page 3-4 for more information. If you are installing MATLAB on a client system, see “Installing MATLAB on a Client System” on page 2-28 for instructions.

Specify full path of License File

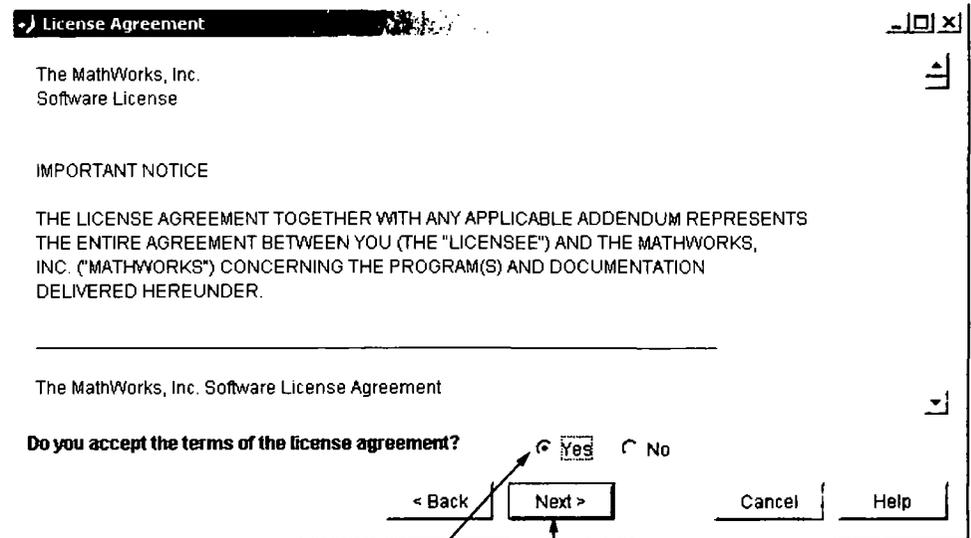


Click Next.

Step 5: Review the Software Licensing Agreement

Review the software licensing agreement and, if you agree with the terms, select **Yes** and click **Next**.

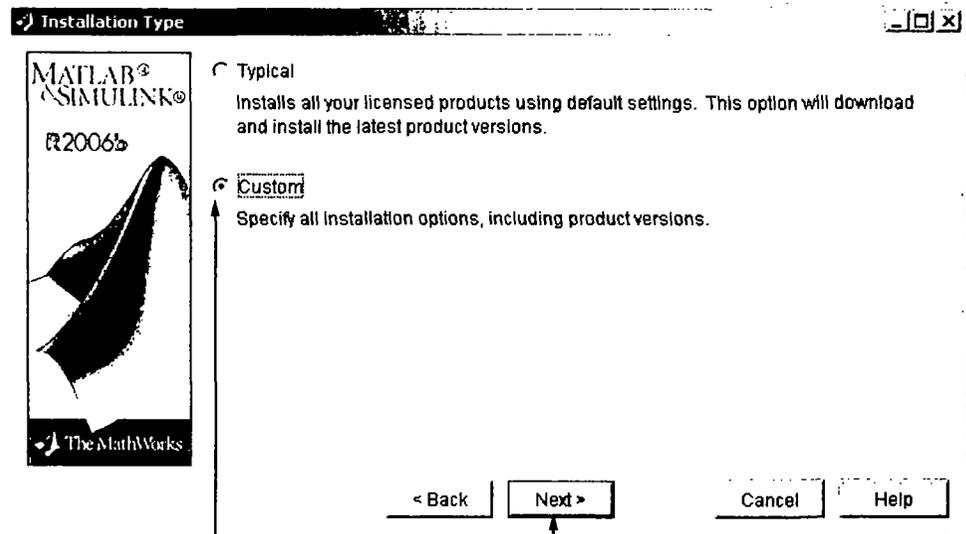
After the installation is complete, you can view or print the license agreement using the file `license.txt` located in the top-level MATLAB installation folder.



Select Yes. Click next.

Step 6: Specify Type of Installation

For network installations that include the license manager, you must choose the **Custom** option. (For information about network installations that do not include the license manager, see “Installing MATLAB on a Client System” on page 2-28.)

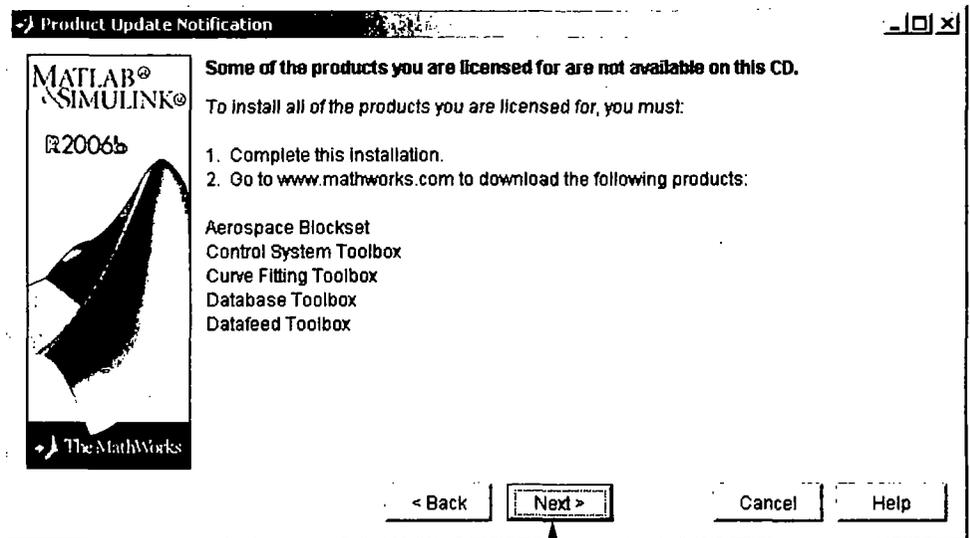


Select Custom.

Click Next.

Product Update Information

If your PLP identifies products that are not on any of the CDs, and you are not currently connected to the Internet or you chose not to download product updates, the installer displays the Product Update Notification dialog box. This dialog box lists the names of the new or updated products that are available at the MathWorks Web site. The installer also includes this information in the Product Configuration Notes dialog box. To download these products, go to the MathWorks Web site after installation is complete. Click **Next** to continue with the installation.

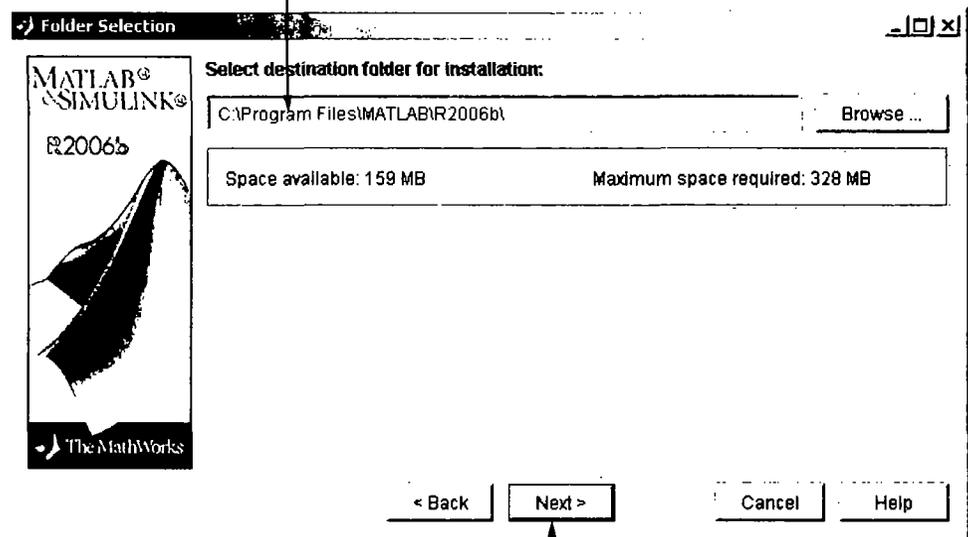


Click next..

Step 7: Specify Installation Folder

Specify the name of the folder where you want to install MATLAB in the Folder Selection dialog box. You can accept the default destination folder or specify the name of a different installation folder. If the folder doesn't exist, the installer creates it. Do not specify a folder name that contains an at (@) sign or a dollar (\$) sign, and the full path of the installation folder must not include a folder named private. Click **Next** to continue with the installation.

Specify name of installation folder.



Click Next.

Step 8: Specify Products to Install

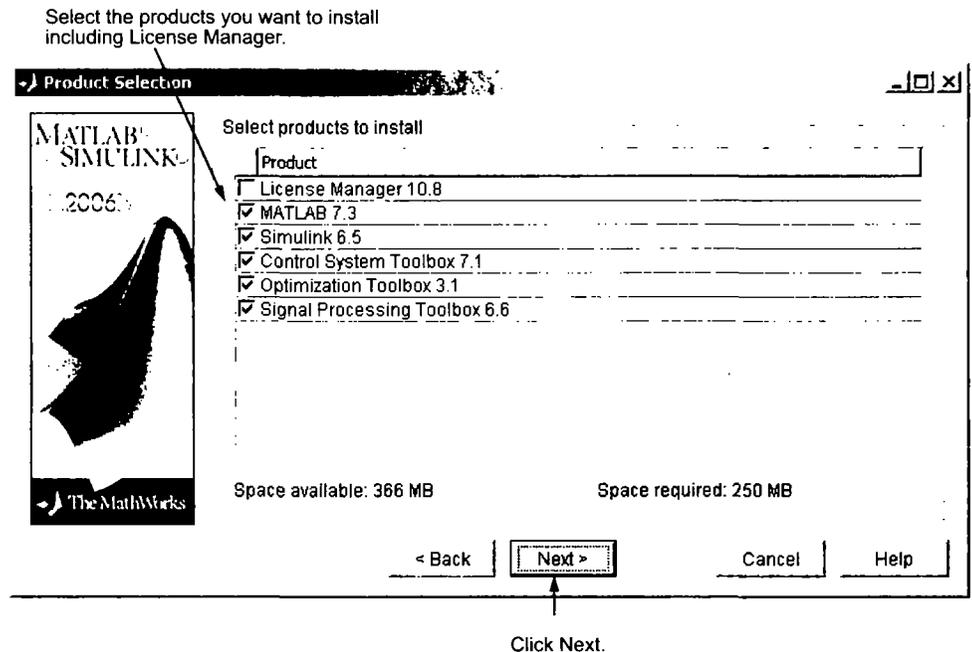
You can specify which products you want to install in the Product Selection dialog box. This dialog box lists all the products you are licensed to install as identified by your PLP. Each product includes the online documentation, in compressed form. PDF documentation is only available at the MathWorks Web site. If you chose to download product updates, see “Product Selection When Updates Are Available” on page 1-13.

In the dialog box, all the products are preselected for installation. If you are setting up a remote access configuration, leave all the products in the

list selected for installation. In this configuration, you install MATLAB and the other products on your license server. You must also select the license manager in the product list by clicking the check box next to its name. By default, the license manager is not preselected for installation.

If you are setting up a local access configuration, clear all the product check boxes in the product list and select only the license manager. In this configuration, you only run the license manager on your license server; users on client systems run local copies of MATLAB.

After making your selections, click **Next** to proceed with the installation. After clicking **Next**, you might receive a message telling you that one or more of the products you selected depend on other products. See “Product Dependencies” on page 4-4 for more information.

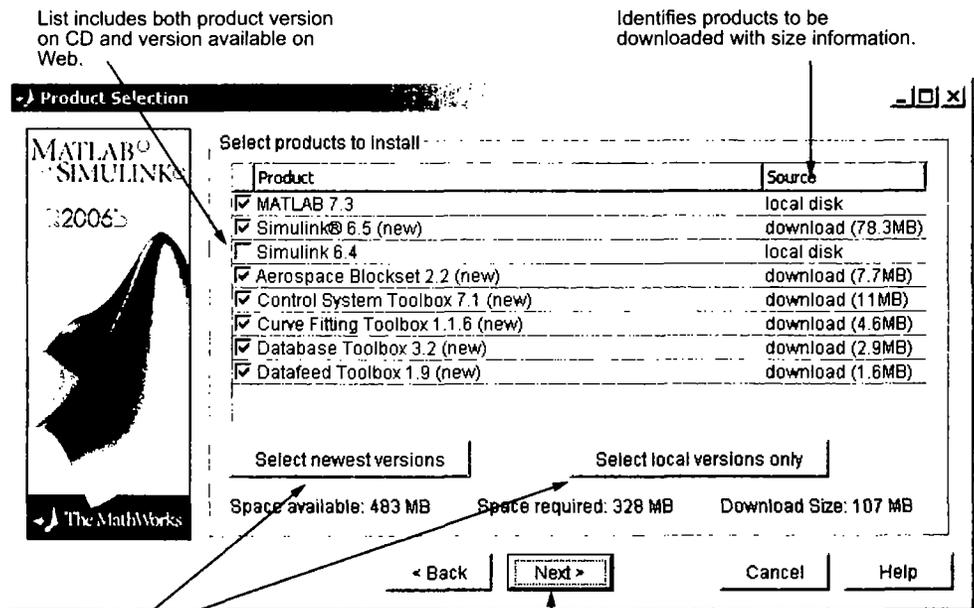


Product Selection When Updates Are Available

If you chose to download product updates, the installer displays both the version that exists on the CD and the new version of the product in the

product list. The updates in the product list are identified with the label **(new)**. The Product Selection dialog box also includes a column, called Source, that identifies whether the product is going to be downloaded during installation and provides information about the size of the download file. (At “Step 11: Confirm Your Choices” on page 2-20, the Confirmation dialog box gives the total download size.)

By default, the latest version is preselected for installation, but you can choose which products you want to install. If you do not want to download any product updates, click the **Select local versions only** button; this deselects all the product downloads in the product list. If you want to install only the latest versions of all products, click **Select newest versions**. After selecting the products you want to install, click **Next** to continue with the installation.

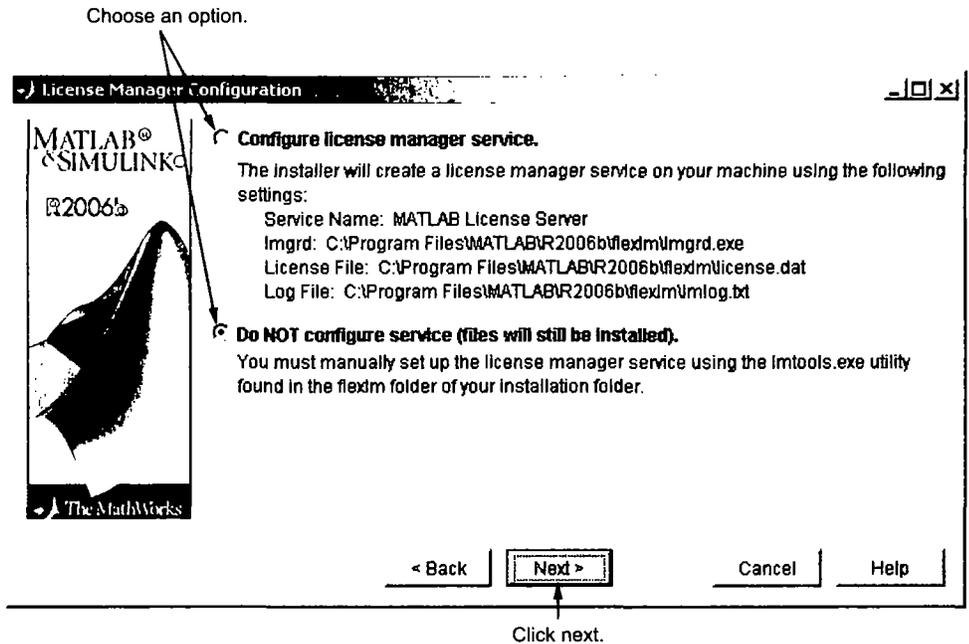


Use buttons to install the latest version of all products, or only products on CD (no downloads).

Click Next.

Step 9: Configure the License Manager Service

Installing the license manager service provides automatic startup at system boot time and centralizes administration of the license manager through the Windows **Services** control panel. If you choose not to install the license manager service, you can do it later. See “Managing the License Manager Service” on page 3-14 for more information.

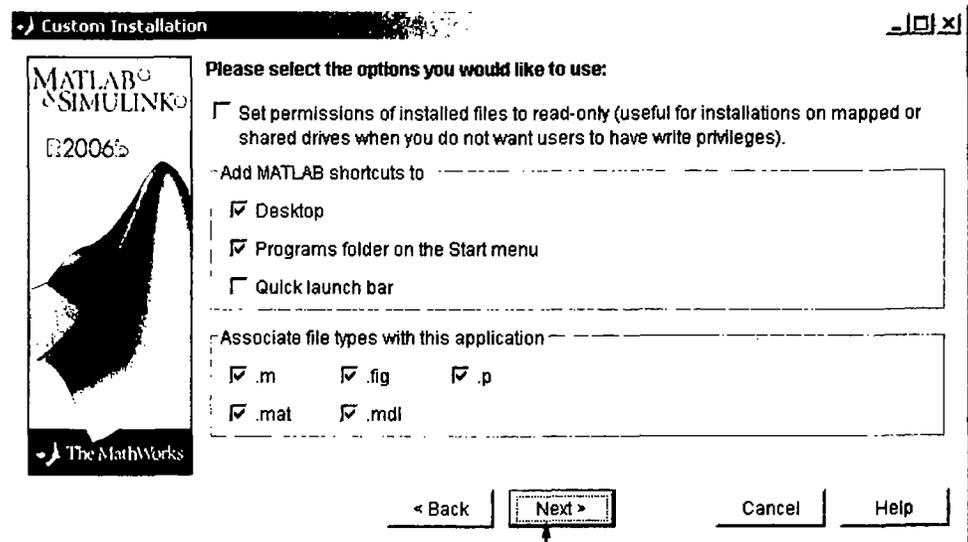


Step 10: Specify Installation Options

For Custom installations, you can specify several installation options in the Custom Installation dialog box, including

- Setting the permissions of all installed files to read only. This option can be useful for remote access configurations where users access the MATLAB files on your license server.
- Determining whether the MathWorks Installer puts shortcuts for MATLAB in the **Start** menu, the **Quick launch bar**, and on the desktop
- Specifying which files the Windows operating system associates with MATLAB based on their file extension. For example, if you associate files with the `.m` file extension with MATLAB, Windows displays MATLAB M-file as the type in the Windows Explorer window.

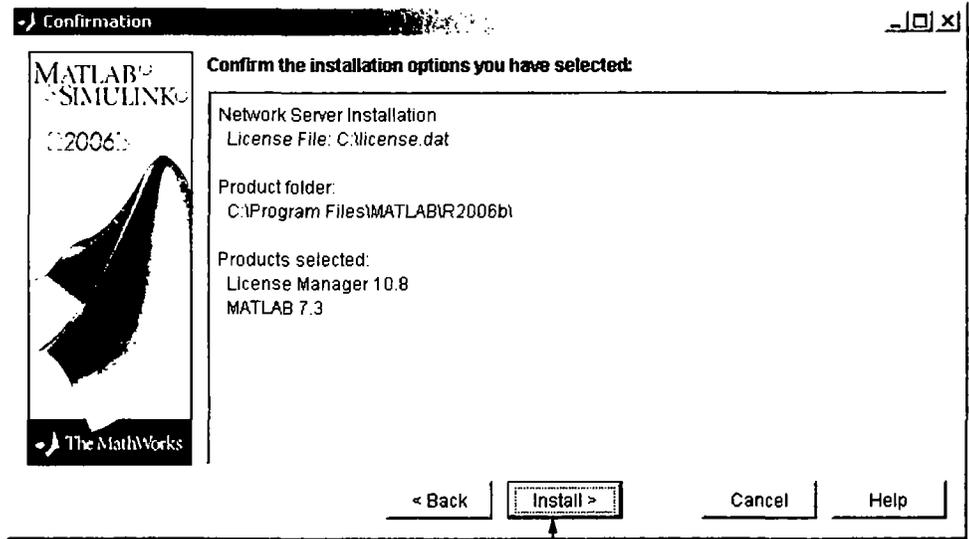
After selecting installation options, click **Next** to proceed with the installation.



Click Next.

Step 11: Confirm Your Choices

Before it begins copying files to your hard disk, the MathWorks Installer displays a summary of your installation choices. To change a setting, click the **Back** button. To proceed with the installation, click **Install**.



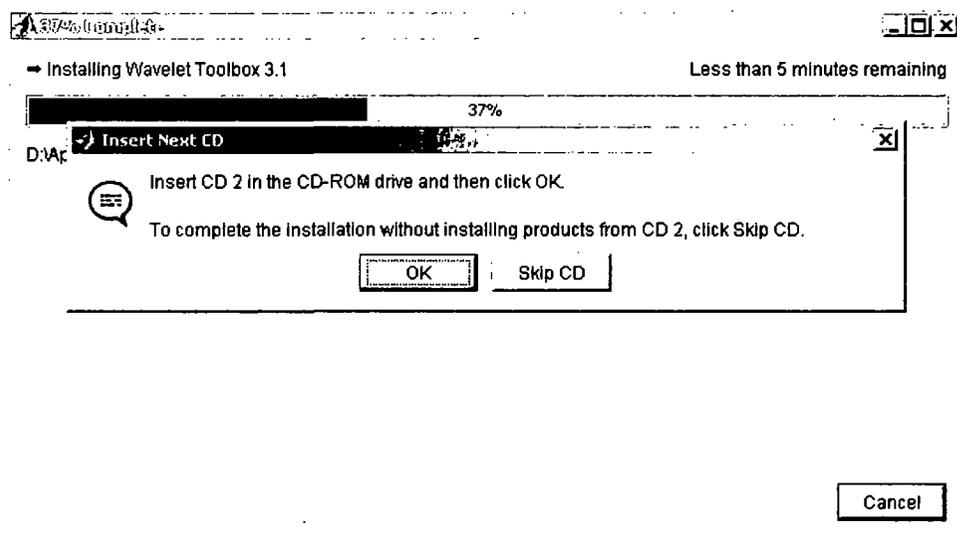
Click Next.

As it copies files to your hard drive, the installer displays a status dialog box to show the progress of the installation.

Installing Products from the Other CDs

You must use CD 1 to start an installation. If you are installing products that are not on CD 1, the installer prompts you to insert another CD in your CD-ROM drive. (Depending on which products you are licensed to install, you may not need to use all of the product CDs during the installation.)

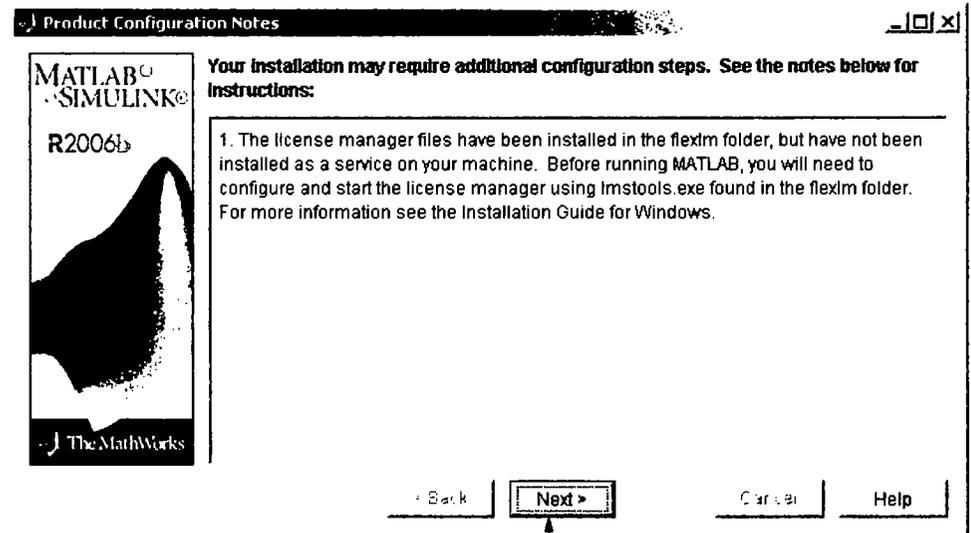
After switching the CDs, click **OK** to continue with the installation. If you do not want to install these products, click **Skip CD**. You can always install these products later. See “Installing Additional Products” on page 1-26 for information.



Step 12: Read Product Configuration Notes

Depending upon which products you install, the MathWorks Installer might display a dialog box containing two types of information:

- **Product configuration information** — Certain products require additional configuration. If you installed one of these products, the installer lists the configuration commands in this dialog box. You can copy this command to your system clipboard for use after the installation is completed. See “Configuring Certain Products” on page 1-21 for more information.
- **Available product updates** — If your PLP identifies products that are not on any of the CDs, and you are not currently connected to the Internet or you chose not to download product updates, the installer lists them in this dialog box. After installation is complete, you must go to the MathWorks Web site to download them.

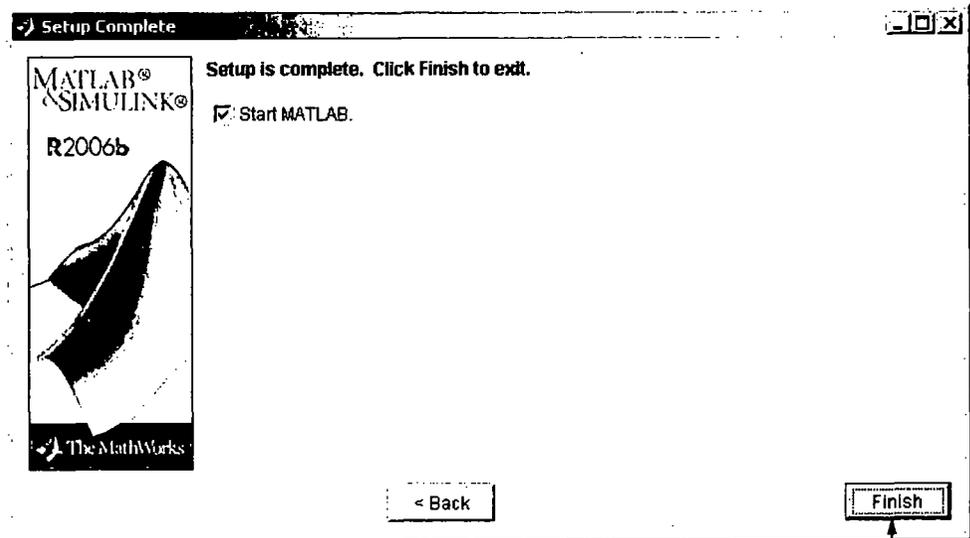


Click Next.

Step 13: Complete the Installation

When the MathWorks Installer finishes, it displays the Setup Complete dialog box. In this dialog box, you can choose to start MATLAB as you exit the installer. If you do not want to start MATLAB, clear this check box. Click **Finish** to exit the installer.

If you installed the license manager as a service, The MathWorks recommends that you restart your computer now. The license manager must be running to start MATLAB. As a service, the license manager gets started automatically at system boot time. To learn more, see “Starting and Stopping the License Manager” on page 3-15.



Click Finish.

MATLAB Top-Level Folder Structure

After a successful installation, your MATLAB installation folder contains a folder for this release. This folder contains the folders listed in the following table. (Your installation might include additional folders, depending on which products you installed.)

Folder	Description
\bin\win32	The MATLAB executable and associated files
\demos	MATLAB demonstration programs
\extern	Tools for building external interfaces to MATLAB
\flexlm	The FLEXnet license manager and license management utilities (network installations only)
\help	Help system
\ja	MATLAB internationalization files
\java	MATLAB Java JAR files
\notebook	MATLAB Notebook
\sys	Tools and operating system libraries that MATLAB depends on
\toolbox	The MathWorks toolboxes
\uninstall	The MATLAB Uninstaller program
\work	Default initial current folder

Step 14: After You Install

For information about how to start MATLAB after exiting the installer and other post-installation tasks, see “Starting MATLAB” on page 1-20. In addition, network installations might require the following post-installation tasks:

- “Setting Up Network Named User Licensing” on page 2-25
- “Allowing Access to License Manager Daemons” on page 2-27

Setting Up Network Named User Licensing

Note This is only required for Network Named User licenses. Concurrent licenses do not require this task.

If you have a Network Named User license, after you install your products, you might need to

- Check that the FLEXnet options file includes all licensed users — see “Checking the FLEXnet Options File” on page 2-26.
- Check that the DAEMON line in your License File includes the options file location, `options=matlabroot\flexlm\MLM.opt`, where *matlabroot* is your MATLAB installation folder.

Note Do not confuse your License File with your FLEXnet options file. The License File contains the encrypted passcodes for each product you are licensed to install and run. The FLEXnet options file is a text file that contains license manager option statements, such as INCLUDE, EXCLUDE, and GROUP statements.

Checking the FLEXnet Options File

The installer creates the FLEXnet options file, *matlabroot\flexlm\MLM.opt*, as part of the installation process. In this options file, the installer puts an INCLUDE line for each product you are licensed to install and run. By default, the installer puts the username of the person who performed the installation in these INCLUDE lines. If you performed your own installation, the options file might not need editing. However, if you performed the installation for another user, or if your license server is installed on a UNIX system and you had root privileges when you performed the installation, you will need to edit this options file.

For example, if your username is *sysadmin*, the options file would look like this.

```
INCLUDE MATLAB USER sysadmin
INCLUDE Signal_Toolbox USER sysadmin
INCLUDE Optimization_Toolbox USER sysadmin
```

You must replace *sysadmin* in these INCLUDE statements with the usernames of the licensed users. Because the installer includes the *GROUPCASEINSENSITIVE* option in the options file and sets it to *ON*, login names are not case sensitive. If multiple users are licensed for the same product, you must create a separate INCLUDE line for each user or use the FLEXnet *GROUP* syntax. The number of users must not exceed the number of license keys available for that product. This example shows both methods.

Note If you edit the options file, make certain that there are no extra space characters at the end of each INCLUDE line.

```
GROUPCASEINSENSITIVE ON
GROUP matlab_users tom judy mike jean brad
GROUP signal_users tom mike brad
INCLUDE MATLAB GROUP matlab_users
INCLUDE Signal_Toolbox GROUP signal_users
INCLUDE Optimization_Toolbox USER mike
INCLUDE Optimization_Toolbox USER jean
```

Allowing Access to License Manager Daemons

If you run the license manager on a server that is protected by a security firewall, you must open the firewall to allow client installations to communicate with the license manager daemons: `mlm.exe` and `lmgrd.exe`. (For information about these daemons, see “License Management Daemons” on page 3-3.)

Opening Access to the License Manager Daemon

Using the interface provided by your firewall program, allow access on port 27000 for the license manager daemon. This is the default port used to communicate with the license manager. If you changed this port number, specify the port number used at your site. See “SERVER Line” on page 3-7 for more information.

Opening Access to the Vendor Daemon

The port used by the vendor daemon is assigned dynamically. If you need to open access to the vendor daemon in your firewall, you must assign a port for vendor daemon communication, rather than using this dynamic port assignment. You can then open the assigned port number in your firewall.

To specify the vendor daemon port number, edit the License File on your license server and add the `port=` syntax to the DAEMON line. For more information about specifying the vendor daemon port number, see “DAEMON Line” on page 3-7.

Installing MATLAB on a Client System

This section describes how to install MATLAB on a client system running Windows in a local client access configuration. See “Local Client Access Configuration” on page 2-4 for information about this type of installation.

The installation procedure on a client system is almost identical to the procedure on a server; however, there are some differences. For example, in a client installation, you do not install the license manager. This table lists all the steps in the network installation procedure and notes where they are different on client systems. For more detailed information about any of the steps, go to the description of the step in “Overview of Steps in Network Installation Procedure” on page 2-5.

Basic Network Installation Procedure	Client Installation Notes
“Step 1: Before You Install” on page 2-6	You do not have to stop the license manager.
“Step 2: Start the Installer” on page 2-8	Same as on a license server.
“Step 3: Enter Identification and License Information” on page 2-9	Use the PLP that you used for your license server installation.
“Step 4: Specify Location of License File” on page 2-11	Specify the location of the License File that was processed by the installer during the license manager installation on the license server. See “License File Location” on page 2-29 for more information.
“Step 5: Review the Software Licensing Agreement” on page 2-12	Same as on a license server
“Step 6: Specify Type of Installation” on page 2-13	You can select either Typical or Custom . Choose Custom to specify which products you want to install.
“Step 7: Specify Installation Folder” on page 2-15	Same as on a license server.
“Step 8: Specify Products to Install” on page 2-15	Do not select the license manager in the product list. You do not need to install the license manager on client systems.

Basic Network Installation Procedure	Client Installation Notes
“Step 9: Configure the License Manager Service” on page 2-18	Not applicable on client systems
“Step 10: Specify Installation Options” on page 2-19	Same as on a license server
“Step 11: Confirm Your Choices” on page 2-20	Same as on a license server
“Step 12: Read Product Configuration Notes” on page 2-22	Same as on a license server
“Step 13: Complete the Installation” on page 2-23	The license manager must be running on your license server before you can run MATLAB on a client.

License File Location

For client installations, you must specify the location of the License File that was processed by the installer during the license manager installation on the license server. (The installer adds the SERVER and DAEMON lines to the License File during installation. See “License File Processing on a License Server” on page 3-6 for more information.)

The following table lists the location in which the installer stores this License File on PC and UNIX license servers.

License Server Platform	Location of License File on License Server
PC running Windows	<i>matlabroot\flexlm\license.dat</i>
UNIX	<i>matlabroot/etc/license.dat</i>

In a client installation, the installer replaces the DAEMON line with the line USE_SERVER and stores the License File in the *matlabroot\bin\win32* folder.

2 Network Installation Procedure

License Management

This chapter describes the licensing architecture used by The MathWorks and provides information about monitoring the license manager in a network installation. This information can be valuable if you want to integrate MATLAB into an existing FLEXnet licensing scheme.

Licensing Components (p. 3-2)	Primary licensing components, such as the License File and the daemons that are used by FLEXnet
FLEXnet License Management Utilities (p. 3-10)	Tools provided to monitor the license manager
Managing the License Manager Service (p. 3-14)	Starting, stopping, and installing the license manager service
Viewing the License Manager Log File (p. 3-17)	Looking at the License Manager Log File (lmlog.txt)
Using Time-Outs to Free Idle License Keys (p. 3-18)	Using the FLEXnet TIMEOUT option to free idle license keys in concurrent installations
Creating Advanced Configurations (p. 3-21)	Describes how to integrate MATLAB into a pre-existing licensing scheme, run multiple versions of MATLAB at the same time, and create redundant configurations
Running the Installer Noninteractively (p. 3-28)	Describes how to use an initialization file with the installer
Removing (Uninstalling) an Installation (p. 3-30)	Describes how to remove a MATLAB installation

Licensing Components

Note The information in this section pertains to network installations only. You do not need to read this section if you are performing an individual installation.

In a network installation, The MathWorks uses a license manager program, called the FLEXnet Publisher Licensing Module (a product of Macrovision Corporation), to control access to its software. This section describes the major components of the FLEXnet licensing architecture.

- “License Management Daemons” on page 3-3
- “Applications” on page 3-3
- “Personal License Passwords (PLPs) and License Files” on page 3-4
- “License File Processing on a License Server” on page 3-6
- “License File Processing on Client Systems” on page 3-8
- “License File Search Path” on page 3-8

Throughout this section, references to the *matlabroot* folder refer to your MATLAB installation folder.

Note For more detailed information about FLEXnet and license management, see the *FLEXnet Licensing End User Manual*, available in PDF format in *matlabroot*\flexlm\enduser.pdf, where *matlabroot* is your MATLAB installation directory, as returned by the *matlabroot* function.

License Management Daemons

FLEXnet uses a group of processes, called *daemons*, to control access to software applications. These daemons run on a server in your network. In this section, this server is referred to as your *license server*.

License Manager Daemon

The main daemon is the license manager daemon (`matlabroot\flexlm\lmgrd.exe`). This daemon is responsible for managing the initial communication between the licensed application and the vendor daemons associated with each licensed application.

Vendor Daemon

Every vendor that uses FLEXnet to manage the licensing of its products must supply a vendor daemon. The vendor daemon is responsible for keeping track of how many license keys are available for a particular application and for granting or denying access to the application, depending on the current usage of these license keys. The MathWorks vendor daemon is `matlabroot\flexlm\mlm.exe`.

Applications

When a user invokes MATLAB anywhere on the network, MATLAB communicates with the license manager daemon on the license server. The license manager forwards the request to the vendor daemon. The vendor daemon checks out a license key for the product. This license key checkout is transparent to MATLAB users.

The license manager program reads a License File to determine the numbers of keys you have purchased for each product.

Personal License Passwords (PLPs) and License Files

When you purchase one of the network license options, The MathWorks sends you licensing information in an e-mail or fax. This licensing information varies depending on your license option, but can include a Personal License Password (PLP) and a License File.

Personal License Passwords (PLPs)

A PLP is a series of five-digit numbers prefixed with a two-digit code, such as 16-12345-12345-12345-12345, that identifies the products that you are licensed to install. Do not confuse your PLP with your License Number.

License Files

A License File is a specially formatted ASCII text file that contains encrypted product passcodes for each product you are licensed to run. Each product passcode identifies the number of keys available for that product. The license manager grants or denies access to a product depending on the availability of the license keys associated with each product. MATLAB client installations use the License File to determine how to connect to the license server.

The following figure presents a sample of a License File sent by The MathWorks. Each line in the License File that begins with the keyword INCREMENT identifies a product you purchased. Each INCREMENT line identifies the product, the number of keys available for the product, and other information. (The INCREMENT lines in your License File might not include every element shown in the example.) The backslash (\) character indicates that the line continues onto the next line.

The lines that begin with a pound sign (#) are comments. The MathWorks Installer uses the information in these comment lines (license server host ID or Internet address) when it processes the License File during installation.

```
# BEGIN-----cut here-----CUT HERE-----BEGIN
# MATLAB license passcode file.
# LicenseNo: 12345          HostID: INTERNET=144.212.101.43
INCREMENT TMW_Archive MLM 16 01-sep-2006 0 \
  BC9DE773A77D15AF8 VENDOR_STRING=83 HOSTID=DEMO SN=123
INCREMENT MATLAB MLM 16 01-sep-2006 1 \
  4C9D3348561BE9E3B USER_BASED DUP_GROUP=U SN=123
INCREMENT SIMULINK MLM 16 01-sep-2006 1 \
  1CD148466EF58DF8B USER_BASED DUP_GROUP=U SN=123
INCREMENT Signal_Toolbox MLM 16 01-sep-2006 1 \
  6CF74B458BA143DC3 USER_BASED DUP_GROUP=U SN=123
# END-----cut here-----CUT HERE-----END
```

Creating a License File

You must create a License File from the licensing information you received in your license passcode e-mail message from The MathWorks. Follow these instructions:

- 1 Open a text file, using any ASCII text editor.
- 2 In the license passcode e-mail message, select all the information from the BEGIN line to the END line. Make sure to select all the information contained in this section, including the comment lines.
- 3 Copy the license information from the e-mail message and paste it into the open text file.

In general, do not edit the licensing information; you may invalidate a license. However, note the following:

- Make sure you remove any line breaks that your e-mail program may have inserted between the start and end of an INCREMENT line. If an INCREMENT line must continue onto more than one line, use the \ (backslash) character, as in the following:

```
INCREMENT Identification_Toolbox MLM 16 01-sep-2006 1 \
  EC5C175AGG025G2AD73A DUP_GROUP=UH SN=12345
```

- Make sure the fields in the INCREMENT line are separated by spaces, not by tab characters.

- Do not change the case of any characters. License Files are case sensitive.

If you received your license passcodes in a fax, create the License File by typing in the license passcode information exactly as it appears in the fax.

- 4 Save the file as a plain ASCII text file, naming it `license.dat`, and store it in any convenient folder.

License File Processing on a License Server

When you install the license manager as part of a network installation, the installer processes your License File, adding two lines to the file: the `SERVER` line and the `DAEMON` line. These lines provide information about the location of the license server and daemon. The installer puts this processed version of the License File into the `matlabroot\flexlm` folder.

The following is a sample of a License File after it was processed by the installer during a license server installation. Note the addition of the `SERVER` and `DAEMON` lines at the top of the file.

```
SERVER yourserver INTERNET=144.212.101.43 27000
DAEMON MLM C:\Program Files\MATLAB\R2006b\flexlm\mlm.exe \
options=C:\Program Files\MATLAB\R2006b\flexlm\MLM.opt
# BEGIN-----cut here-----CUT HERE-----BEGIN
# MATLAB license passcode file.
# LicenseNo: 123           HostID: INTERNET=144.212.101.43
INCREMENT TMW_Archive MLM 16 01-sep-2006 0 \
BC9DE773A77D15AF8 VENDOR_STRING=83 HOSTID=DEMO SN=123
INCREMENT MATLAB MLM 16 01-sep-2006 1 \
4C9D3348561BE9E3B USER_BASED DUP_GROUP=U SN=123
INCREMENT SIMULINK MLM 16 01-sep-2006 1 \
1CD148466EF58DF8B USER_BASED DUP_GROUP=U SN=123
INCREMENT Signal_Toolbox MLM 16 01-sep-2006 1 \
6CF74B458BA143DC3 USER_BASED DUP_GROUP=U SN=123
# END-----cut here-----CUT HERE-----END#
```

SERVER Line

The **SERVER** line identifies the system on which the license manager is running, using the format

```
SERVER hostname HostID TCP_PortNumber
```

where

- **hostname** is the name of your license server, i.e., the system on which the license manager is running.
- **hostID** is the host ID of the license server. The host ID in the **SERVER** line must match the host ID in the comment section of the License File.
- **TCP_PortNumber** specifies the TCP/IP port to use for communication with the license manager daemon. Do not change this number unless you know it conflicts with other software. A valid port number is any unused port number between 0 and 64000. If the TCP/IP port number is not specified, FLEXnet uses a port number in the range 27000 to 27009.

DAEMON Line

The **DAEMON** line identifies the name and location of the vendor daemon and the name and location of the FLEXnet options file, using the format

```
DAEMON vendor vendor_daemon_path options=options_file_path
```

where

- **vendor** is the name of the vendor daemon, MLM for The MathWorks.
- **vendor_daemon_path** is the fully specified path to the daemon executable, for example,

```
C:\Program Files\MATLAB\R2006b\flexlm\mlm.exe
```

- **options=options_file_path** is the full path of the options file, such as,

```
options=C:\Program Files\MATLAB\R2006b\flexlm\MLM.opt
```

The DAEMON line can optionally specify the port the vendor daemon (`mlm.exe`) uses. The FLEXnet daemon (`lmgrd.exe`) and the vendor daemon use separate ports. If no port is specified, the license manager assigns the vendor daemon port dynamically. To specify the port used by the vendor daemon, use the following syntax

```
DAEMON MLM C:\Program Files\MATLAB\R2006b\flexlm\mlm.exe port=####
```

where `####` is an open port.

License File Processing on Client Systems

In a client installation, when you need to specify the name of a License File, you must specify the License File that was processed during the installation of the license manager on your license server, not the License File you created from the license e-mail. The client installation needs the SERVER line that was added during the installation of the license manager. If you specify the name of the License File you created from the license e-mail, the installer will proceed with the installation, but you will not be able to run MATLAB on the client until you edit the template License File in the `matlabroot\bin\win32` folder, adding the SERVER line from the processed License File.

Note During the client installation process, the installer deletes all the INCREMENT lines and the DAEMON line in the License File and replaces them with a single line `USE_SERVER`.

License File Search Path

When you start MATLAB, it reads the License File to determine the host name and port address of the license server. MATLAB must contact the server to get licenses. MATLAB expects to find the License File, `license.dat`, in a standard location, but you can specify another location using either of the following methods:

- “Using Environment Variables to Specify the License File Location” on page 3-9
- “Using the `-c` Command Line Option” on page 3-9

Using Environment Variables to Specify the License File Location

You can specify the License File you want MATLAB to use by assigning a value to either of two environment variables, listed in the following table.

Environment Variable	Purpose
LM_LICENSE_FILE	Specifies the location of the License File for all applications using FLEXnet on this server.
MLM_LICENSE_FILE	Specifies the location of the License File, but only for products of The MathWorks.

To create these environment variables on Windows systems, use the System Properties dialog box. For example, on Windows XP systems, right-click the **My Computer** icon on your desktop and select **Properties**. In the System Properties dialog box, select the **Advanced** tab. On this tab, click the **Environment Variables** button. On the Environment Variables dialog box, click the **New** button to create a variable.

Note Do not specify a License File with a network path when your system is not on a network. This can cause MATLAB to hang at startup, until the network connection time-out expires.

Using the -c Command Line Option

You can specify the License File that you want MATLAB to use by using the -c option with the `matlab` command. When you use the -c option with the `matlab` command, you explicitly prohibit the searching of these environment variables. When using the -c option, if the path to your license file contains a space, enclose the pathname in quotes.

FLEXnet License Management Utilities

You typically do not need to know anything about the license manager. However, there can be times when you want to get the current status of the license manager or perform other license management tasks. FLEXnet provides several utilities that you can use to monitor and manage licensing activities at your site. The following sections describe these utilities.

- “Using the FLEXnet Command Line Utilities (lmutil.exe)” on page 3-10
- “Using the FLEXnet Utilities Graphical User Interface (LMTOOLS)” on page 3-12

Note If you want to make changes to your licenses, such as combining several licenses or moving an existing toolbox license, you must contact The MathWorks. See “Obtaining Your Licenses” on page 1-28 for contact information.

Using the FLEXnet Command Line Utilities (lmutil.exe)

FLEXnet packages all their license management utilities in a single executable called `lmutil.exe`. The MathWorks Installer includes this utility in every server installation in the `matlabroot\flexlm` folder. To view a list of all the capabilities of the `lmutil` utility, open a Command Prompt window, go to the `matlabroot\flexlm` folder and type

```
lmutil
```

Note You must run the `lmutil` function on the system on which the license manager is running.

The following table lists the names of the most useful tools available through `lmutil`. For more detailed information, see the *FLEXnet Licensing End User Manual*, included in PDF format in your MATLAB installation (`matlabroot\flexlm\enduser.pdf`). For information about accessing these tools through a graphical user interface, see “Using the FLEXnet Utilities Graphical User Interface (LMTOOLS)” on page 3-12.

Utility	Description
lmdiag	Diagnose license checkout problems.
lmdown	Shut down all license daemons (both lmgrd and all vendor daemons) on the license server node.
lmhostid	Report the host ID of a system.
lmreread	Make the license manager daemon reread the License File and start any new vendor daemons.
lmstat	Display the status of all network licensing activities.
lmswitchr	Switch the report log file.
lmver	Determine the FLEXnet version of a library or binary file.

Using lmutil to Get the Current Status of the License Manager

To illustrate using these FLEXnet tools, this section describes how to view the current status of the license manager.

- 1 Open a DOS command prompt window and enter the lmutil command, specifying the name of the lmstat utility.

```
lmutil lmstat -a -c "C:\Program Files\MATLAB\R2006b\flexlm\license.dat"
```

The example uses the -a option to get a verbose listing of available licenses and the -c option to specify the location of the License File to use. You must specify the -c option with every tool invoked through lmutil.exe. If your installation pathname contains spaces, enclose the pathname in quotes.

- 2 View the status information returned.

```
lmutil - Copyright (c) 1989-2005 by Macrovision Corporation. All rights reserved.
```

```
Flexible License Manager status on Fri 10/16/2006 16:57
```

```
License server status: 27000@smithj
```

```
License File(s) on smithj: C:\Program Files\MATLAB\R2006b\flexlm\license.dat:
```

```
smithj: license server UP (MASTER) v.10.8
```

Vendor daemon status (on smithj):

MLM: UP v10.8

Feature usage info:

Users of MATLAB: (Total of 5 licenses available)

Users of SIMULINK: (Total of 5 licenses available)

Users of Control_Toolbox: (Total of 5 licenses available)

Users of Identification_Toolbox: (Total of 5 licenses available).

Using the FLEXnet Utilities Graphical User Interface (LMTOOLS)

FLEXnet provides a graphical user interface (GUI), called LMTOOLS. The LMTOOLS GUI provides access to the same set of tools as the `lmutil` command and displays the results returned in an integrated display window. (For information about using the `lmutil` program, see “Using the FLEXnet Command Line Utilities (`lmutil.exe`)” on page 3-10.)

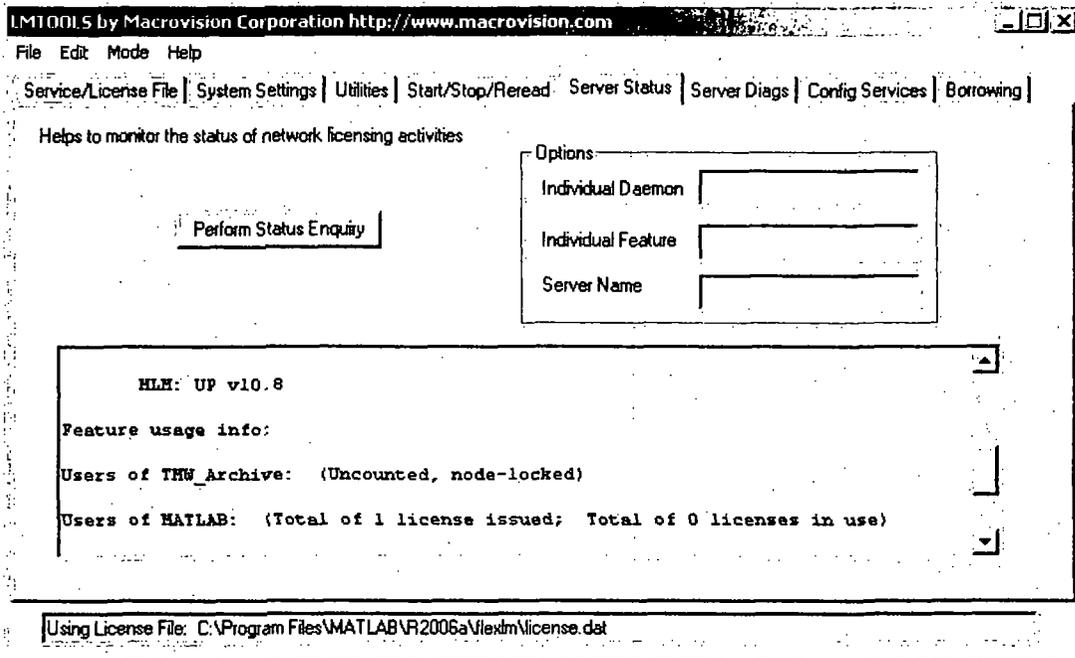
Using LMTOOLS to Get the Current Status of the License Manager

To illustrate how to use LMTOOLS, this section describes getting the current status of the license manager.

- 1 Start LMTOOLS by double-clicking the `lmtools.exe` file in the `matlabroot\flexlm` folder.

2 Click the **Server Status** tab.

3 Click the **Perform Status Enquiry** button. LMTTOOLS displays the same information as if you had entered the command `lmutil lmstat -a`.



Managing the License Manager Service

This section describes how to manage the license manager service in the Windows environment.

- “Installing the License Manager as a Service” on page 3-14
- “Removing the License Manager Service” on page 3-15
- “Starting and Stopping the License Manager” on page 3-15

Installing the License Manager as a Service

To install the license manager service using the `install.exe` utility, follow this procedure:

- 1 Open a DOS Command Prompt window.
- 2 Move to the `matlabroot` folder, where `matlabroot` represents your MATLAB installation folder.
- 3 Enter the command

```
install -c matlabroot\flexlm\license.dat -e matlabroot\flexlm\lmgrd.exe
```

where the `-c` flag specifies the name of the License File that was processed during the installation of the license manager on your license server and the `-e` flag specifies the name of the license manager daemon.

Specifying the Name of the License Manager Service

By default, the license manager service is called the **MATLAB License Server** in the Windows **Services** control panel. To specify a different name, specify the `-n` flag when you use `install.exe`, as follows:

```
install -c matlabroot\flexlm\license.dat -e matlabroot\flexlm\lmgrd.exe  
-n my server
```

Removing the License Manager Service

There are several ways to remove the license manager service:

- Use the `install.exe` utility, specifying the `-r` flag. If the name of the license service is not `MATLAB License Server`, you must specify the name using the `-n` flag, as follows:

```
install -e matlabroot\flexlm\lmgrd.exe -n my server -r
```

For more information, see “Installing the License Manager as a Service” on page 3-14.

- Use the `LMTTOOLS` Utility. Click the **Remove Service** button on the **Config Services** tab. For more information, see “Using the FLEXnet Utilities Graphical User Interface (LMTTOOLS)” on page 3-12.

Starting and Stopping the License Manager

If you installed the license manager as a service, the license manager gets started automatically at system boot time.

There are also several other ways to start and stop the license manager:

- Use the Windows Services control panel. See “Using the Windows Services Control Panel” on page 3-16 for more information.
- Use the `lmutil` utility. See “Using the FLEXnet Command Line Utilities (`lmutil.exe`)” on page 3-10 for more information.
- Use the `LMTTOOLS` Utility. Click the **Start Server** or **Stop Server** button on the **Start/Stop/Reread** tab. See “Using the FLEXnet Utilities Graphical User Interface (LMTTOOLS)” on page 3-12 for more information.

Using the Windows Services Control Panel

To start or stop the license manager using the Windows **Services** control panel:

- 1** Click the **Start** menu.
- 2** Select **Settings**.
- 3** On the **Settings** menu, select **Control Panel**.
- 4** Open the **Services** control panel. On Windows XP systems, click the **Start** button and select **Settings**. On the **Settings** menu, select **Control Panel>Administrative Tools>Services**.
- 5** In the **Services** control panel, scroll through the list of services and right-click the **MATLAB License Server** entry. Select **Start** or **Stop** from the menu.

Viewing the License Manager Log File

To view the license manager log file, open the `matlabroot\flexlm\lmlog.txt` file with any text editor. Note, however, that you cannot access this file with a text editor while the license manager is running. To view the log while the license manager is running, use the LMTTOOLS utility, as follows:

- 1** Start LMTTOOLS by double-clicking the `lmttools.exe` file in the `matlabroot\flexlm` folder.
- 2** Click the **Config Services** tab.
- 3** Click the **View Log** button. LMTTOOLS displays the contents of the license manager log file in a separate window.

Using Time-Outs to Free Idle License Keys

Note You can only use time-outs with concurrent licenses.

If you have a concurrent license, you can use the FLEXnet time-out option to automatically return idle license keys at your site. With license time-outs, the license manager automatically frees license keys that sit idle for a specified period of time, returning them to the pool of available license keys.

For example, when you start MATLAB and other products, such as toolboxes, you check out license keys for these products. If you go home without exiting your MATLAB session, these licenses remain checked out and are not available to other users. With time-out enabled, after your licenses sit idle for a specified period of time, the license manager releases the license keys for all the products you had been using and these keys become available to other users.

For more information about time-outs, see the following sections:

- “Enabling License Time-Outs” on page 3-18
- “Experiencing a License Time-Out” on page 3-20
- “Reacquiring License Keys After a Time-Out” on page 3-20

Enabling License Time-Outs

To take advantage of time-outs, you must specify the FLEXnet TIMEOUT option in a FLEXnet options file. A FLEXnet options file is a text file that contains license manager options statements, such as INCLUDE, EXCLUDE, and GROUP statements.

To enable time-outs, follow these instructions:

- 1 Create a FLEXnet options file, if one does not already exist, using any text editor. Name the options file `MLM.opt` and put the file in the `flexlm` folder in your MATLAB installation folder. For example, you might create the file

```
C:\Program Files\MATLAB\flexlm\MLM.opt
```

- 2 Add the FLEXnet TIMEOUT option to the options file, using the following syntax:

```
TIMEOUT feature seconds
```

For feature, specify MATLAB. You cannot specify separate time-out values for individual MathWorks products, such as toolboxes.

For seconds, specify the amount of idle time you want to allow. The minimum value is 4 hours (14,400 seconds). If you specify a time-out value that is smaller, the minimum value is used. For example, to specify a five hour time-out, enter the following in your options file.

```
TIMEOUT MATLAB 18000
```

Note FLEXnet supports the TIMEOUTALL option that you can use to specify the time-out value used for all products. However, The MathWorks does not recommend using this option. The TIMEOUTALL option affects all the products listed in the License File which could include products from other vendors, not just MathWorks products. In addition, when you use the TIMEOUTALL option, the license manager log file can become misleading. For example, log file entries include whatever value you specify in the TIMEOUTALL option even if the actual value used is another value, such as the MathWorks minimum time-out value.

- 3 Make sure the DAEMON line in your License File specifies the location of your options file. The License File DAEMON line must include the options= syntax. For example, your DAEMON line might look like the following:

```
DAEMON MLM "C:\Program Files\MATLAB\R2006b\flexlm\mlm.exe" \  
options="C:\Program Files\MATLAB\R2006b\flexlm\mlm.opt"
```

- 4 Stop the license manager and restart it. Whenever you change the License File, or change the time-out value in the options file, you must restart the license manager. See “Managing the License Manager Service” on page 3-14 for information about starting and stopping the license manager.

Experiencing a License Time-Out

When time-out is enabled, if you open a MATLAB session, and then leave it inactive for more than 4 hours, when you return, you will find MATLAB still open on your desktop. However, you will see the following message displayed in the Command Window:

```
Your MATLAB session has timed out. All license keys have been returned.
```

The time-out is also recorded in the license manager log file, as follows:

```
Idle session. Returning keys for <user>@<machine>
```

(For more information, see “Viewing the License Manager Log File” on page 3-17.)

Reacquiring License Keys After a Time-Out

To resume a timed-out MATLAB session, just start using MATLAB again. If a license key is available, MATLAB checks it out automatically. Likewise, as you use the toolboxes and other products that you had checked out, MATLAB obtains license keys for these products as well.

If a license key is not available, MATLAB periodically attempts to check out a license, issuing warning messages. After 10 warnings, if a license key is still not available, MATLAB saves the workspace and exits.

Note If you return to a timed-out Simulink® session, you cannot run or modify a currently open model. These menu options are disabled until a Simulink license key becomes available. You can only save a model in a timed-out Simulink session.

Creating Advanced Configurations

Using the MathWorks Installer, you can set up a network installation with little knowledge of license management. Follow the instructions in Chapter 2, “Network Installation Procedure” However, there are configurations that require some additional setup, including

- “Running Multiple Versions of MATLAB” on page 3-21
- “Using Products in a Concurrent Installation from a Stand-Alone Installation” on page 3-22
- “Running MATLAB with Other FLEXnet Applications” on page 3-24
- “Configuring Redundant License Servers” on page 3-24

Running Multiple Versions of MATLAB

You can run multiple versions of MATLAB on the same system without any special considerations in stand-alone, individual license configurations. In remote access configurations, described in “Network Configuration Options” on page 2-2, you must perform some additional set up on your license server and possibly on client systems to provide users with access to multiple versions of MATLAB over the network.

Required Setup on Your License Server

To make multiple versions of MATLAB available in a remote access configuration, follow this procedure on your license server:

- 1 Check the version of your currently installed MATLAB. You must be running MATLAB Version 5.2.x or later to be able to run multiple versions.
- 2 Stop the license manager. (See “Starting and Stopping the License Manager” on page 3-15.)
- 3 Remove the license manager service. (See “Removing the License Manager Service” on page 3-15.)
- 4 Install the license manager included with the current version of MATLAB on your license server. Run the installer and select the license manager in the product list. (See “Step 7: Specify Installation Folder” on page 2-15.)

Required Setup on Client Systems

You must make sure that client installations of MATLAB use the latest version of the license manager.

On Windows systems, the License File in a client installation typically includes a `SERVER` line and a `USE_SERVER` line, as follows.

```
SERVER my_server INTERNET=144.212.121.14 27000
USE_SERVER
```

If your `SERVER` line still identifies the correct license server, you do not need to make any changes to your License File.

In heterogeneous configurations that include clients running on UNIX systems, or if you changed which system is your license server, you must replace the License File in client installations with the License File created by the installer during the license manager installation on your license server. On license servers, the License File resides in the `matlabroot\flexlm` folder. The following table lists the location of the License File on client systems for different MATLAB releases.

MATLAB Version	Location of License File on Client Systems
MATLAB 6.x or later	<code>matlabroot\bin\win32\license.dat</code>
MATLAB 5.2.x or 5.3.x	<code>matlabroot\bin\license.dat</code>

Using Products in a Concurrent Installation from a Stand-Alone Installation

If you have an Individual license in a stand-alone installation, you can still use toolboxes that have a Concurrent license, and are made available over the network from a license server. You must use two License Files: one for the individual licensed products and another for the concurrently licensed products.

Note This capability is only available for an Individual license stand-alone installation on a PC running the Windows operating system.

To set up this configuration, follow these instructions:

- 1** Perform a stand-alone installation of the products for which you have an Individual license, if you have not already done so. Use the PLP for your Individual license in this installation. See “Overview of Steps in Standard Installation Procedure” on page 1-2 to learn how.
- 2** Move the License File (`license.dat`) processed by the installer during this installation out of the `matlabroot\bin\win32` folder to any convenient folder, such as the desktop.
- 3** Run the installer again, this time to install the toolboxes that have a Concurrent license. Use the PLP for your concurrent license in this installation. See “Overview of Steps in Network Installation Procedure” on page 2-5 for more information.
- 4** Specify the location of the License File on your license server (`matlabroot\flexlm\license.dat`) when asked by the installer—see “Step 4: Specify Location of License File” on page 2-11.
- 5** Specify the same installation folder as your stand-alone installation and select only the products you want to use that are not already part of your Individual license installation—see “Step 7: Specify Installation Folder” on page 2-15.
- 6** After the installation is finished, rename the License File that was processed by the installer during this installation. Open the `matlabroot\bin\win32` folder and change the name of the `license.dat` file to `license.lic`.
- 7** Move the original License File, `license.dat`, that was created during the standalone installation, back into the `matlabroot\bin\win32` folder.
- 8** Start MATLAB. You can now run the toolboxes with a concurrent license from your stand-alone installation.

Running MATLAB with Other FLEXnet Applications

If you have another application that uses FLEXnet for license management, you can share a single license manager, or run separate license managers either on the same or different hosts.

The following sections describe these options. For more information, read the *FLEXnet Licensing End User Manual*, available in PDF form in `matlabroot\flexlm\enduser.pdf`, or go to the Macrovision Corporation Web page at www.macrovision.com.

Sharing a Single License Manager

To share a license manager with other applications, combine the DAEMON and INCREMENT lines from both License Files into a single License File. In the combined License File, keep only one SERVER line that identifies the license server host. If the other applications are using a different version of the FLEXnet software than the MathWorks products, run the newest version of the license manager.

Running Separate License Managers

To run separate license managers, use separate License Files. If you are running the license managers on the same host, be sure to use a different TCP port number on the SERVER line in each License File.

Configuring Redundant License Servers

If a large number of licenses are governed by a single license server, failure of the server can become a serious problem. To prevent problems, you may want to set up redundant servers so that, if one server goes down, the license manager can still function.

In the redundant server configuration, three systems are designated to be license servers. All three systems must be running at the time the license manager is started. However, once the license manager is running, only two systems need to be running at any time; this is called a *quorum*. Once the daemons are up and connected on all three systems, only two systems are needed for a quorum; the loss of any one system will not cause any licenses to be revoked.

Setting up a Redundant Installation

Perform these steps to configure a redundant installation:

- 1 Choose the systems that you intend to use as servers. When choosing servers, keep in mind the following:
 - The servers should be able to handle the network traffic associated with license management. A primary server must be chosen. This is the system to which clients connect first and which receives the majority of the network traffic. The primary server is the first server listed in the License File. If this system fails, the next server listed in the License File becomes primary.
 - The servers should be running supported versions of their operating systems.
- 2 Request a License File from The MathWorks.

Once you have chosen the servers, determine the host IDs of each server and provide them to The MathWorks when you request your License File. The MathWorks will generate an appropriate License File. The following example shows a License File that supports redundant servers. Note that the License File lists three Host IDs.

```
# BEGIN-----cut here-----CUT HERE-----BEGIN
# MATLAB license passcode file.
# LicenseNo: 123           HostID: 00034660893e
#                           HostID: 00053327865e
#                           HostID: 00470635883e
INCREMENT TMW_Archive MLM 16 01-jan-0000 0 \
  CD74754B9E91B8036  VENDOR_STRING=87 HOSTID=DEMO SN=123
INCREMENT MATLAB MLM 16 01-jan-0000 5 \
  8D1B1BE24413B8126  HOSTID=ID=123 DUP_GROUP=UH SN=123
INCREMENT SIMULINK MLM 16 01-jan-0000 5 \
  2D031C43289BE055B  HOSTID=ID=123 DUP_GROUP=UH SN=123
INCREMENT Control_Toolbox MLM 16 01-jan-0000 5 \
  0F441940A6E1648FE  HOSTID=ID=123 DUP_GROUP=UH SN=123
INCREMENT Signal_Toolbox MLM 16 01-jan-0000 5 \
  2D245A439FEEA2336  HOSTID=ID=123 DUP_GROUP=UH SN=123
# END-----cut here-----CUT HERE-----END
```

- 3** Install the license manager on the three server systems in the redundant configuration. During each installation, the installer processes the License File, adding SERVER and DAEMON lines. For example, on the system named pooh, the processed License File might contain these lines.

```
SERVER pooh 08002be68759 27000
DAEMON MLM /usr/local/matlab/etc/lm_matlab
```

- 4** Edit the License File on each license server so that each License File contains SERVER lines for all three license servers. For information about where the installer stores this License File on Windows and UNIX systems, see “License File Location” on page 2-29. Do not copy the DAEMON lines; the License Files should contain only one DAEMON line. Note in the following example that the License File contains three SERVER lines and one DAEMON line.

```
# BEGIN-----cut here-----CUT HERE-----BEGIN
# MATLAB license passcode file.
# LicenseNo: 123           HostID: 00034660893e
#                          HostID: 00053327865e
#                          HostID: 00470635883e
SERVER pooh 00034660893e 27000
SERVER piglet 00053327865e 27000
SERVER rabbit 00470635883e 27000
DAEMON MLM /usr/local/matlab/etc/lm_matlab
INCREMENT TMW_Archive MLM 16 01-jan-0000 0 \
  CD74754B9E91B8036  VENDOR_STRING=87 HOSTID=DEMO SN=123
INCREMENT MATLAB MLM 16 01-jan-0000 5 \
  8D1B1BE24413B8126 HOSTID=ID=123 DUP_GROUP=UH SN=123
INCREMENT SIMULINK MLM 16 01-jan-0000 5 \
  2D031C43289BE055B HOSTID=ID=123 DUP_GROUP=UH SN=123
INCREMENT Control_Toolbox MLM 16 01-jan-0000 5 \
  0F441940A6E1648FE HOSTID=ID=123 DUP_GROUP=UH SN=123
INCREMENT Signal_Toolbox MLM 16 01-jan-0000 5 \
  2D245A439FEEA2336 HOSTID=ID=123 DUP_GROUP=UH SN=123
# END-----cut here-----CUT HERE-----END
```

- 5** Put a copy of this edited License File in the `matlabroot\flexlm` folder on all of the systems in the redundant configuration.

- 6 Start the license manager on each system in the redundant configuration and wait for the license manager daemons on the three systems to synchronize with each other. This may take a few minutes.

If, for some reason, the license manager daemons do not connect, take the daemons down on each machine, and rerun the procedure. Network traffic can affect synchronization, so it may require several attempts to establish a proper connection.

- 7 Set up the client systems that access the redundant servers.

To ensure that client systems can switch servers in the event of a failure, you must create an environment variable named `MLM_LICENSE_FILE` on each client system. FLEXnet reads this environment variable to determine which License Files to use. On client systems, specify the host name and port for all the servers in the redundant configuration. Use a semicolon to separate the names.

For example:

```
port@server1;port@server2;port@server3
```

Alternatively, you can specify the License files when you start MATLAB on a client, using the `-c` option with the `matlab` command. To do this, right-click on the MATLAB icon on the desktop and choose **Properties**. In the **Target** field of the **Properties** dialog box, add the `-c` option to the command.

For example:

```
matlabroot\bin\win32\matlab.exe -c port@server1;port@server2;port@server3
```

where *matlabroot* represents your MATLAB installation folder.

Running the Installer Noninteractively

If you have many MATLAB installations to perform, and the information you need to enter for each installation is the same, you can put this information in an installer initialization file and run the MathWorks Installer noninteractively. The initialization file provides the installer with the information that you would otherwise provide using installer dialog boxes. Noninteractive installs (also called *silent* installs) can save time and prevent errors.

Using an Installer Initialization File

To run the installer noninteractively, perform these steps:

- 1 Create your installer initialization file.

The MathWorks provides an initialization file template, named `installer.ini`, located in the top-level folder on the MathWorks CD. Using Windows Explorer, create a copy of this template file in any convenient folder, giving it any name you want. For example, you might create the initialization file

```
C:\temp\my_install.ini
```

- 2 Open your copy of the initialization file, using any text editor, and enter all the required installation information. The initialization file template includes parameters for all the information you need to enter. For example, to specify where you want to install products, set the value of the `destination` parameter to the full path of the installation folder.

```
destination=C:\Program Files\MATLAB\R2006b
```

There are several required parameters, such as name, company name, and PLP. If your initialization file is missing a required field, the installer stops the installation and writes an error to the output log. Two optional parameters you might want to set are `visible` and `outlog`.

- **Controlling visibility** — By default, when you run the installer in silent mode, it still displays the dialog boxes as it progresses through the installation, even though no interactive response is required. If you want to suppress the display of these dialog boxes during a silent installation, set the value of the `visible` parameter to `false`.
- **Requesting an output log** — To request that the installer create an output file that reports on the status of the installation, specify the full pathname as the value of the `outlog` parameter.

3 Save your changes to the file.

4 Run the installer (`setup.exe`), using the `-if` flag to specify the initialization file as a command-line argument. You must specify the full pathname of the initialization file. For example, click the **Start** menu and select the **Run** option. In the Run dialog box, enter

```
setup.exe -if C:\temp\my_install.ini
```

Removing (Uninstalling) an Installation

To remove MathWorks products from your system, use the MathWorks uninstaller program (`uninstall.exe`). This section describes

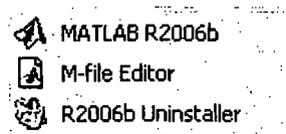
- “Using the Uninstaller” on page 3-30
- “Running the Uninstaller Noninteractively” on page 3-32
- “Removing Installations That Use Dongles” on page 3-34

Using the Uninstaller

To remove MathWorks products from your system, follow this procedure:

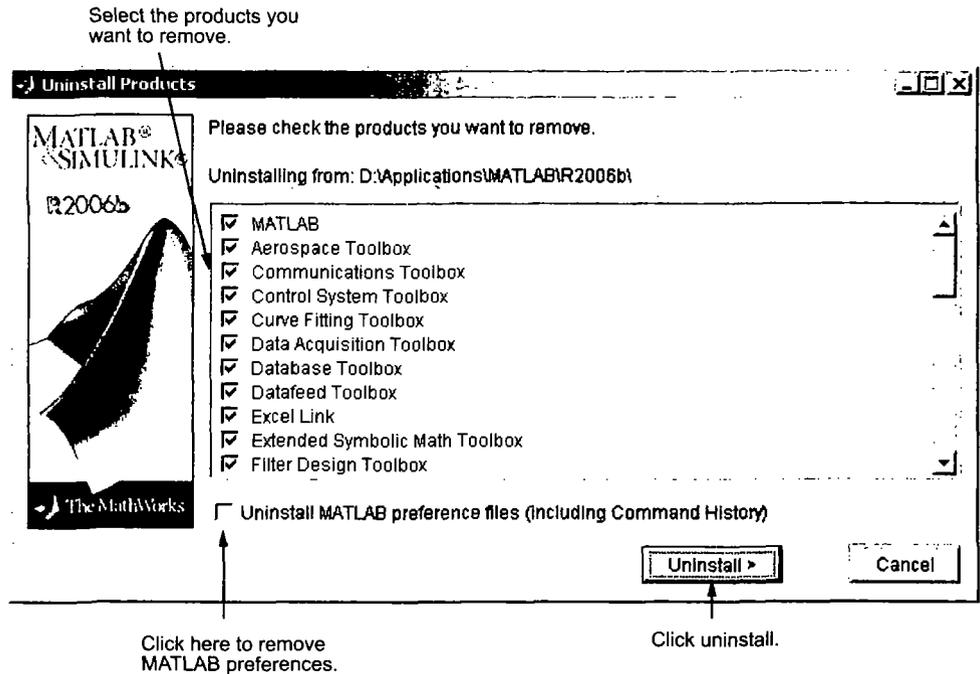
- 1 Shut down MATLAB before trying to uninstall it.
- 2 Start the uninstaller.

Click the **Start** button, select **Programs**, and move the cursor over the **MATLAB** entry. On the application menu that appears, click the entry for the uninstaller.



Alternatively, you can remove an installation using the Windows **Add/Remove Programs** control panel. Select **MATLAB R2006b** from the list of products.

Whatever method you use, when it starts, the uninstaller displays the Uninstall Product List dialog box.



- 3 Select the products you want to remove by selecting the check box next to their names. If you select MATLAB, you automatically remove all other MathWorks products as well.

Note You might receive a message telling you that the product you want to uninstall has other products that depend on it. For example, all MathWorks products require MATLAB. To dismiss the message and continue removing products, Click **OK**. To change the products you had selected for deletion, click **Cancel**. For more information, see “Product Dependencies” on page 4-4.

- 4 (Optional) Specify whether you want to remove your MATLAB preferences file.

The MathWorks stores your MATLAB environment preferences, such as command history and Help favorites, in a set of preference files. These

preference files are stored in a system folder that is associated with a particular user. To find where your preferences are located, use the `prefdir` command at the MATLAB prompt, as in the following example.

```
prefdir
ans =
C:\WINNT\Profiles\username\Application Data\MathWorks\MATLAB\R2006b
```

where *username* is the username of the person performing the uninstall.

By default, the uninstaller does not delete these preference files. If you want to remove these files, select the **Uninstall MATLAB preferences** check box. When you make this selection, the uninstaller deletes the contents of your MATLAB preferences folder.

5 To proceed with the removal, click the **Uninstall** button.

The uninstaller leaves the MATLAB installation folder on your system, empty except for the `\uninstall` folder, the `\work` folder if it contains files, and, if you downloaded product updates, an `\archives` folder that contains the download archive files. To completely remove a MATLAB installation, you must delete these folders yourself. You might want to move any M-files you created in the `\work` folder to another folder before deleting it.

Running the Uninstaller Noninteractively

You can run the uninstaller noninteractively. In this mode, you create an uninstaller initialization file, specifying the root folder of the installation you want to remove and the products you want to remove.

To run the uninstaller noninteractively, perform these steps:

1 Create an uninstaller initialization file.

The MathWorks provides a template for this file, named `uninstaller.ini`, located in the `uninstall` folder in your MATLAB installation folder. Make a copy of this template file, giving it any name you want.

```
copy matlabroot\uninstall\uninstaller.ini
C:\temp\my_uninstall.ini
```

- 2 Open your copy of the initialization file, using any text editor, and enter all the required information. For example, use the `directory` parameter to specify the name of the root folder of the installation you want to remove.

```
directory=C:\Program Files\MATLAB\R2006b
```

In this initialization file, you can specify the name of the products you want to remove and other parameters. By default, the uninstaller removes all products. If your initialization file is missing a required parameter, the uninstaller stops and writes an error to the output log. Two optional parameters you might want to set are

- **Controlling visibility** -- By default, when you run the uninstaller in silent mode, the uninstaller still displays the dialog box, even though no interactive response is required. If you want to suppress the display of this dialog box, set the value of the `visible` parameter to `false`.

```
visible=false
```

- **Requesting an output log** -- To request that the uninstaller create an output log file that reports on the status of removing an installation, specify the full pathname as the value of the `outlog` parameter.

```
outlog=C:\temp\my_uninstall.log
```

- 3 Save your changes to the file.
- 4 Run the uninstaller (`uninstall.exe`), using the `-if` flag to specify the initialization file as a command-line argument.

For example, click the **Start** button and select the **Run** option. In the Run dialog box, enter the `uninstall` command, specifying the full pathname of the initialization file as a command-line argument.

```
uninstall.exe -if C:\temp\my_uninstall.ini
```

Removing Installations That Use Dongles

In some installations, MATLAB requires a *dongle*. A dongle is a hardware-based security device that attaches to the USB port on your computer. It uses embedded codes and passwords to control access to software applications.

If your system uses dongles, the uninstaller asks if you want to delete the dongle drivers. If other applications you run on your system use dongles, do not delete these drivers. For more information, see “Troubleshooting an Installation That Uses Dongles” on page 4-18.

Troubleshooting

This chapter provides information that can help you solve common problems that you might encounter installing MATLAB on a PC running Microsoft Windows.

- | | |
|---|--|
| Problems During Installation (p. 4-2) | Answers to some common problems encountered during a MATLAB installation |
| Problems After Installation (p. 4-5) | Answers to some common problems encountered after a successful installation, including information about common license manager errors |
| Troubleshooting an Installation That Uses Dongles (p. 4-18) | Diagnosing problems with a MATLAB installation that uses dongles |

If you cannot find a description here that matches your problem, check the MATLAB Release Notes documentation that you received with your MATLAB products. This document contains the most up-to-date information available about products and installation procedures. Installation troubleshooting information is also available at the MathWorks Web site on the Support page, www.mathworks.com.

Problems During Installation

This section provides answers to some problems commonly encountered during installation.

- “Installer Does Not Start Automatically” on page 4-2
- “Invalid Personal License Password” on page 4-3
- “License File Could Not Be Updated” on page 4-3
- “Failure Overwriting an Existing Toolbox” on page 4-4
- “Product Dependencies” on page 4-4

Note If the MathWorks Installer appears to hang, you may want to turn off any virus checking software you are running on your system and restart the installer.

Installer Does Not Start Automatically

The installer should start automatically when you insert the MathWorks CD into your CD-ROM drive or when you double-click on the installer file you downloaded from the MathWorks Web site.

If the installer does not start, open a Windows Explorer window, select your CD-ROM drive, and double-click the setup.exe program in the top-level folder.

Invalid Personal License Password

During installation, you must enter your PLP. If your PLP is invalid, you cannot continue with the installation. Verify that

- You entered your PLP in “Step 3: Enter Identification and License Information” on page 1-6. Do not confuse your License Number with your PLP. A PLP is a series of five-digit numbers, prefixed by a two-digit code, such as:

15-12345-12345-12345-12345

- You used the PLP for the version of MATLAB you are trying to install. PLPs are version specific. Do not use an old PLP with a new version of MATLAB.

License File Could Not Be Updated

You can receive this error message when you are using the installer to update your licenses -- see “Updating a License” on page 1-23.

There are two reasons why you received this message:

- The license file (`license.dat`) in the `matlabroot\R2006b\bin\win32` folder is invalid.
- You have multiple valid license files in the specified folder (e.g., `demo.lic` and `license.dat`).

For more information about License Files that might help you troubleshoot these issues, see “Personal License Passwords (PLPs) and License Files” on page 3-4.

Failure Overwriting an Existing Toolbox

You can receive this error message when you attempt to perform an installation when an existing MATLAB installation is running. Make sure that you exit from all existing copies of MATLAB before attempting to run the installation procedure.

Product Dependencies

After clicking **Next** in the Product Selection dialog box, you might receive a message telling you that one or more of the products that you selected for installation have dependencies on other products that were not selected. For example, the following message indicates that certain products require the Aerospace Toolbox. To dismiss the message and continue with the installation, click **OK**. For a Custom installation, if you want to adjust your product selections, click **Cancel**.

Problems After Installation

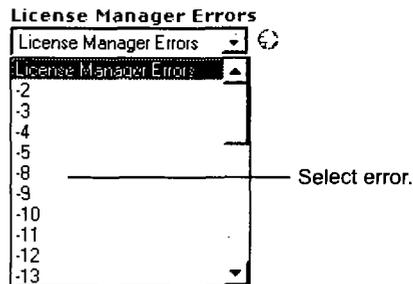
After a successful installation, you may encounter problems starting MATLAB. Many of these problems originate in the license manager (FLEXnet) used by the MathWorks products. This section describes

- “Getting Help with License Manager Problems” on page 4-5
- “License Manager Error Messages” on page 4-6

Getting Help with License Manager Problems

When a problem occurs, the license manager displays an error message in a window on your display and also writes the message to the FLEXnet log file, `lmlog.txt`, in the `matlabroot\#2006b\flexlm` folder. Read the error message carefully; it provides information that can help solve the problem. For information about viewing the license manager log file, see “Viewing the License Manager Log File” on page 3-17.

The section “License Manager Error Messages” on page 4-6 contains brief descriptions of many common error messages with recommended solutions. For more complete diagnostic information about an error message, go to the MathWorks Web site, www.mathworks.com, and click **Support**. The Support page includes a Search facility that lets you search the support database for help with an error message by error number. You can also select the error number from the **License Manager Errors** menu.



You can also get information about license manager error messages in the *FLEXnet Licensing End User Manual*, available in PDF format in `matlabroot\R2006b\flexlm\enduser.pdf` or at the Macrovision Corporation Web site, www.macrovision.com.

License Manager Error Messages

This section provides brief descriptions of some common license manager errors, listed in order by error number. For more complete diagnostic information about an error message, go to the MathWorks Web site, www.mathworks.com, and click Support. See “Getting Help with License Manager Problems” on page 4-5 for more information.

- “License Manager Error -2: License Checkout Failed. Invalid License File Syntax” on page 4-7
- “License Manager Error -3: License Checkout Failed. Cannot Connect to a License Server” on page 4-8
- “License Manager Error -5: License Checkout Failed. No Such Feature Exists” on page 4-9
- “License Manager Error -8: License Checkout Failed. Encryption Code in License File Is Inconsistent” on page 4-9
- “License Manager Error -9: License Checkout Failed. Invalid Host” on page 4-10
- “License Manager Error -10: License Checkout Failed. Feature Has Expired” on page 4-10
- “License Manager Error -12: License Checkout Failed. Invalid Return Data from License Server” on page 4-11
- “License Manager Error -13: License Checkout Failed. No SERVER Line in License File” on page 4-12
- “License Manager Error -14: License Checkout Failed. Cannot Find SERVER Hostname in Network Database” on page 4-13
- “License Manager Error -15: License Checkout Failed. Cannot Connect to License Server” on page 4-13
- “License Manager Error -16 License Checkout Failed. Cannot Read Data from License Server” on page 4-13

- “License Manager Error -18: License Checkout Failed. License Server Does Not Support This Feature” on page 4-14
- “License Manager Error -21: License Checkout Failed. License File Does Not Support This Version” on page 4-14
- “License Manager Error -25: License Checkout Failed. License Server Does Not Support This Version of This Feature” on page 4-15
- “License Manager Error -31: License Checkout Failed. Feature Not Yet Available” on page 4-15
- “License Manager Error -39: User/Host Not on INCLUDE List for Feature” on page 4-15
- “License Manager Error -83: License Checkout Failed. FLEXnet Version of Client Newer Than Server” on page 4-15
- “License Manager Error -84: USER_BASED License Has No Specified Users” on page 4-16
- “License Manager Error -96: License Checkout Failed. Server Node is Down or Not Responding” on page 4-16
- “License Manager Error -97: License Checkout Failed” on page 4-17

License Manager Error -2: License Checkout Failed. Invalid License File Syntax

This error typically indicates that there is a problem in your License File. On client systems, make sure your License File (*matlabroot\R2006b\bin\win32\license.dat*) has the following format.

```
SERVER host hostID port_num  
USE_SERVER
```

Note that there should be a carriage return immediately following the USE_SERVER line.

License Manager Error -3: License Checkout Failed. Cannot Connect to a License Server

This error can indicate that the SERVER or DAEMON lines in your License File is incorrect or missing. To remedy this error, try the following:

Note If you make any changes to the License File on the license server, you must restart the license manager.

- Make sure that the SERVER line in your License File matches the SERVER line in the License File on your license server. On client systems, the License File is `matlabroot\R2006b\bin\win32\license.dat`; on your license server, the License File is `matlabroot\R2006b\flexlm\license.dat`, where `matlabroot` is the root MATLAB installation folder.
- Make sure that your License File includes a DAEMON line. See “DAEMON Line” on page 3-7 for more information.
- Make sure the TCP port number used by the license manager is not already being used by another application. This port number appears at the end of the SERVER line in your License File, as in the following.

```
SERVER hostname INTERNET=128.36.2.12 27000
```

To determine if another application is using this port number, use the `netstat` command. On Windows 2000 and Windows XP systems, run

```
C:\WINNT\system32\netstat -p tcp
```

If you find the port number used by the license manager in the information returned by `netstat`, the port is in use. Change the port number on the SERVER line in your License File to a port that is not already in use. By default, FLEXnet uses a port number in the range 27000 to 27009.

- Make sure that you do not have multiple MathWorks vendor daemons (`mlm.exe`) running. Check your FLEXnet log file (`matlabroot\R2006b\flexlm\lmlog.txt`) for the error Multiple MLM Servers Running.

License Manager Error -5: License Checkout Failed. No Such Feature Exists

This error usually indicates that you are trying to use a product for which you are not licensed.

Note You must restart the license manager after making any changes to the License File on the license server.

- Check your License File to see if you are licensed to use this product. Your License File should contain an INCREMENT line for each licensed product
- Make sure that your License File does not contain extra carriage returns or tabs. To avoid this, make sure that each line begins with either #, SERVER, DAEMON, or INCREMENT and replace any tab characters in INCREMENT lines with spaces. One way to check for extraneous tabs and carriage returns is to open your License File in a text editor and view all nonprinting characters. Remove these tabs and replace with them spaces.
- Make sure that your system date is set to a date after the date that your license was made. See “License Manager Error -10: License Checkout Failed. Feature Has Expired” on page 4-10 to learn how to reset the system date.

License Manager Error -8: License Checkout Failed. Encryption Code in License File Is Inconsistent

This error typically indicates that you are running the license manager on a host that has a different host ID than the one for which your License File was generated.

Note You must restart the license manager after making any changes to the License File on the license server.

- Verify that the host ID specified in the License File matches the system’s host ID. See “Finding the Host ID on a Windows System” on page 1-29.

- Make sure that your License File does not contain extra carriage returns or tabs. To avoid this, make sure that each line begins with either #, SERVER, DAEMON, or INCREMENT and replace any tab characters in INCREMENT lines with spaces. One way to check for extraneous tabs and carriage returns is to open your License File in a text editor and view all nonprinting characters. Remove these tabs and replace with them spaces.
- Make sure that the license information in your License File matches the license information The MathWorks sent you.

License Manager Error -9: License Checkout Failed. Invalid Host

If you get this error on a system running an individually licensed MATLAB, this error typically indicates that you are not running MATLAB on the same system it was installed on. You can also get this error if you

- Replaced your hard drive
- Changed the partition on an existing hard drive
- Do not have a C:\ drive
- Tried to clone an installation from one system to another

To remedy this situation, rerun the installer. At “Step 2: Start the Installer” on page 1-5, select the **Update license** option. This option lets you update your license without requiring that you reinstall MATLAB.

License Manager Error -10: License Checkout Failed. Feature Has Expired

This error indicates that one or more of your license keys has expired.

Verify that the expiration date on each of the INCREMENT lines in your License File (*matlabroot\R2006b\bin\win32*) is still current. If the dates have already passed, contact the Customer Service department at The MathWorks (through the Web or through e-mail at service@mathworks.com). When contacting The MathWorks, include your License Number and the Host ID of your license server.

If none of your product licenses has expired, verify that the system date and time on your computer are correct:

- 1 Click the **Start** menu.
- 2 Choose **Settings**.
- 3 Click **Control Panel**.
- 4 Choose **Date/Time**.

License Manager Error -12: License Checkout Failed. Invalid Return Data from License Server

This error can be caused by your network setup. When the license manager tries to connect to the server, the DNS Server does not find the server because it does not add the domain name suffix into the search.

If all of your client systems are having this problem, change the name of the server in the `SERVER` line in the License File on client systems (located in `matlabroot\R2006b\bin\win32`) to the fully qualified name and restart the license manager on the server. (See “Starting and Stopping the License Manager” on page 3-15.)

If only some of the clients in your network installation are having this problem, add the hostname of your license server to the TCP/IP Hosts file on your client systems:

- 1 Open the file, `C:\WINNT\system32\drivers\etc\hosts`
- 2 Add the hostname of your license server to this file. You can find your license server information in the `SERVER` line of your `license.dat` file.

For example, if your `SERVER` line looks like this,

```
SERVER hostname INTERNET=128.36.2.12 27000
```

Add the following line to your TCP/IP hosts file.

```
128.36.2.12 hostname
```

- 3 Save the changes to the TCP/IP hosts file.

License Manager Error -13: License Checkout Failed. No SERVER Line in License File

This error indicates that your License File is missing the `SERVER` line, `DAEMON` line, or both. These lines should be added to the License File by the MathWorks Installer during the installation process.

To remedy this problem, run the installer again or manually edit the License File, `matlabroot\R2006b\flexlm\license.dat`, adding the `SERVER` line. See “Personal License Passwords (PLPs) and License Files” on page 3-4 for information about the format and contents of the `SERVER` and `DAEMON` lines.

Before modifying your License File, verify that the host ID in your license server matches the host ID given in the commented line of the file. If your system’s host ID does not match the host ID used to generate your License File, contact The MathWorks; you will need your license regenerated for the correct host ID.

Note You must restart the license manager after making any changes to the License File on the license server.

License Manager Error -14: License Checkout Failed. Cannot Find SERVER Hostname in Network Database

If this error occurs on a client system, make sure that the hostname on the SERVER line in the `license.dat` file on your client system matches the hostname on the SERVER line in the `license.dat` file on your license server. Edit the SERVER line in the client `license.dat` file, if necessary.

Also, make sure that the server's hostname included in the SERVER line is listed in the `C:\WINNT\system32\drivers\etc\hosts` file.

License Manager Error -15: License Checkout Failed. Cannot Connect to License Server

This error indicates that MATLAB cannot connect to the license server.

- Verify that both license manager daemons, `lmgrd` and `mlm`, are running on the server. You can check this via the Task Manager, which can be accessed by pressing the **Ctrl+Alt+Delete** key sequence. If the license manager isn't running, see "Starting and Stopping the License Manager" on page 3-15 for information about starting it.
- Verify that you have read permission to your MATLAB installation folder.

For additional suggestions, go to the Support page on the MathWorks Web site, www.mathworks.com and search for this license manager error number.

License Manager Error -16 License Checkout Failed. Cannot Read Data from License Server

This error indicates that the path to the MATLAB vendor daemon on the DAEMON line in your License File is incorrect. The DAEMON line can vary depending on your license type. See "Personal License Passwords (PLPs) and License Files" on page 3-4 for an examples.

License Manager Error -18: License Checkout Failed. License Server Does Not Support This Feature

This error typically indicates that you have updated your License File without restarting the license manager. Restart the license manager and try to start MATLAB again.

If you get this error on the client, but not the server, the `license.dat` file on the client probably needs to be updated.

Make sure that the INCREMENT line in your License File exactly matches the INCREMENT line in the license information you received from The MathWorks.

License Manager Error -21: License Checkout Failed. License File Does Not Support This Version

If you are getting this error on your license server, make sure you have installed the version of FLEXnet included with the release you are trying to install. Older FLEXnet versions will not work with the License Files for new releases.

If you have updated an existing `license.dat` file with new passcodes, you still need to install the latest version of FLEXnet included with the release.

If you are getting this error on one of your client systems and not on the license server, make sure the License File on the client is in the following format.

```
SERVER host hostname port_num  
USE_SERVER
```

The SERVER line in the client system License File should be the same as the SERVER line found in the License File on the license server.

License Manager Error -25: License Checkout Failed. License Server Does Not Support This Version of This Feature

This error indicates that you are trying to run MATLAB with an old license. You must update your license.dat file with your new license passcodes.

If you do not have an up-to-date license, contact Customer Service at The MathWorks.

License Manager Error -31: License Checkout Failed. Feature Not Yet Available

This error indicates that the system date is set to a date that is earlier than the date MATLAB was installed. Make sure that your system date is set to a date after the date that your license was made. See “License Manager Error -10: License Checkout Failed. Feature Has Expired” on page 4-10 to learn how to reset the system date.

License Manager Error -39: User/Host Not on INCLUDE List for Feature

This error can indicate that your FLEXnet options file contains an incorrect username on the INCLUDE line. For Stand-Alone Named User and Network Named User installations, the FLEXnet options file requires an INCLUDE line for all licensed products. See “Setting Up Network Named User Licensing” on page 2-25 for more information.

In addition, make sure that each user named in the FLEXnet options file has read permission for the options file.

License Manager Error -83: License Checkout Failed. FLEXnet Version of Client Newer Than Server

This error usually indicates that the version of the license manager on the client system is newer than the license manager on the server system. When MATLAB starts, the license file points to the wrong version of the license manager. To remedy this problem, update the license manager on the server to the latest version of FLEXnet that is included on the MathWorks CD.

License Manager Error -84: USER_BASED License Has No Specified Users

The license manager cannot find your options file. Some possible resolutions are

- Verify that the installer created an options file during the installation procedure. The full path of the options file is *matlabroot\R2006b\flexlm\MLM.opt*, where *matlabroot* is your MATLAB installation folder. See “Setting Up Network Named User Licensing” on page 2-25 for more information.
- Verify that the DAEMON line in your License File includes the full pathname of your options file. Check the processed License File, *matlabroot\R2006b\flexlm\license.dat*. If the DAEMON line wraps onto a second line, make sure there is a line continuation character (\) at the end of the first line.
- Verify that you are listed as a valid user of the product you are trying to use.

License Manager Error -96: License Checkout Failed. Server Node is Down or Not Responding

This error can occur if you are running DHCP (to obtain a dynamically assigned IP address) on your license server. MATLAB cannot connect with the license manager because it uses TCP/IP protocols for communication and the IP address is changing.

To fix this problem, follow this procedure:

- 1** Obtain the current IP address for your server. If you are on UNIX or Linux, use the `ifconfig` command. On a PC running Windows, you can use the `ipconfig` command at a DOS prompt.
- 2** Edit the hosts file on your system and change the IP address for your server’s entry to be the correct IP address. On UNIX or Linux systems, edit the `/etc/hosts` file. On Windows systems, edit the `C:\WINNT\system32\drivers\etc\hosts` file.

License Manager Error -97: License Checkout Failed

This error can occur when trying to start the license manager while an existing vendor daemon (`mlm.exe`) is already running. To resolve this, open the Task Manager and terminate any `mlm.exe` processes that are running. If you cannot end the process in the Task Manager, reboot your computer. Once you reboot, restart the license manager.

If the same error occurs after rebooting, it is also possible that the vendor daemon is trying to run on an occupied port. To specify an open port for the vendor daemon, edit the `DAEMON` line in the `license.dat` file, found in the `matlabroot\R2006b\flexlm` folder, where `matlabroot` is the root MATLAB folder

```
DAEMON MLM matlabroot\R2006b\flexlm\mlm.exe port=#####
```

where ##### is an open port. You can use any unused port number between 0 and 64000. By default, the license manager uses a port in the range 27000 to 27009.

Troubleshooting an Installation That Uses Dongles

If your MATLAB installation requires the use of dongles, it can potentially run into conflicts with other applications you may be running that also use dongles. A dongle is a hardware-based security device that attaches to the USB port of a computer.

To use a dongle, MATLAB requires a specific set of dongle drivers. If you remove other applications from your computer, these applications may delete the MATLAB dongle drivers as well as their own.

To help troubleshoot problems with dongles, MATLAB includes a utility program called `dongleutil.exe`. The dongle utility program can be used to

- Determine the version of the currently installed dongle drivers. The MathWorks Technical Support might request this information if you call.
- Specify the minimum version of the dongle drivers required by MATLAB.
- Install the dongle drivers required by MATLAB.

If an installation uses dongles, the installer includes the `dongleutil.exe` utility in the `matlabroot\R2006b\bin\win32` folder, where `matlabroot` represents your MATLAB installation folder.

A

adding products 1-26

C

client systems

 accessing MATLAB 2-4

 in redundant configurations 3-27

 location of processed License File 3-8

concurrent access configurations 2-3

concurrent installations

 procedure

 on client systems 2-28

 time-outs 3-18

configuration

 after installation 1-21

contacting The MathWorks 1-28

current folder

 initial 1-21

D

DAEMON line

 in License Files 3-7

daemons

 license manager daemon 3-3

 opening access in security firewalls 2-27

 vendor daemon 3-3

deleting a MATLAB installation 3-30

demo licenses

 converting to standard licenses 1-23

dongles

 troubleshooting 4-18

 uninstalling 3-34

dongleutil.exe 4-18

E

error messages

 license manager 4-6

expiration date

 extending 1-23

F

firewalls

 opening access to license manager

 daemons 2-27

FLEXnet

 installation folder 3-3

 troubleshooting 4-5

H

host ID

 determining on license server 1-29

 in License File 3-4

hosts file 4-16

I

INCREMENT lines

 in License Files 3-4

initial current folder 1-21

initialization files

 using with installer 3-28

 using with uninstaller 3-32

install.exe

 installing license manager as a
 service 3-14

installation folder

 moving 1-23

installation procedure

 basic 1-2

 network 2-5

 noninteractive 3-28

 prerequisites

 concurrent licenses 2-6

 individual licenses 1-3

 selecting products

 concurrent installations 2-15

- turning off virus checking software 4-2
- installer
 - concurrent installations 2-8
 - running noninteractively 3-28
 - standard installations 1-5
 - updating a license 1-23
- installer initialization file
 - specifying 3-28
- installer.ini 3-28
- installing MATLAB
 - adding products 1-26
 - basic procedure 1-2
 - network procedure 2-4
- ipconfig 1-29

L

- License Files
 - checking line breaks in 3-5
 - combining licenses 1-28
 - comment lines 3-4
 - creating 3-5
 - DAEMON line 3-7
 - definition 3-4
 - encryption code inconsistency 4-9
 - INCREMENT lines 3-4
 - location on client system 3-8
 - processing by installer 2-11
 - processing on servers 3-6
 - running separate license servers 3-24
 - search path 3-8
 - SERVER line 3-7
 - sharing with other applications 3-24
 - specifying path 2-11
 - understanding 3-4
 - with multiple SERVER lines 3-26
- license management
 - advanced configurations 3-21
 - integrating with an existing FLEXnet
 - license installation 3-24
 - overview 3-2
- license manager
 - administering 3-10
 - administration 3-2
 - daemons 3-3
 - error messages 4-6
 - installation folder 3-3
 - installing service 3-14
 - removing license manager service 3-15
 - running multiple 3-24
 - running with other applications 3-24
 - sharing with other applications 3-24
 - starting 3-15
 - synchronizing in redundant
 - configurations 3-27
 - troubleshooting 4-5
 - using the LMTTOOLS GUI 3-12
 - utility programs 3-10
 - viewing log file 3-17
- license manager daemon
 - port numbers 3-7
- License Number 1-28
- License Password
 - See Personal License Password 1-6
- license servers
 - defined 2-3
 - redundant 3-24
- license.dat
 - creating 3-6
 - definition 3-4
- license.txt
 - software license agreement text 1-8
- licenses
 - administration tools 3-10
 - extending expiration date 1-23
 - managing 1-28
 - types of 1-1
 - user-based 2-25
- licensing
 - troubleshooting 4-5

- using concurrent licenses with a stand-alone installation 3-22
- viewing the log file 4-5
- LM_LICENSE_FILE environment variable 3-8
- lmdiag
 - diagnosing license checkout problems 3-11
- lmdown
 - shutting down the license manager 3-11
- lmgrd.exe 3-3
 - license manager daemon 3-3
- lmhostid
 - getting system host ID 3-11
- lmlog.txt
 - FLEXnet log file 3-17
 - license manager log file 4-5
- lmreread
 - forcing license manager to reprocess the License File 3-11
- lmstat
 - viewing license manager status 3-11
- lmswitchr
 - specifying the report log file 3-11
- LMTTOOLS
 - using the FLEXnet utilities GUI 3-12
- lmttools.exe
 - FLEXnet utilities GUI 3-12
- lmutil.exe
 - license manager utilities 3-10
- lmver
 - get FLEXnet version number 3-11
- local client access 2-4
- log files
 - license manager 4-5
 - viewing license manager log file 3-17

M

MATLAB

- getting started 1-22

installing

- basic procedure 1-2
- in network configurations 2-4
- moving an installation 1-23
- removing an installation 3-30
- running multiple versions
 - on one system 3-21
 - troubleshooting 4-17

starting 1-20

MATLAB Installer

- concurrent installations 2-8
- standard installation 1-5
- updating a license 1-23

MATLAB Notebook

- configuration 1-21

mlm.exe

- vendor daemon 3-3

MLM_LICENSE_FILE environment variable 3-8

MLM.opt

- FLEXnet options file 2-26

N

network installation procedure

- installing MATLAB in network configurations 2-5
- instructions 2-1
- on client systems 2-28
- post-installation tasks 2-25

Notebook

- configuration 1-21

O

options files

- enabling time-outs 3-18
- specifying location of 3-7
- with named user licensing 2-26

overwriting an existing installation

- individual installation 1-4 2-7

P

- passcode
 - encrypted in license file 3-4
- password
 - See Personal License Password 1-28
- Personal License Password (PLP)
 - compared to License Number 1-3
 - definition 3-4
 - installing additional products 1-26
 - specifying
 - concurrent installations 2-9
 - standard installations 1-6
 - when updating a license 1-24
- port numbers
 - changing 4-17
 - opening access in security firewalls 2-27
 - specifying in License File 3-7
 - specifying on DAEMON line 3-7
 - used for license manager 3-7
 - vendor daemon 3-8
- preference files
 - removing 3-31
- products
 - adding 1-26
 - selecting
 - concurrent installations 2-15

R

- Real-Time Windows Target
 - configuration 1-21
- redundant license servers 3-24
 - adding SERVER lines for each server 3-26
 - setting up client systems 3-27
- remote client access
 - setting up 2-3
- removing a MATLAB installation 3-30
- requirements, *see* system requirements

S

- security firewalls
 - opening access to license manager daemons 2-27
- SERVER line
 - in License Files 3-7
 - sharing License Files 3-24
 - troubleshooting 4-12
- setup.exe
 - in noninteractive installations 3-28
 - standard installations 4-2
- shortcut
 - MATLAB shortcut on desktop 1-20
- silent installations
 - running without screen displays 3-28
- Simulink
 - behavior with license time-out 3-20
- software license agreement 1-8
- stand-alone licenses
 - using concurrently licensed products with 3-22
- start-up folder 1-21
- starting MATLAB 1-20
 - troubleshooting 4-5
- starting the license manager 3-15
- system requirements 1-30

T

- time-outs
 - enabling 3-18
 - of license keys 3-18
 - reacquiring timed out license keys 3-20
 - user experience of 3-20
- toolboxes
 - adding 1-26
 - transferring licenses 1-28
- troubleshooting
 - during installation 4-2
 - FLEXnet 4-5

starting MATLAB 4-5

U

uninstall.exe

 using the -if option 3-33

uninstaller

 running noninteractively 3-32

 starting 3-30

uninstalling MATLAB 3-30

updating a license 1-23

upgrading an installation

 network installations 2-7

 standard installations 1-4

USE_SERVER

 client License File 3-8

user-based licensing

 setting up 2-25

V

vendor daemon 3-3

 specifying port numbers 3-8

virus checking software

 disabling during network installations 2-7

 disabling in a standard installation 1-3

 troubleshooting installations 4-2

W

work folder 1-21

www.mathworks.com

The MathWorks, Inc.
3 Apple Hill Drive
Natick, MA 01760-2098 USA
Phone: 508-647-7000

MATLAB® & SIMULINK®

Product Families

R2006a

- Computation
- Visualization
- Programming
- Modeling
- Simulation
- Implementation

Release Notes for
Release 2006a

How to Contact The MathWorks:



www.mathworks.com
comp.soft-sys.matlab
www.mathworks.com/contact_TS.html

Web
Newsgroup
Technical support



suggest@mathworks.com
bugs@mathworks.com
doc@mathworks.com
service@mathworks.com
info@mathworks.com

Product enhancement suggestions
Bug reports
Documentation error reports
Order status, license renewals, passcodes
Sales, pricing, and general information



508-647-7000 (Phone)



508-647-7001 (Fax)



The MathWorks, Inc.
3 Apple Hill Drive
Natick, MA 01760-2098

For contact information about worldwide offices, see the MathWorks Web site.

Release Notes for Release 2006a

© COPYRIGHT 2006 by The MathWorks, Inc.

The software described in this document is furnished under a license agreement. The software may be used or copied only under the terms of the license agreement. No part of this manual may be photocopied or reproduced in any form without prior written consent from The MathWorks, Inc.

FEDERAL ACQUISITION: This provision applies to all acquisitions of the Program and Documentation by, for, or through the federal government of the United States. By accepting delivery of the Program or Documentation, the government hereby agrees that this software or documentation qualifies as commercial computer software or commercial computer software documentation as such terms are used or defined in FAR 12.212, DFARS Part 227.72, and DFARS 252.227-7014. Accordingly, the terms and conditions of this Agreement and only those rights specified in this Agreement, shall pertain to and govern the use, modification, reproduction, release, performance, display, and disclosure of the Program and Documentation by the federal government (or other entity acquiring for or through the federal government) and shall supersede any conflicting contractual terms or conditions. If this License fails to meet the government's needs or is inconsistent in any respect with federal procurement law, the government agrees to return the Program and Documentation, unused, to The MathWorks, Inc.

Trademarks

MATLAB, Simulink, Stateflow, Handle Graphics, Real-Time Workshop, and xPC TargetBox are registered trademarks of The MathWorks, Inc. Other product or brand names are trademarks or registered trademarks of their respective holders.

Patents

The MathWorks products are protected by one or more U.S. patents. Please see www.mathworks.com/patents for more information.

Revision History

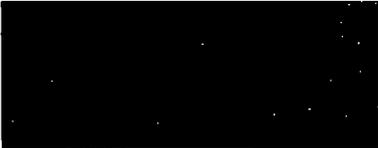
March 2006 First printing New for Release 2006a

Release Notes for R2006a

Introduction	2
Where to Find Information About This Release	2
Release Notes for R2006a	2
Summary of Product Updates	4
Which Products Are Listed	4
Explanation of Table Columns	4
Release Summary	6
Summary of R2006a New Features	10
MATLAB	11
Simulink	12
Aerospace Blockset	13
Bioinformatics Toolbox	13
Communications Blockset	14
Communications Toolbox	14
Control System Toolbox	14
Datafeed Toolbox	15
Distributed Computing Toolbox and MATLAB Distributed Computing Engine	15
Embedded Target for Motorola MPC555	15
Embedded Target for TI C2000 DSP	15
Embedded Target for TI C6000 DSP	16
Filter Design HDL Coder	17
Filter Design Toolbox	17
Financial Toolbox	17
Fixed-Point Toolbox	18
GARCH Toolbox	18
Image Acquisition Toolbox	18
Image Processing Toolbox	18
Instrument Control Toolbox	19
Link for Code Composer Studio Development Tools	19
Link for ModelSim	19

Mapping Toolbox	19
MATLAB Compiler	20
Neural Network Toolbox	20
OPC Toolbox	21
Real-Time Workshop	21
Real-Time Workshop Embedded Coder	21
RF Blockset	22
RF Toolbox	22
Signal Processing Blockset	22
Signal Processing Toolbox	22
SimDriveline	22
SimEvents	23
SimMechanics	23
SimPowerSystems	23
Simulink Control Design	23
Simulink Fixed Point	24
Simulink Report Generator	24
Simulink Response Optimization	24
Spline Toolbox	24
Stateflow and Stateflow Coder	24
Statistics Toolbox	25
System Identification Toolbox	25
Video and Image Processing Blockset	26
Virtual Reality Toolbox	26
xPC Target	26
New Products	27
MATLAB Builder for .NET	27
SimBiology	28
SimEvents	28
MathWorks Products in Transition in R2006a	29
MATLAB Builder for COM	29
Financial Time Series Toolbox	29
Upgrading from a Previous Release	30
Compatibility Considerations in This Release	30
Compatibility Considerations for New Functions and Models ..	32

System Requirements	34
All Platforms	34
Microsoft® Windows®	34
Mac OS® X	35
Linux	35
UNIX (Solaris)	36
Accessibility Notes	37
Products Updated	37
Summary of Accessibility Support	37
Accessibility Support Details	38
Assistive Technologies	39
Troubleshooting	39
Documentation	41
Installation Notes for Accessibility Support	43
Quick Access to Product-Specific Release Notes	46
MATLAB Software Acknowledgments	49



Release Notes for R2006a

Note The HTML version of the release notes includes many links to other MathWorks documentation and to The MathWorks Web site. If you are reading these release notes in printed or PDF format, we recommend viewing the HTML version if you wish to access more detailed information that is linked to from these release notes.

Introduction

Release 2006a (R2006a) includes many new features and two new products. The major focus of R2006a is on improving the quality of the MathWorks products.

Where to Find Information About This Release

The following describe what's new in this release:

- Release Notes for R2006a (described below)
- Product-specific release notes (“Quick Access to Product-Specific Release Notes” on page 46 includes links to product-specific release notes)

To view important fixed and open bugs in R2006a, you can use the Bug Reports interface on The MathWorks Web site.

Release Notes for R2006a

These general release notes summarize the products updated for R2006a. Major topics are

- “Summary of Product Updates” on page 4
- “Summary of R2006a New Features” on page 10
- “New Products” on page 27
- “MathWorks Products in Transition in R2006a” on page 29
- “Upgrading from a Previous Release” on page 30
- “System Requirements” on page 34
- “Accessibility Notes” on page 37
- “Quick Access to Product-Specific Release Notes” on page 46

Highlights of Changes Since R14SP3

These general release notes highlight changes in products since Release 14 with Service Pack 3 (R14SP3). This includes products that updated for R14SP3 and products that have had a Web release since R14SP3.

If you are upgrading from a release earlier than R14SP3, you may want to refer to the Release Notes for Release 14 with Service Pack 3 (on the MathWorks Web site).

Printing the Release Notes for R2006a

If you are reading these general release notes for R2006a online and would like to print them, a PDF version is available on The MathWorks Web site.

Summary of Product Updates

The release summary table in this section summarizes the kinds of updates made to each product for R2006a.

R2006a includes three new products that were introduced since R14SP3. See “New Products” on page 27 for an introduction to these new products.

- MATLAB® Builder for .NET
- SimBiology
- SimEvents

Which Products Are Listed

Products are listed in the following sections *only* if they have significant updates for R2006a.

However, the release notes for products not listed in the following tables may discuss upgrade issues or other important information. See “Quick Access to Product-Specific Release Notes” on page 46 for links to all the product-specific release notes.

Explanation of Table Columns

The release summary table that follows has these columns:

Product

MATLAB® and Simulink® are listed first in the table. The rest of the MathWorks products are listed alphabetically.

An asterisk (*) after a product name indicates the product has had a Web release since R14SP3.

Product name links take you to the individual release notes for specific products.

Note The product-specific release notes are available in the online documentation for each product and on the MathWorks Web site.

New Features

The links in the “New Features” column take you to a summary of major new features for the product. For details about a product’s new features, see that product’s release notes.

Major Bug Fixes

A “Yes” in the “Major Bug Fixes” column indicates R2006a includes major bug fixes for the product.

Note To view important fixed and open bugs in R2006a, use the Bug Reports interface on the MathWorks Web site. If you are not already logged in to Access, you will be prompted to log in or create a MathWorks Account.

To view bug fixes introduced before R14SP2, see the product-specific release notes.

Compatibility Considerations

A “Yes” in the “Compatibility Considerations” column indicates that there are compatibility issues associated with upgrading from R14SP3 to R2006a. The “Yes” link takes you to the product-specific release notes, where the Compatibility Summary table provides more information.

Release Summary

Product (Links to Release Notes)	New Features	Major Bug Fixes	Compatibility Considerations
MATLAB®	Yes	Yes	Yes
Simulink®	Yes	Yes	Yes
Aerospace Blockset	Yes	Yes	No
Bioinformatics Toolbox*	Yes	No	No
Communications Blockset	Yes	Yes	Yes
Communications Toolbox	Yes	Yes	No
Control System Toolbox	Yes	No	
Data Acquisition Toolbox*	No	Yes	No
Datafeed Toolbox	Yes	No	No
Distributed Computing Toolbox*	Yes	Yes	No
Embedded Target for Motorola® MPC555	Yes	Yes	Yes
Embedded Target for TI C2000™ DSP	Yes	Yes	No
Embedded Target for TI C6000™ DSP	Yes	Yes	Yes
Filter Design HDL Coder	Yes	No	Yes
Filter Design Toolbox	Yes	Yes	No
Financial Toolbox	Yes	No	No
Fixed-Point Toolbox	Yes	Yes	Yes

Product (Links to Release Notes)	New Features	Major Bug Fixes	Compatibility Considerations
GARCH Toolbox	Yes	No	No
Gauges Blockset	No	Yes	No
Image Acquisition Toolbox*	Yes	Yes	No
Image Processing Toolbox	Yes	Yes	Yes
Instrument Control Toolbox	Yes	No	No
Link for Code Composer Studio™ Development Tools	Yes	No	Yes
Link for ModelSim®	Yes	Yes	No
Mapping Toolbox	Yes	Yes	Yes
MATLAB® Compiler	Yes	Yes	No
MATLAB® Distributed Computing Engine*	Yes	Yes	No
MATLAB® Report Generator	No	Yes	No
Model-Based Calibration Toolbox*	No	Yes	No
Neural Network Toolbox	Yes	No	Yes
OPC Toolbox	Yes	Yes	No
Optimization Toolbox	No	Yes	No
Partial Differential Equation Toolbox (no release notes)	No	Yes	No

Product (Links to Release Notes)	New Features	Major Bug Fixes	Compatibility Considerations
Real-Time Workshop®	Yes	Yes	Yes
Real-Time Workshop® Embedded Coder	Yes	Yes	Yes
RF Blockset	Yes	Yes	No
RF Toolbox	Yes	Yes	No
Robust Control Toolbox	No	Yes	No
Signal Processing Blockset	Yes	Yes	No
Signal Processing Toolbox	Yes	Yes	Yes
SimDriveline	Yes	Yes	No
SimEvents*	Yes	Yes	Yes
SimMechanics	Yes	Yes	No
SimPowerSystems	Yes	Yes	No
Simulink® Accelerator	No	Yes	No
Simulink® Control Design	Yes	Yes	No
Simulink® Fixed Point	Yes	No	No
Simulink® Report Generator	Yes	Yes	No
Simulink® Response Optimization	Yes	No	No
Simulink® Verification and Validation	No	Yes	Yes
Spline Toolbox	Yes	Yes	Yes

Product (Links to Release Notes)	New Features	Major Bug Fixes	Compatibility Considerations
Stateflow® and Stateflow® Coder	Yes	Yes	No
Statistics Toolbox	Yes	Yes	No
System Identification Toolbox	Yes	No	Yes
Video and Image Processing Blockset*	Yes	Yes	Yes
Virtual Reality Toolbox	Yes	Yes	No
xPC Target	Yes	Yes	Yes

Summary of R2006a New Features

This section summarizes the major new features and enhancements introduced in R2006a.

MATLAB and Simulink are described first. The rest of the updated MathWorks products are listed alphabetically.

The products summarized below are

- “MATLAB” on page 11
- “Simulink” on page 12
- “Aerospace Blockset” on page 13
- “Bioinformatics Toolbox” on page 13
- “Communications Blockset” on page 14
- “Communications Toolbox” on page 14
- “Control System Toolbox” on page 14
- “Datafeed Toolbox” on page 15
- “Distributed Computing Toolbox and MATLAB Distributed Computing Engine” on page 15
- “Embedded Target for Motorola MPC555” on page 15
- “Embedded Target for TI C2000 DSP” on page 15
- “Embedded Target for TI C6000 DSP” on page 16
- “Filter Design HDL Coder” on page 17
- “Filter Design Toolbox” on page 17
- “Financial Toolbox” on page 17
- “Fixed-Point Toolbox” on page 18
- “GARCH Toolbox” on page 18
- “Image Acquisition Toolbox” on page 18
- “Image Processing Toolbox” on page 18
- “Instrument Control Toolbox” on page 19
- “Link for Code Composer Studio Development Tools” on page 19
- “Link for ModelSim” on page 19
- “Mapping Toolbox” on page 19

- “MATLAB Compiler” on page 20
- “Neural Network Toolbox” on page 20
- “OPC Toolbox” on page 21
- “Real-Time Workshop” on page 21
- “Real-Time Workshop Embedded Coder” on page 21
- “RF Blockset” on page 22
- “RF Toolbox” on page 22
- “Signal Processing Blockset” on page 22
- “Signal Processing Toolbox” on page 22
- “SimDriveline” on page 22
- “SimEvents” on page 23
- “SimMechanics” on page 23
- “SimPowerSystems” on page 23
- “Simulink Control Design” on page 23
- “Simulink Fixed Point” on page 24
- “Simulink Report Generator” on page 24
- “Simulink Response Optimization” on page 24
- “Spline Toolbox” on page 24
- “Stateflow and Stateflow Coder” on page 24
- “Video and Image Processing Blockset” on page 26
- “Virtual Reality Toolbox” on page 26
- “xPC Target” on page 26

You can see a summary of all new functions and models introduced in R2006a in the documentation page on the MathWorks Web site.

MATLAB

- Desktop Tools and Development Environment
 - M-Lint code analyzer, now built into the Editor/Debugger, continuously checks your code for problems and recommends modifications to maximize performance and maintainability.

- Tab completion for functions and variables in the Editor/Debugger
- Ability to return to previous points in an M-file in the Editor/Debugger
- Mathematics
 - New `ddesd` function solves initial value problems for delay differential equations with general delays
 - Upgraded optimized Basic Linear Algebra Subprogram (BLAS) libraries for Intel processors on Windows and Linux platforms (MKL 8.0.1) and AMD processors on Linux platforms (ACML 2.7)
 - Improved performance of sparse Cholesky factorization with use of CHOLMOD 1.0 routines

Simulink

- Large-scale modeling
 - Simulink now detects and pinpoints excessive zero-crossing events to specific blocks in your model, to help you diagnose and fix performance problems in your model.
 - A new tiled printing feature lets you print a block diagram across multiple pages that can be taped together to create a large printout.
 - The Subsystem block context menu now includes a Convert to Model item that converts an atomic subsystem to a model reference.
 - The new Concatenate block concatenates its input signals to create a single output signal whose elements occupy contiguous locations in memory. The block typically uses less memory than the Matrix Concatenation block that it replaces.
 - Support for disabling the inline parameters option for the top model in a model reference hierarchy. This lets you tune the top model's parameters without sacrificing the simulation speed resulting from model referencing.
 - New command-line API for running the Model Advisor from MATLAB
 - Signal logs now preserve the hierarchy of bus data, which facilitates generation of test vectors for large models.
- Embedded systems design
 - Support for fixed-point values to specify the values of block parameters, which can improve the performance of generated code

- Support for using signal objects to initialize any signal or state, not just the outputs of blocks that have an initial condition or value parameter. This increases the reliability of code generated from the model.
- Simulink now checks to ensure that the values of signal and parameter objects fall within the ranges specified by their Minimum and Maximum properties.
- Greater usability:
 - Support for customizing any Simulink or Stateflow editor menu
 - New MATLAB desktop item on the Model Editor's View menu brings the MATLAB desktop to the top of your screen.
 - Double-clicking a Model block opens the model referenced by the block instead of the Model block's parameter dialog box as in previous releases.
 - You can now configure a Logical Operator block to have the same shape as the IEEE symbol for the logical operation that it performs. This lets you tell at a glance what option the block performs.

Aerospace Blockset

- Compatibility with FlightGear v0.9.9 flight simulator
- Documentation improvements

Bioinformatics Toolbox

The following features were added in Version 2.2 of the Bioinformatics Toolbox. Version 2.2 was released via the Web, after R14SP3. No additional new features were added in Version 2.2.1, which is included in R2006a.

- New GUI to interactively view, explore, and make manual modifications to multiple alignments
- Functions to read Agilent Feature Extraction Software files
- Functions to import the Gene Ontology database from the Web into MATLAB and extract a subset of the ontology

Communications Blockset

- Rician Channel block enhanced
- Channel visualization added to Multipath Rician Fading Channel block
- Viterbi Decoder and Convolutional Encoder blocks updated with puncturing
- Demodulator blocks enhanced
- M-PSK and Rectangular QAM blocks enhanced
- Phase/Frequency Offset block enhanced
- Additional C data type support
- DVBS2 demo added
- Warning for obsolete blocks

Communications Toolbox

- `convenc` and `vitdec` updated with puncturing and erasing
- `pamdmod`, `pskdmod`, and `qamdmod` functions enhanced

Control System Toolbox

- Siso Design Tool
 - Multiloop compensator design support
 - Automated compensator tuning using PID, IMC, and LQG methods
 - Support for compensator design in Simulink Control Design and Simulink Response Optimization
- LTI Viewer
 - `lsim` and `initial` commands let you change input data and initial conditions directly in the LTI Viewer.
- LTI Objects
 - Full support for descriptor state-space functions
 - New commands `stepinfo` and `lsiminfo` display time-response characteristics
 - Full support for I/O delays in MIMO transfer functions
 - New methods for LTI, FRD, and SS objects

Datafeed Toolbox

- The Thomson Datastream data server added

Distributed Computing Toolbox and MATLAB Distributed Computing Engine

The following features were added in Version 2.0 of the Distributed Computing Toolbox and MATLAB Distributed Computing Engine. Version 2.0 was released via the Web, after R14SP3. No additional new features were added in Version 2.0.1, which is included in R2006a.

- Third-party schedulers supported as an alternative to the MathWorks job manager for running jobs and distributing tasks to the cluster
- Parallel jobs in which the tasks running simultaneously on separate machines are able to communicate with each other during their execution
- Support for adding tasks to running jobs when using the MathWorks job manager
- Unicast support for all communications between distributed computing processes.
- Support for remotely starting and stopping MATLAB Distributed Computing Engine processes.
- Enhanced `nodestatus` command supports monitoring all MATLAB Distributed Computing Engine processes

Embedded Target for Motorola MPC555

- TPU emulation mode to download functions to DPTRAM that are not part of the standard ROM mask of TPU functions

Embedded Target for TI C2000 DSP

- Asynchronous interrupt scheduling support for C280x and C281x
- Expanded support for C2000 targets including:
 - Spectrum Digital F2808 eZdsp(tm) DSP board
 - Custom boards based on the C280x and C281x processor families
- Flash memory programming support

- New C280x Chip Support Library including the following blocks:
 - C280x Hardware Interrupt
 - Idle Task
 - C280x ADC
 - C280x ePWM
 - C280x eQEP
 - C280x eCAN Receive
 - C280x eCAN Transmit
 - From Memory
- New demos for the Spectrum Digital F2812 eZdsp(tm) DSP and F2808 eZdsp(tm) DSP boards

Embedded Target for TI C6000 DSP

- Compatibility with Texas Instruments Code Composer Studio 3.1
- Asynchronous scheduler support added to many blocks
- New host communication blocks in the new Host Communication library, including UDP send/receive blocks and byte manipulation blocks
- New DSP/BIOS blocks in DSP/BIOS Library
- Additional DM642 EVM blocks to support IP/Ethernet communications, including specialized UDP send/receive blocks and IP configuration block
- New Audio ADC and DAC blocks for DM642 EVM
- Additional processors supported by the C6000 Custom Target Preferences block
- Added general blocks to use the new asynchronous scheduler, such as Hardware Interrupt, Task, and CPU Timer blocks
- Two new demos that introduce the asynchronous scheduler in Embedded Target for TI C6000 DSP: “Host to Target Communication” and “Audio and Video Loopback.” In addition, most demos now use the new asynchronous scheduler capability

Filter Design HDL Coder

- Extended control over speed versus area tradeoffs in the realization of single-rate direct form FIR filter designs. To achieve the tradeoff, you can specify for generated HDL filter code either a fully parallel architecture or one of several serial architectures.
- Support for code generation for the Delay filter type

Filter Design Toolbox

- Farrow filters
- Cost method
- Fixed-point report (qreport)
- Wave digital filters in IIR halfband and IIR multirate designs
- Arbitrary Magnitude and Phase filter design
- IIR multirate designs using allpass polyphase components

Financial Toolbox

- The functionality previously available in the Financial Time Series Toolbox has been incorporated into the Financial Toolbox.
- The Financial Time Series frequency conversion functions extensively modified and expanded
- New statistical functions have been added:
 - Multivariate normal regression without missing data
 - Multivariate normal regression with missing data (expectation conditional maximization)
 - Least squares regression with missing data (expectation conditional maximization)
- New financial time series data transformation function

Fixed-Point Toolbox

- Slope Bias math support added
- Scaled double data type support added to `fi` object
- Global `DataTypeOverride` property added to `fipref` object
- Embedded MATLAB support added to more Fixed-Point Toolbox functions
- “round” value added to `fimath` Object `RoundMode` property

GARCH Toolbox

- Augmented Dickey-Fuller and Phillips-Perron unit root tests on univariate financial time series

Image Acquisition Toolbox

- Support for additional Coreco boards
- Support for Coreco Sapera driver. The Image Acquisition Toolbox now supports both the Coreco IFC driver and the Coreco Sapera driver.
- Support for additional Matrox boards
- Support for QImaging devices
- Native Bayer demosaicing — New color space setting interpolates Bayer pattern encoded images into standard RGB images

Image Processing Toolbox

- Enhanced ICC profile capabilities
 - `iccread` and `iccwrite` functions support recent changes to the ICC specification.
 - `iccread`, `makecform`, and `applycform` functions support `DeviceLink`, `ColorSpace`, `Abstract`, and `Grayscale` profiles.
 - `iccread` can read parametric curve types.
- New pointer management functions
 - New `iptPointerManager`, `iptGetPointerBehavior`, and `iptSetPointerBehavior` utility functions let you control what happens when a figure's mouse pointer moves over and then exits an object

- New constraint creation function
 - New `makeConstrainToRectFcn` utility function lets you specify drag constraints for the `indistline`, `inline`, `impoint`, and `imrect` functions. Drag constraints limit how a user can move line, point, and rectangle objects in a figure.

Instrument Control Toolbox

- Support for generic instrument drivers to communicate with arbitrary devices that do not use industry-standard drivers
- Support for LeCroy instrument drivers
- Simulations of the Agilent 33120a function generator and Agilent e3648 DC power supply

Link for Code Composer Studio Development Tools

- Compatible with Texas Instruments Code Composer Studio 3.1.

Link for ModelSim

- Support for a native interface to Verilog models under simulation in ModelSim; VHDL wrappers are no longer required for cosimulation of Verilog code
- Frame-based processing supported in cosimulation
- HDL Cosimulation block supports auto-configuration of ports by sending a query to ModelSim

Mapping Toolbox

- GeoTIFF and PROJ.4 library upgrades
- SDTS++ library upgraded to Version 1.5
- Three new utility functions for NaN-separated polygons and lines:
 - `closePolygonParts` — Close all rings in multipart polygon.
 - `isShapeMultipart` — True if polygon or line has multiple parts.
 - `removeExtraNaNSeparators` — Clean up NaN separators in polygons and lines.

- 32-bit floating-point GeoTIFF
- The coastlines in `coast.MAT` now follow the convention used by `geoshow`, `mapshow`, and `mapview` to display polygons with “holes.” (Outer contours run clockwise, inner contours run counterclockwise.)
- Three new demos:
 - Converting Coastline Data (GSHHS) to Shapefile Format
 - Plotting a 3-D Dome as a Mesh Over a Globe
 - Unprojecting a Digital Elevation Model (DEM)

MATLAB Compiler

- The MATLAB Compiler supports a new target, .NET components. With the optional MATLAB Builder for .NET product, you can create both COM components that can be used in native code applications and .NET components that can be used in managed code applications.

Neural Network Toolbox

- New dynamic learning topologies
 - Time Delay Neural Network
 - Nonlinear Autoregressive Network (NARX)
 - Layer-Recurrent Network
 - Backpropagation support for arbitrarily connected custom dynamic networks, (both forward-propagation and backpropagation, gradients and Jacobians)
- New quick-start wizard
 - NFTOOL - Neural Fitting Tool
- Networks and Data
 - More general transfer, net input, and weight functions
 - Support for unknown weights and unknown/don't care targets
 - Support for dividing data for training, validating, testing
 - New and unified pre/post processing functions

OPC Toolbox

- `opcregister` function enhanced with a `-silent` option to install OPC Foundation Core components without dialog boxes

Real-Time Workshop

- New application programming interface (API) for managing model build information
- New mechanism for customizing post-code generation phase of the build process
- New model configuration option for suppressing makefile generation
- New RSim target option for feeding Inport blocks with MAT-file data
- Switch block optimization for wide control port signals
- Multiport Switch block enhancement that generates a default switch case statement
- Support for Simulink signal object initialization
- Support for Simulink parameters of all numeric data types, including fixed-point, user-defined, and alias data types
- Support for separate value and data type properties for Simulink parameter objects
- Support for new “simplest” rounding mode for fixed-point Simulink blocks

Real-Time Workshop Embedded Coder

- New atomic subsystem option for generating modular functions
- New Export Function capability for generating software components for function-call subsystems
- New Symbol Format options for precise control of auto-generated identifiers
- New Memory Section options for target-specific data placement
- New API for registering module packaging options, including build process hooks and data object customizations
- New Data Object Wizard M-API for using root I/O block names on signals and enforcing signal object resolution
- Enhanced MPT signal object initialization

RF Blockset

- Upper and lower power limits (between which AM/PM conversion applies) added to Amplifier block in the Mathematical Sublibrary.

RF Toolbox

- extract function added
- S-parameter conversion functions added

Signal Processing Blockset

- Integration of filter blocks with Signal Processing Toolbox `fvtool` and filter objects
- Transposed direct-form structure added to FIR Decimation block
- Data type specification modes added to CIC Decimation block
- `taylorwin` window type added to Window Function block
- Reduced simulation memory footprint for fixed-point capable blocks
- Improved usability for To Wave Device block

Signal Processing Toolbox

- Taylor window function (`taylorwin`) added
- SPTool Filter Designer replaced by FDATool
- `sgolay` example improved
- `zeroflag` parameter added to `zp2sos`
- Help for objects changed

SimDriveline

- Run-time parameters now enabled for key blocks
- Improved implementation for clutch locking and unlocking
- Model Reference code generation now compatible with variable step solvers

SimEvents

SimEvents was introduced as a new product via Web, after R14SP3. Version 1.1, which is included in R2006a, includes the following new feature:

- Support for partial replication in the Replicate block

SimMechanics

- New block dialog boxes now compatible with Simulink Model Explorer
- Model Reference code generation now compatible with variable step solvers
- Improved CAD translation
- SolidWorks CAD translator now compatible with SolidWorks versions 2004–2006

SimPowerSystems

- Average value models implemented for two more blocks in Electric Drives library
- Transformer blocks with SI units now available
- Open Circuit option added for the RLC blocks
- Several new demos and enhancements to existing demos

Simulink Control Design

- Design compensators directly in a Simulink model. Simulink Control Design automatically analyzes the model and identifies open and closed loop systems with which to populate an enhanced version of the Control System Toolbox SISO Design Tool.
- Tune multiloop compensators in a single environment that allows the effects of coupling between feedback loops to be immediately understood.
- Tune blocks using both block parameters (for example, PID gains) or zero-pole-gain representations.
- Tune compensators in a control system using all of the graphical and automated design tools associated with the Control System Toolbox SISO Design Tool and Simulink Response Optimization.

Simulink Fixed Point

- New “simplest” rounding mode for more efficient code

Simulink Report Generator

- Support for exporting a Simulink model or a Stateflow diagram as a Web View, a rendition of the model or diagram that you can view and navigate in a Web browser that supports Scalable Vector Graphics (SVG)

Simulink Response Optimization

- Perform response optimization within the SISO Design Tool.
- Add frequency-domain design requirements to SISO Design Tool plots.
- Add time-domain design requirements to LTI Viewer plots (when launched from the SISO Design Tool).
- Use response optimization to design compensators for control systems.
 - Tune parameters within compensator blocks (e.g., poles, zeros, gains).
 - Tune open and closed loop systems.

Spline Toolbox

- Controlled extrapolation beyond the basic interval
- Interpolation of planar tangent-continuous piecewise biarc curve

Stateflow and Stateflow Coder

- Embedded system design
 - Stateflow provides a new chart option that lets you initialize the value of outputs every time a chart wakes up, not only at time 0. The option ensures that outputs are defined in every chart execution and prevents latching of outputs.
- Greater usability
 - Support for customizing any Stateflow Editor menu by adding items and submenus, and by disabling or hiding menu items

Statistics Toolbox

- New parametric methods for copulas including `copulacdf`, `copulapdf`, `copulaparam`, `copularnd`, and `copulastat` functions
- New Markov chain sampler algorithms — Metropolis-Hasting (`mhsample`) and slice sampling (`slicesample`) random number generators
- New random number generators for the Pearson (`pearsrnd`) and Johnson (`johnsrnd`) systems of distributions
- New multivariate normal (`mvncdf`) and t (`mvtcdf`) cumulative distribution functions
- New Bootstrap confidence interval functionality (`bootci`)
- New functionality for jackknife estimation (`jackknife`)
- New Durbin-Watson test functionality (`dwtest`)
- New functionality for finding fractional factorial designs (`fracfactgen`)
- Enhanced anovan support for nested factors and continuous factors
- Enhanced time series data support for `xbarplot`, `schart`, and `ewmaplot` functions
- Robust fitting enhancements for `nlinfit`, `nlparci`, `nlpredci`, and `robustfit` functions
- D-optimal design enhancements for `daugment`, `dcovary`, `rowexch`, `candexch`, `candgen`, `cordexch`, and `x2fx` functions
- New and updated demos

System Identification Toolbox

- `balred` function introduced for model reduction
- Search direction for minimizing criteria can be computed by adaptive Gauss-Newton method
- Maximum number of bisections used by line search increased

Video and Image Processing Blockset

- Autothreshold block enhanced
- Draw Shapes block enhanced
- `isfilterseparable` function added
- Lane Departure Warning System demo added
- MPlay GUI access changed

Virtual Reality Toolbox

- Utilities library added
- VRLM animation playback controls

xPC Target

- Uploading xPC Target signals to Simulink scopes
- Support for uploading real-time data to Simulink scopes when using Simulink external mode
- Support for the MATLAB Compiler: you can generate redistributable, stand-alone xPC Target applications from your M-files with the MATLAB Compiler
- Support added for a number of boards, including the MPL PATI (PowerPC controlled Analog and Timing I/O Intelligence)

New Products

Three new products have been introduced since R14SP3:

MATLAB Builder for .NET	Create MATLAB based .NET and COM components
SimBiology	Model, design, and simulate biochemical pathways
SimEvents	Model and simulate discrete-event systems

MATLAB Builder for .NET

MATLAB Builder for .NET (also called .NET Builder) is an extension to the MATLAB Compiler. Use MATLAB Builder for .NET to

- Package MATLAB functions so that .NET programmers can access them from C#, VB.NET and C++ with managed extensions, or any CLS-compliant language.
- Create components that preserve the flexibility of MATLAB.
.NET Builder provides robust data conversion, indexing, and array formatting capabilities. To support the MATLAB data types, .NET Builder provides the `MWArray` class hierarchy, which is defined in the .NET Builder `MWArray` assembly. You need to reference this assembly in your managed application to convert native arrays to MATLAB arrays and vice versa.
- Handle errors originating from MATLAB functions as standard managed exceptions.
Error descriptions contain specific references to the MATLAB code, thus simplifying the debugging process.
- Create COM components.
COM stands for Component Object Model, which is a software architecture developed by Microsoft to build component-based applications. COM objects expose interfaces that allow applications and other components to access the features of the objects. COM objects are accessible through Visual Basic, C++, or any language that supports COM objects.

Note As of R2006a, the functionality previously available in MATLAB Builder for COM has been incorporated into MATLAB Builder for .NET. Customers current on maintenance for MATLAB Builder for COM will receive a license for MATLAB Builder for .NET.

SimBiology

SimBiology is an integrated environment for modeling biological processes, simulating the dynamic behavior of these processes, and analyzing simulation and experimental data. These features were included in Version 1.0 of SimBiology, which was released in Web-downloadable form after R14SP3. There are no additional new features introduced in R2006a.

- Graphical user interface — SimBiology provides access to the command-line functionality through a graphical user interface (GUI). All the features in SimBiology are also accessible and executable from the MATLAB Command Window.
- Data formats and projects — Organize and save related models, simulation data, and analysis results in project files. Save user-defined kinetic laws and units. Share models by exporting SBML level 2 files.
- Modeling — Create biological models by adding components that include reactions, species, parameters, kinetic laws, rules, and submodels.
- Simulation — Select either deterministic solvers with dimensional analysis, or stochastic solvers with dimensional analysis and unit conversion.
- Analysis — SimBiology is fully integrated with MATLAB. Log data during a simulation and analyze results in MATLAB.

SimEvents

SimEvents extends Simulink with tools for modeling and simulating discrete-event systems using queues and servers.

With SimEvents you can create a discrete-event simulation model in Simulink to simulate the passing of entities through a network of queues, servers, gates, and switches based on events. SimEvents and Simulink provide an integrated environment for modeling hybrid dynamic systems containing continuous-time, discrete-time, and discrete-event components.

MathWorks Products in Transition in R2006a

In R2006a, the following two products have been incorporated into other MathWorks products.

MATLAB Builder for COM

As of R2006a, the functionality previously available in MATLAB Builder for COM has been incorporated into MATLAB Builder for .NET. The MATLAB Builder for .NET documentation contains information previously included in the documentation for MATLAB Builder for COM.

Customers who are current on maintenance for MATLAB Builder for COM will receive a license for MATLAB Builder for .NET.

Financial Time Series Toolbox

As of R2006a, the functionality previously included in the Financial Time Series Toolbox is now included in the Financial Toolbox.

Customers who are current on maintenance for the Financial Time Series Toolbox and the Financial Toolbox will continue to have access to this functionality via the Financial Toolbox.

Upgrading from a Previous Release

These topics summarize the potential compatibility considerations when upgrading from R14SP3 to R2006a:

- “Compatibility Considerations in This Release” on page 30
- “Compatibility Considerations for New Functions and Models” on page 32

Compatibility Considerations in This Release

Upgrading to R2006a is not expected to cause any significant compatibility problems for most users.

To see if any known compatibility considerations exist for your product, see “Release Summary” on page 6. If the table shows that there are compatibility considerations for any of your products, follow the link to the product’s release notes and refer to the table summary of compatibility considerations. From this table you can link to details. If you are upgrading from a release before R14SP3, refer to entries in the table that summarize compatibility considerations with earlier versions.

The compatibility considerations likely to be noticed by the broadest cross-section of users are summarized in the following table:

Product and Area	Compatibility Consideration	Potential Impact	Recommended Actions
MATLAB Programming	str2func now issues a warning for invalid input.	In previous versions, str2func accepted a path to a file instead of a function name. Now it issues a warning, and will produce an error in future versions.	Change the pathname input to str2func to a function name to avoid the warning.
MATLAB External Interfaces	MEX-files and utilities that rely on the GCC compiler might need to be recompiled due to an upgrade to GCC. 3.4.	At the source-code level, GCC 3.4 now complies much more strictly with the C++ standard and rejects certain constructs that were previously accepted, but which are contrary to the language standard.	This means that all objects linked into MATLAB (either statically or dynamically) must be recompiled with GCC 3.4.

Product and Area	Compatibility Consideration	Potential Impact	Recommended Actions (Continued)
Simulink	Certain MEX-files created in versions before Version 6 (R14) will need to be recompiled to work in subsequent versions.	S-functions created prior to R14 that access the sigmap, siglists and sigregions structures might generate segmentation violations.	Recompile S-functions in Version 6 (R14) or a more recent version.
Real-Time Workshop	ComplexSignal, DataTypeIdx, Dimensions, and Width fields removed from records BlockOutputs, Dworks, ExternalInputs, ExternalOutputs, and Parameters in model.rtw	If your application retrieves data from model.rtw without using documented TLC library functions, this application will be incompatible and will produce incorrect results.	Reprogram your application to use the documented TLC library functions to retrieve data from model.rtw.

Compatibility Considerations for New Functions and Models

The introduction of new functions and models could cause a conflict between any of your own M-files, models, and variables having the same names.

Example

If you created a function named `idivide` in a previous release, it might conflict with the new MATLAB `idivide` function (introduced in R2006a). Another conflict that might arise is when you load a MAT-file that contains a variable named `idivide`.

Recommendations

- In the **By Product List** of new functions and models (on the documentation page of the MathWorks Web site), scan the products you have installed for function, model, and variable names you already use, and note potential conflicts.
- Because you could experience conflicts with names from products you do not use, scan the **Alphabetical List** of new functions and models for all products (on the documentation page of the MathWorks Web site), and note any other potential conflicts. For example, you might have a name conflict if you share files with users who have different products installed.
- To identify and address name conflicts, see these topics in the **MATLAB Programming** documentation:
 - **Avoid Using Function Names for Variables**
 - **Naming a Function Uniquely**

System Requirements

Note For more information on system requirements, visit <http://www.mathworks.com/support/sysreq/r2006a/>

All Platforms

- CD-ROM or DVD drive (for installation)
- 512 MB RAM (1024 MB recommended)
- 500 MB disk space

Microsoft® Windows®

32-Bit Operating Systems

- Windows XP (Service Pack 1 or 2); Windows 2000 (Service Pack 3 or 4); or Windows Server 2003
- Intel® Pentium® III processor or later, and AMD processors
- 16-, 24-, or 32-bit OpenGL® capable graphics adaptor

Note R2006a does not support Windows NT®; R14SP2 (Release 14 with Service Pack 2) was the last release to support Windows NT.

64-Bit Operating Systems

- Windows XP x64
- Intel® EM64T or AMD AMD64 processors
- 16-, 24-, or 32-bit OpenGL® capable graphics adaptor

Matrix Size Limitations. Historically, MATLAB matrices have been limited in size to those that would fit in a 32-bit address space. Some of those limitations have been lifted, but some remain. Specifically, the data type used to index into an

`mxArray` is still a 32-bit signed integer. This limits the number of elements in any one array to `INT_MAX - 1`, or 2147483646 (approximately $2 \cdot 10^9$). With this limit, you can create matrices up to 16 GB (for doubles). You can create as many of these as your machine has memory to support.

Mac OS® X

32-Bit Operating Systems

- PowerPC G4 or G5 processor
- Mac OS X 10.4, 10.4.2 (Tiger™), or 10.3.8 or 10.3.9 (Panther™); 10.3.9 requires an Apple® Java patch
- 16-bit graphics or higher adaptor and display (24-bit recommended)
- X11 (X Server)

Linux

32-Bit Operating Systems

- Linux built using Kernel 2.4.x or 2.6.x and `glibc (glibc6) 2.3.2` or later
- Intel® Pentium® III processor or later, and AMD processors
- 16-bit graphics or higher adaptor and display (24-bit recommended)

64-Bit Operating Systems

- Linux built using Kernel 2.4.x or 2.6.x and `glibc (glibc6) 2.3.4` or later
- Intel® EM64T or AMD AMD64 processors
- 16-bit graphics or higher adaptor and display (24-bit recommended)

Matrix Size Limitations. Historically, MATLAB matrices have been limited in size to those that would fit in a 32-bit address space. Some of those limitations have been lifted, but some remain. Specifically, the data type used to index into an `mxArray` is still a 32-bit signed integer. This limits the number of elements in any one array to `INT_MAX - 1`, or 2147483646 (approximately $2 \cdot 10^9$). With this limit, you can create matrices up to 16 GB (for doubles). You can create as many of these as your machine has memory to support.

CDF Writing Not Supported. CDF writing is not supported, but you can read CDF files.

UNIX (Solaris)

Note R2006a does not support HP-UX; R14SP3 (Release 14 with Service Pack 3) was the last release to support HP-UX.

32-Bit Operating Systems

- Sun[®] Solaris[®] 2.10, 2.9, or 2.8 (running R2006a on Solaris 2.8 or 2.9 requires that the operating system have the current recommended patches; see Solution 1-21Q99R)
- SPARC or ultraSPARC
- 16-bit graphics or higher adaptor and display (24-bit recommended)

Sun Solaris Run-Time Libraries. MATLAB dynamically links against the Solaris C++ Run-Time Library, which must reside on the same machine.

To determine whether this library is already installed, type the following line in a UNIX shell. (If you get a No match response, then you need to install it from your Solaris system CD.)

```
ls -l /usr/lib/libCstd.so.*
```

Accessibility Notes

MathWorks products includes a number of modifications to make them more accessible to all users.

For installation instructions relating to accessibility support, see “Installation Notes for Accessibility Support” on page 43.

Products Updated

The MathWorks has made general modifications to make its products more accessible. Particular emphasis has been placed on the accessibility of the following products/features:

- The product installation process
- MATLAB®
- Simulink®
- Control System Toolbox
- Curve Fitting Toolbox
- Excel® Link
- MATLAB® Compiler
- Optimization Toolbox
- Signal Processing Toolbox
- Statistics Toolbox

Summary of Accessibility Support

Accessibility support for blind and visually impaired users includes

- Support for screen readers and screen magnifiers, as described in “Assistive Technologies” on page 39
- Command-line alternatives for most graphical user interface (GUI) options
- Keyboard access to GUI components
- A clear indication of the current cursor focus
- Information available to assistive technologies about user interface elements, including the identity, operation, and state of the element

- Nonreliance on color coding as the sole means of conveying information about working with a GUI
- Noninterference with user-selected contrast and color selections and other individual display attributes, as well as noninterference for other operating system-level accessibility features
- Consistent meaning for bit-mapped images used in GUIs
- HTML documentation that is accessible to screen readers

The MathWorks believes that its products do not rely on auditory cues as the sole means of conveying information about working with a GUI. However, if you do encounter any issues in this regard, please report them to the MathWorks Technical Support group.

http://www.mathworks.com/contact_TS.html

Keyboard access to the user interface includes support for “sticky keys,” which allow you to press key combinations (such as **Ctrl+C**) sequentially rather than simultaneously.

Except for scopes and real-time data acquisition, the MathWorks software does not use flashing or blinking text, objects, or other elements having a flash or blink frequency greater than 2 Hz and lower than 55 Hz.

Accessibility Support Details

These notes about product accessibility cover the following topics:

- “Assistive Technologies” on page 39
- “Troubleshooting” on page 39
- “Documentation” on page 41

If you are using a screen reader, such as JAWS, see also “Installation Notes for Accessibility Support” on page 43.

Assistive Technologies

Note To take advantage of accessibility support features, you must use MathWorks products on a Microsoft Windows platform.

Tested Assistive Technologies

For R2006a, The MathWorks has tested the following assistive technologies:

- JAWS 4.5 and 5.0 (recommended) for Windows (screen reader) from Freedom Scientific
- Built-in accessibility aids from Microsoft, including the Magnifier and “sticky keys”

Use of Other Assistive Technologies

Although The MathWorks has not tested other assistive technologies, such as other screen readers or ZoomText Xtra (screen magnifier) from Ai Squared, The MathWorks believes that most of the accessibility support built into its products should work with most assistive technologies that are generally similar to the ones tested.

If you use other assistive technologies than the ones tested, The MathWorks is very interested in hearing from you about your experiences.

Troubleshooting

This section identifies workarounds for some possible issues you may encounter related to accessibility support in MathWorks products.

JAWS Does Not Detect When the MATLAB Installation Has Started

When you select `setup.exe`, the Windows copying dialog box opens and you are informed. After the files have been copied, the installation splash screen opens, and then the installer starts. However, JAWS does not inform you that the installer has begun: the installer either starts up below other windows or applications or it is minimized. Since the installer is not an active item, nothing is read.

Therefore, check the Windows applications bar for the installer. After you go to the installer, you can use JAWS to perform the installation.

JAWS Stops Speaking

When many desktop components are open, JAWS with MATLAB sometimes stops speaking.

If this happens, close most of the desktop components, exit MATLAB, and restart.

Command Output Not Read

In the MATLAB Command Window, JAWS does not automatically read the results of commands.

To read command output, first select **File > Preferences > Command Window**, select the option **Use arrow keys for navigation instead of command history recall**, and click **OK**. Then, in the Command Window, press the arrow keys to move to the command output and use JAWS keystrokes to read the output.

With this preference set, you cannot use arrow keys to recall previous commands. Instead use the following key bindings:

- Windows key bindings:
 - Previous history: **Ctrl+up arrow**
 - Next history: **Ctrl+down arrow**
- Emacs key bindings:
 - Previous history: **Ctrl+p**
 - Next history: **Ctrl+n**

To return to using the up and down arrow keys to recall previous commands, clear the preference.

Some GUI Menus Are Treated as Check Boxes

For some GUIs (for example, the figure window), menus are treated by JAWS as though they are check boxes, whether or not they actually are.

You can choose a menu item for such GUIs by using accelerator keys (e.g., **Ctrl+N** to select **New Figure**), if one is associated with a menu item. You can also use mnemonics for menu navigation (e.g., **Alt+E**).

Note that check boxes that you encounter by tabbing through the elements of a GUI are handled properly.

Text Ignored in Some GUIs

For some dialog boxes, JAWS reads the dialog box title and any buttons, but ignores any text in the dialog box.

Also, in parts of some GUIs, such as some text-entry fields, JAWS ignores the label of the field. However, JAWS will read any text in the text box.

Documentation

Documentation is available in HTML format for all MathWorks products that are included in R2006a.

Accessing the Documentation

To access the documentation with a screen reader, go to the R2006a documentation area on the MathWorks Web site at

<http://www.mathworks.com/access/helpdesk/help/helpdesk.html>

Navigating the Documentation

Note that the first page that opens lists the products. To get the documentation for a specific product, click the link for that product.

The table of contents is in a separate frame. You can use a document's table of contents to navigate through the sections of that document.

Because you will be using a general Web browser, you will not be able to use the search feature included in the MATLAB Help browser. You will have access to an index for the specific document you are using. The cross-product index of the MATLAB Help browser is not available when you are using a general Web browser.

Products

The documentation for all products is in HTML and can be read with a screen reader. However, for most products, most equations and most graphics are not accessible.

The following product documentation has been modified (as described below) to enhance its accessibility for people using a screen reader such as JAWS:

- MATLAB (many sections, but not the function reference pages (however, M-file help is accessible))
- Excel Link
- Optimization Toolbox
- Signal Processing Toolbox
- Statistics Toolbox

Documentation Modifications

Modifications to the documentation include the following:

- Describing illustrations in text (either directly or via links)
- Providing text to describe the content of tables (as necessary)
- Restructuring information in tables to be easily understood when a screen reader is used
- Providing text links in addition to any image mapped links

Equations

Equations that are integrated in paragraphs are generally explained in words. However, most complex equations that are represented as graphics are not currently explained with alternative text.

Installation Notes for Accessibility Support

Note If you are not using a screen reader such as JAWS, you can skip this section.

This section describes the installation process for setting up your MATLAB environment to work effectively with JAWS.

Use the regular MATLAB installation script to install the products for which you are licensed. The installation script has been modified to improve its accessibility for all users.

Note Java Access Bridge 1.1 is installed automatically when you install MATLAB.

After you complete the product installation, there are some additional steps you need to perform to ensure JAWS works effectively with MathWorks products.

Setting Up JAWS

Make sure that JAWS is installed on your machine. If it is, there is probably a shortcut to it on the Windows desktop.

Setting up JAWS involves these tasks:

- 1 Add the Access Bridge to your Windows path (for networked installations only).
- 2 Create the `accessibility.properties` file.

These tasks are described in more detail below.

(For Networked Installations Only) Add Access Bridge to Your Path. If you are running MATLAB in a networked installation environment (that is, if the MATLAB Installer was not run on your machine), you need to take the following steps to add Access Bridge to your Windows path.

Note This procedure assumes your Windows **Start** button is set to Classic mode. To set Classic mode, from the **Start** button, select **Settings**. Next select **Task Bar and Menu**. Then select the **Start Menu** tab and make sure the Classic Start Menu option is enabled. Click **OK** and you are done.

- 1 From the **Start** button, select **Settings**, next select **Control Panel**. Scroll down and click the **System** icon to display the System Properties dialog box.
- 2 In the System Properties dialog box, select the **Advanced** tab.
- 3 Click **Environment Variables**.
- 4 Under **System variables**, select the Path option.
- 5 Click the **Edit** button.
- 6 To the start of the Path environment variable, add the directory that contains `matlab.exe`; for example:

C:\matlab71\bin\win32;

Be sure to include that semicolon between the end of this directory name and the text that was already there.
- 7 Click **OK** three times.
- 8 If JAWS is already running, exit and restart.

Note JAWS must be started with these path changes in effect to work properly with MATLAB.

Create the `accessibility.properties` File.

- 1 Create a text file that contains the following two lines:

screen_magnifier_present=true
assistive_technologies=com.sun.java.accessibility.AccessBridge

2 Use the filename `accessibility.properties`.

3 Move the `accessibility.properties` file into

```
$(matlabroot)\sys\java\jre\win32\jre1.5.0\lib\
```

JAWS Pronunciation Dictionary. As a convenience, The MathWorks provides a pronunciation dictionary for JAWS. This dictionary is in a file called `MATLAB.jdf`.

During installation, the file is copied to your system under the MATLAB root directory at `sys\Jaws\matlab.jdf`.

To use the dictionary, you must copy it to the `\SETTINGS\ENU` folder located beneath the JAWS root installation directory.

You need to restart JAWS and MATLAB for the settings to take effect.

Testing

After you install JAWS and set up your environment as described above, you should test to ensure JAWS is working properly:

1 Start JAWS.

2 Start MATLAB.

JAWS should start talking to you as you select menu items and work with the MATLAB user interface in other ways.

Quick Access to Product-Specific Release Notes

This section provides online links to the product-specific release notes for each MathWorks product. If a product is *not* listed below, it has not changed significantly since Release 11.

MATLAB and Simulink are listed first in the table. The rest of the MathWorks products are listed alphabetically.

Note Products followed by an asterisk (*) are updated since R14SP3. Refer to release notes for products that do *not* have an asterisk only if you are upgrading from a release before R14SP3.

- MATLAB®*
- Simulink®*
- Aerospace Blockset*
- Bioinformatics Toolbox*
- CDMA Reference Blockset
- Communications Blockset*
- Communications Toolbox*
- Control System Toolbox*
- Curve Fitting Toolbox
- Data Acquisition Toolbox*
- Database Toolbox
- Datafeed Toolbox*
- Distributed Computing Toolbox*
- Embedded Target for Infineon® C166 Microcontrollers
- Embedded Target for Motorola® HC12
- Embedded Target for Motorola® MPC555*
- Embedded Target for OSEK/VDX®
- Embedded Target for TI C2000™ DSP*
- Embedded Target for TI C6000™ DSP*

- [Excel Link](#)
- [Filter Design HDL Coder*](#)
- [Filter Design Toolbox*](#)
- [Financial Derivatives Toolbox](#)
- [Financial Time Series Toolbox \(now part of the Financial Toolbox\)](#)
- [Financial Toolbox*](#)
- [Fixed-Income Toolbox](#)
- [Fixed-Point Toolbox*](#)
- [Fuzzy Logic Toolbox \(no release notes\)](#)
- [GARCH Toolbox*](#)
- [Gauges Blockset*](#)
- [Genetic Algorithm and Direct Search Toolbox](#)
- [Image Acquisition Toolbox*](#)
- [Image Processing Toolbox*](#)
- [Instrument Control Toolbox*](#)
- [Link for Code Composer Studio™ Development Tools*](#)
- [Link for ModelSim®*](#)
- [Mapping Toolbox*](#)
- [MATLAB® Builder for COM \(now part of MATLAB® Builder for .NET\)](#)
- [MATLAB® Builder for Excel](#)
- [MATLAB® Builder for .NET \(new product\)](#)
- [MATLAB® Compiler*](#)
- [MATLAB® Distributed Computing Engine*](#)
- [MATLAB® Report Generator](#)
- [MATLAB® Web Server](#)
- [Model-Based Calibration Toolbox*](#)
- [Model Predictive Control Toolbox](#)
- [Neural Network Toolbox*](#)
- [OPC Toolbox*](#)
- [Optimization Toolbox*](#)
- [Partial Differential Equation Toolbox* \(no release notes\)](#)

- Real-Time Windows Target
- Real-Time Workshop®*
- Real-Time Workshop® Embedded Coder*
- RF Blockset*
- RF Toolbox*
- Robust Control Toolbox*
- Signal Processing Blockset*
- Signal Processing Toolbox*
- SimBiology (new product)
- SimDriveline*
- SimEvents*
- SimMechanics*
- SimPowerSystems*
- Simulink® Accelerator*
- Simulink® Control Design*
- Simulink® Fixed Point*
- Simulink® Parameter Estimation
- Simulink® Report Generator*
- Simulink® Response Optimization*
- Simulink® Verification and Validation*
- Spline Toolbox*
- Stateflow® and Stateflow® Coder*
- Statistics Toolbox*
- Symbolic Math Toolbox
- System Identification Toolbox*
- Video and Image Processing Blockset*
- Virtual Reality Toolbox*
- Wavelet Toolbox
- xPC Target*
- xPC TargetBox®

MATLAB Software Acknowledgments

MATLAB and/or its associated products include software developed by the following third parties.

ARnoldi PACKage (ARPACK)

Rich Lehoucq, Kristi Maschhoff, Danny Sorensen, and Chao Yang
<http://www.caam.rice.edu/software/ARPACK>

Assertion blocks were developed in cooperation with

Helmut Keller, Andreas Rau, and Joachim Boensch, members of the Control System Design (CSD) group at DaimlerChrysler Germany.

Automatically Tuned Linear Algebra Software (ATLAS)

R. Clint Whaley and Jack Dongarra
<http://www.netlib.org/atlas>

The Image Acquisition Toolbox DCAM adaptor uses the Carnegie Mellon University driver to communicate with cameras compatible with the IIDC 1394-based Digital Camera Specification (DCAM).

Copyright© 2003 Carnegie Mellon University All Rights Reserved
The Robotics Institute - Carnegie Mellon University
<http://www-2.cs.cmu.edu/~iwan/1394/>

FDLIBM C math library for machines that support IEEE 754 floating point

Developed at SunSoft, a Sun Microsystems, Inc. business, by Kwok C. Ng and others. FDLIBM is freely redistributable and is available through NetLib. For information about FDLIBM, see <http://www.netlib.org>.

fft and related MATLAB functions are based on the FFTW library.

Developed by Matteo Frigo and Steven G. Johnson
Copyright © 2003 Matteo Frigo
Copyright © 2003 Massachusetts Institute of Technology. All rights reserved.
Used under terms of a commercial license
<http://www.fftw.org>

FreeType2 Project library is included with Simulink.

FreeType was created by David Turner, Robert Wilhelm, and Werner Lemberg <http://freetype.org>

A few MathWorks products contain the graphviz code from AT&T. ("AT&T Software") proprietary to AT&T Corp. ("AT&T").

The AT&T Software is provided to you "AS IS". YOU ASSUME TOTAL RESPONSIBILITY AND RISK FOR USE OF THE AT&T SOFTWARE. AT&T DOES NOT MAKE, AND EXPRESSLY DISCLAIMS, ANY EXPRESS OR IMPLIED WARRANTIES OF ANY KIND WHATSOEVER, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, WARRANTIES OF TITLE OR NON-INFRINGEMENT OF ANY INTELLECTUAL PROPERTY RIGHTS, ANY WARRANTIES ARISING BY USAGE OF TRADE, COURSE OF DEALING OR COURSE OF PERFORMANCE, OR ANY WARRANTY THAT THE AT&T SOFTWARE IS "ERROR FREE" OR WILL MEET YOUR REQUIREMENTS

Unless you accept a license to use the AT&T Software, you shall not reverse compile, disassemble or otherwise reverse engineer this product to ascertain the source code for any AT&T Software.

© AT&T Corp. All rights reserved. AT&T is a registered trademark of AT&T Corp.

HDF capability in the functions `imread`, `imwrite`, `imfinfo`, and `hdf` and HDF 5 capability in the functions `hdf5info` and `hdf5read` are based on code of which portions were developed at

The National Center for Supercomputing Applications at the University of Illinois at Urbana-Champaign.

J2PrinterWorks .class files are a product of Wildcrest Associates.

JPEG capability in the functions `imread`, `imwrite`, `imfinfo`, `print`, and `saveas`:

This software is based in part on the work of the Independent JPEG Group.

Linear Algebra PACKage (LAPACK)

<http://www.netlib.org/lapack> (for general information about LAPACK)

For details, see the *LAPACK User's Guide*.

E. Anderson, Z. Bai, C. Bischof, L. S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen

For a printed version of the *LAPACK User's Guide*, go to <http://www.siam.org>.

For an online version of the *LAPACK User's Guide*, go to

http://www.netlib.org/lapack/lug/lapack_lug.html.

ScaLAPACK. Copyright 1997 by the Society for Industrial and Applied Mathematics. Authors: L.S. Blackford, J. Choi, A. Cleary, E.D'Azevedo, J. Demmel, I. Dhillon, J. Dongarra, S. Hammarling, G. Henry, A. Petitet, K. Stanley, D. Walker, R.C. Whaley.

Acknowledgements: This material is based upon work supported by the National Science Foundation under Grant No. ASC-9313958 and DOE Grant No. DE-FG03-94ER25219. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation (NSF) or the Department of Energy (DOE). We give credit to and thank all of the LAPACK authors for allowing the reuse of large portions of the LAPACK Users' Guide. The printed version of the ScaLAPACK Users' Guide is available from SIAM.

openVRML, developed by The OpenVRML project (www.openvrml.org), is used in the Virtual Reality Toolbox. openVRML is redistributed herein under The GNU Lesser General Public License (LGPL), Version 2.1.

Qhull based computational geometry capability in MATLAB

Qhull copyright (c) 1993-2003 The National Science and Technology Research Center for Computation and Visualization of Geometric Structures, The Geometry Center, University of Minnesota

e-mail: qhull@qhull.org

For complete copyright information, issue the MATLAB command `help qhull`.

Sparse matrix minimum degree permutation functions `colamd` and `symamd`

Copyright © 1998-2000 by the University of Florida. All rights reserved.

Authors of the code are Stefan I. Larimore and Timothy A. Davis (davis@cise.ufl.edu), University of Florida. The algorithm was developed in collaboration with John Gilbert, Xerox PARC, and Esmond Ng, Oak Ridge National Laboratory.

This work was supported by the National Science Foundation, under grants DMS-9504974 and DMS-9803599.

For complete copyright information, issue the MATLAB command `edit colamd` or `edit symamd`.

The SLICOT library of numerical algorithms for computations in systems and control theory is used in the Control System Toolbox. The SLICOT library is developed by the NICONET group (www.win.tue.nl/niconet/NIC2/slicot.html).

More detailed information on SLICOT can be found in:

Benner, P., Mehrmann, V., Sima, V., Van Huffel, S., and A. Varga: "SLICOT - A Subroutine Library in Systems and Control Theory", June 1997, NICONET Report 97-3.

SLICOT is freely available through WWW: (<http://www.win.tue.nl/wgs/>) or anonymous ftp: (<ftp://wgs.esat.kuleuven.ac.be/pub/WGS/SLICOT/>).

The MATLAB implementation of TeX is compiled from Donald Knuth's original TeX parser (Version: 3.14159) located on the TeX Archive Network: www.ctan.org. The LaTeX distribution was also obtained from www.ctan.org.

TIFF capability in the functions `imread`, `imwrite`, `imfinfo`, `print`, and `saveas`:

Copyright © 1988-1999 Sam Leffler

Copyright © 1991-1999 Silicon Graphics, Inc.

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that (i) the above copyright notices and this permission notice appear in all copies of the software and related documentation, and (ii) the names of Sam Leffler and Silicon Graphics may not be used in any advertising or publicity relating to the software without the specific, prior written permission of Sam Leffler and Silicon Graphics.

THE SOFTWARE IS PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EXPRESS, IMPLIED OR OTHERWISE, INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL SAM LEFFLER OR SILICON GRAPHICS BE LIABLE FOR ANY SPECIAL, INCIDENTAL, INDIRECT OR CONSEQUENTIAL DAMAGES OF ANY KIND, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER OR NOT ADVISED OF THE POSSIBILITY OF DAMAGE, AND ON ANY THEORY OF LIABILITY, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Unsymmetric MultiFrontal PACKage (UMFPACK) for solving unsymmetric sparse linear systems.

UMFPACK Version 4.0, April 11, 2002. Copyright © 2002 by Timothy A. Davis, University of Florida, davis@cise.ufl.edu. All Rights Reserved.

See <http://www.cise.ufl.edu/research/sparse/umfpack> for general information about UMFPACK. For details, the *UMFPACK Version 4.0 User Guide* is available at <http://www.cise.ufl.edu/research/sparse/umfpack/v4.0/UserGuide.pdf>.

Unsymmetric MultiFrontal PACKage (UMFPACK) for solving unsymmetric sparse linear systems.

UMFPACK Version 4.0, April 11, 2002. Copyright © 2002 by Timothy A. Davis, University of Florida, davis@cise.ufl.edu. All Rights Reserved.

wxWidgets cross-platform native UI framework, developed by the wxWidgets project (www.wxwidgets.org), is used in the Virtual Reality Toolbox. wxWidgets is redistributed herein under The wxWidgets Library License, Version 3.0. The wxWidgets Library License does not apply to any other part of MATLAB, Simulink, or other products or services distributed by the The MathWorks, Inc.

MATLAB includes an XML parser developed by the Apache Software Foundation (www.apache.org) redistributed herein under The Apache Software License, Version 1.1. The Apache Software License does not apply to any other part of MATLAB, Simulink, or other products or services distributed by The MathWorks, Inc.

Revision: 1.1.2.1

Copyright Notice and Statement for NCSA Hierarchical Data Format (HDF)
Software Library and Utilities

NCSA HDF5 (Hierarchical Data Format 5) Software Library and Utilities
Copyright 1998, 1999, 2000, 2001, 2002 by the Board of Trustees of the University of Illinois.
All rights reserved.

Contributors: National Center for Supercomputing Applications (NCSA) at the University of Illinois at Urbana-Champaign (UIUC), Lawrence Livermore National Laboratory (LLNL), Sandia National Laboratories (SNL), Los Alamos National Laboratory (LANL), Jean-loup Gailly and Mark Adler (gzip library).
Redistribution and use in source and binary forms, with or without modification, are permitted for any purpose (including commercial purposes) provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions, and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions, and the following disclaimer in the documentation and/or materials provided with the distribution.
3. In addition, redistributions of modified forms of the source or binary code must carry prominent notices stating that the original code was changed and the date of the change.

4. All publications or advertising materials mentioning features or use of this software are asked, but not required, to acknowledge that it was developed by the National Center for Supercomputing Applications at the University of Illinois at Urbana-Champaign and to credit the contributors.

5. Neither the name of the University nor the names of the Contributors may be used to endorse or promote products derived from this software without specific prior written permission from the University or the Contributors, as appropriate for the name(s) to be used.

6. THIS SOFTWARE IS PROVIDED BY THE UNIVERSITY AND THE CONTRIBUTORS "AS IS" WITH NO WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED. In no event shall the University or the Contributors be liable for any damages suffered by the users arising out of the use of this software, even if advised of the possibility of such damage.

Portions of HDF5 were developed with support from the University of California, Lawrence Livermore National Laboratory (UC LLNL).

The following statement applies to those portions of the product and must be retained in any redistribution of source code, binaries, documentation, and/or accompanying materials:

This work was partially produced at the University of California, Lawrence Livermore National Laboratory (UC LLNL) under contract no. W-7405-ENG-48 (Contract 48) between the U.S. Department of Energy (DOE) and The Regents of the University of California (University) for the operation of UC LLNL.

DISCLAIMER:

This work was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately-owned rights. Reference herein to any specific commercial products, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

ICU License - ICU 1.8.1 and later COPYRIGHT AND PERMISSION NOTICE

Copyright (c) 1995-2003 International Business Machines Corporation and others All rights reserved. Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, provided that the above copyright notice(s) and this permission notice appear in all copies of the Software and that both the above copyright notice(s) and this permission notice appear in supporting documentation.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder.

All trademarks and registered trademarks mentioned herein are the property of their respective owners.

The Distributed Computing Toolbox and MATLAB Distributed Computing Engine include software developed by the following third parties.

The Java Service Wrapper by Tanuki Software.

Copyright (c) 1999, 2004 Tanuki Software

Permission is hereby granted, free of charge, to any person obtaining a copy of the Java Service Wrapper and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sub-license, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Portions of the Software have been derived from source code developed by Silver Egg Technology under the following license:

Copyright (c) 2001 Silver Egg Technology

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sub-license, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

MATLAB® & SIMULINK®

Product Families

R2006b

- Computation
- Visualization
- Programming
- Modeling
- Simulation
- Implementation

Release Notes for
Release 2006b



How to Contact The MathWorks:



www.mathworks.com
comp.soft-sys.matlab
www.mathworks.com/contact_TS.html

Web
Newsgroup
Technical support



suggest@mathworks.com
bugs@mathworks.com
doc@mathworks.com
service@mathworks.com
info@mathworks.com

Product enhancement suggestions
Bug reports
Documentation error reports
Order status, license renewals, passcodes
Sales, pricing, and general information



508-647-7000 (Phone)



508-647-7001 (Fax)



The MathWorks, Inc.
3 Apple Hill Drive
Natick, MA 01760-2098

For contact information about worldwide offices, see the MathWorks Web site.

Release Notes for Release 2006b

© COPYRIGHT 2006 by The MathWorks, Inc.

The software described in this document is furnished under a license agreement. The software may be used or copied only under the terms of the license agreement. No part of this manual may be photocopied or reproduced in any form without prior written consent from The MathWorks, Inc.

FEDERAL ACQUISITION: This provision applies to all acquisitions of the Program and Documentation by, for, or through the federal government of the United States. By accepting delivery of the Program or Documentation, the government hereby agrees that this software or documentation qualifies as commercial computer software or commercial computer software documentation as such terms are used or defined in FAR 12.212, DFARS Part 227.72, and DFARS 252.227-7014. Accordingly, the terms and conditions of this Agreement and only those rights specified in this Agreement, shall pertain to and govern the use, modification, reproduction, release, performance, display, and disclosure of the Program and Documentation by the federal government (or other entity acquiring for or through the federal government) and shall supersede any conflicting contractual terms or conditions. If this License fails to meet the government's needs or is inconsistent in any respect with federal procurement law, the government agrees to return the Program and Documentation, unused, to The MathWorks, Inc.

Trademarks

MATLAB, Simulink, Stateflow, Handle Graphics, Real-Time Workshop, and xPC TargetBox are registered trademarks and SimBiology, SimEvents, and SimHydraulics are trademarks of The MathWorks, Inc. Other product or brand names are trademarks or registered trademarks of their respective holders.

Patents

The MathWorks products are protected by one or more U.S. patents. Please see www.mathworks.com/patents for more information.

Revision History

September 2006 First printing New for Release 2006b

Release Notes for R2006b

Highlights of R2006b	2
Summary of New Features	3
Products with New Features	3
MATLAB®	5
Simulink®	6
Aerospace Blockset	7
Bioinformatics Toolbox	7
Communications Blockset	8
Communications Toolbox	8
Control System Toolbox	9
Data Acquisition Toolbox	9
Database Toolbox	9
Datafeed Toolbox	10
Distributed Computing Toolbox	10
Embedded Target for Infineon® C166	10
Embedded Target for TI C2000™ DSP	10
Embedded Target for TI C6000™ DSP	11
Excel Link	11
Filter Design HDL Coder	12
Filter Design Toolbox	12
Financial Toolbox	13
Fixed Point Toolbox	13
GARCH Toolbox	13
Image Acquisition Toolbox	14
Image Processing Toolbox	14
Link for Code Composer Studio™	14
Link for ModelSim®	15
Mapping Toolbox	15
MATLAB® Builder for .NET	15
MATLAB® Builder for Excel	16
MATLAB® Compiler	16
MATLAB® Distributed Computing Engine	16

Model-Based Calibration Toolbox	16
Optimization Toolbox	17
Real-Time Windows Target	17
Real-Time Workshop®	18
Real-Time Workshop® Embedded Coder	18
RF Toolbox	18
Robust Control Toolbox	19
Signal Processing Blockset	19
Signal Processing Toolbox	20
SimBiology™	20
SimEvents™	20
SimHydraulics™	21
SimMechanics	21
SimPowerSystems	21
Simulink® Fixed Point	21
Simulink® Report Generator	21
Simulink® Response Optimization	22
Simulink® Verification and Validation	22
Spline Toolbox	22
Stateflow®	22
Stateflow® Coder	23
Statistics Toolbox	23
System Identification Toolbox	23
Video and Image Processing Blockset	23
Virtual Reality Toolbox	24
Wavelet Toolbox	24
xPC Target	24
New Products	26
Aerospace Toolbox	26
Link for TASKING®	27
MATLAB® Builder for Java™	28
SimHydraulics™	28
Simulink® HDL Coder	29
SystemTest	30
R2006b Products with License-Related Changes	31
Aerospace Blockset	31
Embedded Target for Infineon C166® Microcontrollers	31
Embedded Target for OSEK/VDX®	31

Discontinued Products	32
CDMA Reference Blockset	32
MATLAB® Web Server	32
Compatibility Considerations	33
R2006b Compatibility Assessment	33
Compatibility for Each Product	35
Compatibility Considerations for New Functions and Models ..	35
System Requirements	36
No Major Supported Platform Changes	36
Matrix Size Limitations	36
For Linux: CDF Writing Not Supported	36
Sun Solaris Run-Time Libraries	36
Release Summary	37

Release Notes for R2006b

Highlights of R2006b

Release 2006b (R2006b) includes updates to MATLAB® and Simulink®, six new products released since R2006a, and updates and bug fixes for 80 products.

New capabilities for the MATLAB product family include

- Additional support for large data set handling in MATLAB, including improved memory efficiency and performance when processing large arrays
- MATLAB application deployment to Java™
- Distributed computing tools supporting Windows® Compute Cluster Server 2003, distributed arrays, and parallel math functions
- Graphical interface for defining and solving optimization problems

New capabilities for the Simulink product family include

- Verifying compliance with modeling standards
- Managing tests and analyzing results for system verification and validation
- HDL code generation for Simulink and Stateflow®
- Easier incorporation of legacy C/C++ code for simulation and code generation
- Embedded MATLAB function block support for MATLAB structures and Simulink buses
- Modeling Mealy and Moore semantics to support C and HDL code generation
- Stateflow support for Simulink buses

Six new products have been introduced since R2006a

- Aerospace Toolbox
- Link for TASKING®
- MATLAB® Builder for Java™
- SimHydraulics™
- Simulink® HDL Coder
- SystemTest

Summary of New Features

Note For information about potential issues involved in upgrading from R2006a, see “Compatibility Considerations” on page 33.

Products with New Features

This section summarizes the major new features and enhancements introduced in R2006b for the following products:

“MATLAB®” on page 5	“Embedded Target for Infineon® C166” on page 10
“Simulink®” on page 6	“Embedded Target for TI C2000™ DSP” on page 10
“Aerospace Blockset” on page 7	“Embedded Target for TI C6000™ DSP” on page 11
“Bioinformatics Toolbox” on page 7	“Excel Link” on page 11
“Communications Blockset” on page 8	“Filter Design HDL Coder” on page 12
“Communications Toolbox” on page 8	“Filter Design Toolbox” on page 12
“Control System Toolbox” on page 9	“Financial Toolbox” on page 13
“Data Acquisition Toolbox” on page 9	“Fixed Point Toolbox” on page 13
“Database Toolbox” on page 9	“GARCH Toolbox” on page 13
“Datafeed Toolbox” on page 10	“Image Acquisition Toolbox” on page 14
“Distributed Computing Toolbox” on page 10	“Image Processing Toolbox” on page 14

“Link for Code Composer Studio™” on page 14	“Signal Processing Toolbox” on page 20
“Link for ModelSim®” on page 15	“SimBiology™” on page 20
“Mapping Toolbox” on page 15	“SimEvents™” on page 20
“MATLAB® Builder for .NET” on page 15	“SimHydraulics™” on page 21
“MATLAB® Builder for Excel” on page 16	“SimMechanics” on page 21
“MATLAB® Compiler” on page 16	“SimPowerSystems” on page 21
“MATLAB® Distributed Computing Engine” on page 16	“Simulink® Fixed Point” on page 21
“Model-Based Calibration Toolbox” on page 16	“Simulink® Report Generator” on page 21
“Optimization Toolbox” on page 17	“Simulink® Response Optimization” on page 22
“Real-Time Windows Target” on page 17	“Simulink® Verification and Validation” on page 22
“Real-Time Workshop®” on page 18	“Spline Toolbox” on page 22
“Real-Time Workshop® Embedded Coder” on page 18	“Stateflow®” on page 22
“RF Toolbox” on page 18	“Stateflow® Coder” on page 23
“Robust Control Toolbox” on page 19	“Statistics Toolbox” on page 23
“Signal Processing Blockset” on page 19	“System Identification Toolbox” on page 23

“Video and Image Processing Blockset” on page 23

“Wavelet Toolbox” on page 24

“Virtual Reality Toolbox” on page 24

“xPC Target” on page 24

MATLAB®

Development Environment

- M-Lint code checker improvements, including ability to disable classes of warnings, support in Simulink Embedded MATLAB Editor, support for MATLAB Compiler messages, and display of messages in Japanese
- New file comparison tool for easier assessment of differences between any two M-files
- Enhancements to debugging functions, including an option to save breakpoints that can be restored in a future session
- Improved inspection of workspace variables with new statistical columns in the Workspace browser and the ability to view time series objects in the Array Editor
- M-code generation from Import Wizard, for programmatic access to interactively imported files
- New look and feel on Linux and Solaris platforms
- Help browser support for exact-phrase and wildcard searches

Mathematics

- New `ldl` function for full factorization and solving for Hermitian matrices
- New permutation vector option for `lu`, `luinc`, and `ldl` functions
- Upper and lower triangular factor options for `chol` and `ldl` for improved performance and memory efficiency

Graphics

- Interactive page layout and print preview allowing customization of plots and page position before printing
- Support for mouse scroll wheel in figure Zoom Tool

File I/O

- New functions giving access to more than 200 routines in the HDF5 C API library, supporting reading and writing of complex data types, HDF5 data subsetting capabilities, and other HDF5 features.

Large Data Set Handling and Performance

- Elimination of temporary memory copies when calling M-file functions and some built-in functions in-place (i.e., `x = myfunction(x)`) for easier handling of large data sets and improve performance
- Ability to read and write MAT-files greater than 2 GB on 64-bit platforms via new switches to the `save` function
- Ability to view larger arrays in Array Editor
- Full JIT/Accelerator support for 64-bit platforms, allowing faster code execution

For details, see the product-specific release notes.

Simulink®

Large-Scale Modeling

- Model Dependency Viewer enhancements for finding a model's dependencies on block libraries and referenced models
- Support for running function-call models without them being referenced

New and Enhanced Blocks

- Lookup table blocks support for fixed-point arithmetic, consistency checking, more efficient code generation, and other enhancements
- Use of same internal routines that MATLAB calls for trigonometric, exponential, and rounding and remainder operations
- Signal Builder block support for printing signal groups

Embedded Software Design and Implementation

- Legacy Code Tool for faster creation of S-functions from legacy C/C++ code

MATLAB Language Support

- Support for structures as inputs, outputs, and persistent variables in Embedded MATLAB Function blocks
- Support for reading and writing Simulink bus signals at inputs and outputs of Embedded MATLAB Function blocks
- Expansion of the Embedded MATLAB runtime library with 36 new functions added for code generation, including `std`, `cov`, `rand`, and `histc`

Usability Enhancements

- New API commands for programmatic access to configuration set, data logging, and Simulink® Accelerator features
- Unified Simulink Fonts Preferences dialog box for simpler selection of fonts for blocks, lines, and annotations

For details, see the product-specific release notes.

Aerospace Blockset

- Aerodynamic Forces and Moments block includes stability and wind axes
- New Digital DATCOM Forces and Moments block for computing aerodynamic forces and moments using Digital DATCOM static and dynamic stability derivatives
- FlightGear 0.9.10 supported by Aerospace Blockset flight simulator interface
- Pack `net_fdm` Packet for FlightGear block generates code for all targets, including xPC Target

For details, see the product-specific release notes.

Bioinformatics Toolbox

Version 2.3 of the Bioinformatics Toolbox, released via the Web after Release 2006a, includes these features:

- Ability to draw linear or circular map of features from a GenBank structure
- Gene expression values calculation from Affymetrix probe-level data using robust multi-array average (RMA)
- Background adjustment on Affymetrix probe-level data using RMA

- Rank-invariant set normalization
- Volcano plot of microarray data
- Two-sample, two-tailed t-test to evaluate differential expression of genes from two experimental conditions or phenotypes

Version 2.4, part of Release 2006b, includes these features:

- New functions to parse annotated sequence data and read mzXML files
- New functions to apply basic graph theory algorithms to sparse matrices and biograph objects
- Updated functions to read and access data
- Updated functions to analyze and visualize microarray data
- Updated functions to align and visualize sequences

For details, see the product-specific release notes.

Communications Blockset

- Fixed-point capability added to the Viterbi Decoder, PAM Demodulator, PAM Modulator, PSK modulator, and Square QAM Modulator blocks
- Bitwise soft-decision outputs enabled for PSK and Rectangular QAM demodulator blocks
- New fixed-point MSK demo
- Faster BCH Encoder and Decoder blocks operate on shortened BCH codes

For details, see the product-specific release notes.

Communications Toolbox

- Bitwise soft-decision outputs enabled for PSK and QAM demodulators
- New LDPC encoder and decoder functions
- Line of sight Doppler Shift property added to the Rician channel
- Theoretical BER results for Hamming, Golay, and Reed Solomon codes available from the command line

For details, see the product-specific release notes.

Control System Toolbox

- SISO Design Tool supports two new feedback structures:
 - Internal model control structure
 - Multiloop cascade configuration

For details, see the product-specific release notes.

Data Acquisition Toolbox

- New Data Acquisition block library to acquire analog or digital data in Simulink and to output analog or digital data from a model to a hardware device
- Data acquisition hardware support for additional National Instruments (NI) data acquisition devices: NI USB-6251 (USB), NI USB-6259 (USB), NI PCIe-6251 (PCI Express), and NI PCIe-6259 (PCI Express)
- Data acquisition hardware support for an additional Measurement Computing Corporation (MCC) data acquisition device: MCC USB-1408FS (USB)

For details, see the product-specific release notes.

Database Toolbox

- Improved `fetch` function executes specified query and imports results into MATLAB, combining capabilities of the previous `exec` and `fetch` functions
- New `fetchmulti` function imports data into MATLAB from multiple resultsets
- New `runstoredprocedure` function executes a stored procedure using input parameters specified in a cell array
- Support for specifying catalog and schema for a data source using the Visual Query Builder

For details, see the product-specific release notes.

Datafeed Toolbox

- Support for Federal Reserve Economic Data (FRED)
- Fields added for Yahoo support

For details, see the product-specific release notes.

Distributed Computing Toolbox

- Support for MPI (parallel jobs) with third-party scheduler or MathWorks job manager
- Enhanced error detection for MPI (parallel jobs)
- Support for Windows Compute Cluster Server (CCS)
- Support for Win64
- Interactive Parallel Mode (pmode)
- New distributed array structure to support parallel algorithms

For details, see the product-specific release notes.

Embedded Target for Infineon® C166

- Compatibility with Link for TASKING®

For details, see the product-specific release notes.

Embedded Target for TI C2000™ DSP

- New host-side SCI blocks for communication with running target processor and for real-time synchronization and verification
- New and enhanced target block support for
 - C280x SCI and SPI device driver blocks
 - C281x SCI custom data configuration and SPI enhanced blocking mode and output error status support
 - SCI and SPI module configuration for Target Preferences blocks
- Support for TMS320C2802 and TMS320C2809 digital signal controllers

- New demos for Spectrum Digital F2812, F2808 eZdsp™ or F2806 custom boards:
 - Hardware-in-the-loop (HIL) demo using SCI
 - Permanent Magnet Synchronous Motor Field Orientation Control (PMSM FOC) demos for simulation and implementation, including asynchronous scheduling and flash memory download

For details, see the product-specific release notes.

Embedded Target for TI C6000™ DSP

- Support for C6455 DSK, C6455 EVM, and TCI6482 EVM targets
- C6727 PADK baseline code generation with CCS 3.1 and BIOS 5.20.05
- D.signT DSK-91C111 daughter card support for C6416 DSK, C6713 DSK, and C6711 DSK
- New TCP/IP target-side blocks and enhanced target-side UDP blocks support variable length messages
- Enhanced support for C6416 DSK, C6711 DSK, C6713 DSK, and DM642 EVM targets
- New Memory Allocate, Memory Copy, and EDMA blocks for managing DSP memory and transferring data
- New demos:
 - Custom Device Driver via Legacy Code Integration
 - User-Controlled Memory Mapping
 - Using the PCI Bus to Exchange Data with xPC

For details, see the product-specific release notes.

Excel Link

- New MLGetFigure function to import current MATLAB figure into Excel spreadsheet
- New MLStartDir function to specify directory where MATLAB will reside after starting
- New MLUseFullDesktop function to specify whether to use full MATLAB desktop or only the command window

- New `MShowMatlabErrors` function used by `MEvalString` to return standard Excel Link errors or full MATLAB errors

For details, see the product-specific release notes.

Filter Design HDL Coder

- Distributed Arithmetic support for FIR Filters in HDL code generated for single-rate and multirate FIR filters
- Multirate support for generation of fully serial architectures for direct-form FIR polyphase decimators and interpolators
- Generate HDL dialog box support for full range of serial, parallel, and distributed arithmetic architecture options

For details, see the product-specific release notes.

Filter Design Toolbox

- Blocks for designing filters added to Signal Processing Blockset (Filter Design Toolbox required to use the new blocks to design and implement filters)
- Support for automated conversion of floating-point filters to fixed-point, based on input data
- Support for designing floating-point and fixed-point fractional-delay Farrow filters
- New `filterbuilder` function for interactive filter design
- Support for designing lattice wave digital filters
- Scaled doubles logging
- New `freqrespest` function for using filtering to estimate the filter frequency response from measured data
- Parallel filter objects support multirate filters

For details, see the product-specific release notes.

Financial Toolbox

- New functions for computing common investment performance and risk-adjusted metrics: `sharpe`, `inforatio`, `portalpha`, `lpm`, `elpm`
- New graphical user interface (`ftstool`) to support loading, manipulation, and analysis of time series (FINTS) objects (interoperates with Database Toolbox, Datafeed Toolbox, Financial Time Series GUI (`ftsgui`), and `chartfts` interface)

For details, see the product-specific release notes.

Fixed Point Toolbox

- Support for use of `fi` objects with the `DataType` property set to double without a Fixed-Point Toolbox license, when the `fipref LoggingMode` property is set to off
- Fixed-point square root support
- Limited dot notation support for `fi` objects added to Embedded MATLAB Function block in Simulink
- `get` function support for use with `fi` objects added to the Embedded MATLAB Function block in Simulink
- New default syntax for `fi` objects

For details, see the product-specific release notes.

GARCH Toolbox

- New Hodrick-Prescott filter for separating time series into trend and cyclical components
- New demo that reproduces the results in original paper on U.S. business cycles by Hodrick and Prescott

For details, see the product-specific release notes.

Image Acquisition Toolbox

- Support for Hamamatsu cameras and frame grabbers

For details, see the product-specific release notes.

Image Processing Toolbox

- New `dicomlookup` function for searching DICOM data dictionary for attributes
- `dicominfo` performance significantly improved
- `graycomatix` symmetric option that produces graylevel co-occurrence matrix (GLCM) that is symmetric across its diagonal
- `iccread` support for named colors in ICC profiles
- `imdistanline` use of a different cursor shape at its endpoints to enhance usability

For details, see the product-specific release notes.

Link for Code Composer Studio™

Version 2.0 of the Link for Code Composer Studio, released via the Web after Release 2006a, includes these features:

- In-process plug-in for Code Composer Studio
- New architecture for communication and data transfer between Code Composer Studio and MATLAB and Simulink
- Demo versions of RTDX Send and RTDX Receive blocks to construct Simulink software test benches to verify and validate DSP code

Version 2.1, part of Release 2006b, includes these features:

- Extended support to two Code Composer Studio versions
 - Added CCS 3.2 (for the C64x+ processor family)
 - Continuing support for CCS 3.1 (C6000, C5000, C2000, Rxx, OMAP processor families)
- New Function Call block in new demo — Comparing Simulation and Target Implementation with Function Call

- New demo featuring Debug component — Comparing Simulation and Target Implementation with Debug Methods
- C64x processor family supported by all demos

For details, see the product-specific release notes.

Link for ModelSim®

- Linux 64-bit operating systems supported

For details, see the product-specific release notes.

Mapping Toolbox

- Standard formulations of five major map projections
- Two new geodetic/geocentric latitude conversion functions: `geocentric2geodeticlat` and `geodetic2geocentriclat`
- Accelerated performance for `geoshow`, `mapshow`, and `bufferm` functions

For details, see the product-specific release notes.

MATLAB® Builder for .NET

- New Deployment Tool graphical user interface for creating and building projects
- Easier referencing of `MWArray` assembly via standard Microsoft Visual Studio `.NET` referencing tab
- Support for specifying which version of .NET framework to use when compiling a component (1.1 or 2.0)
- Support for shared assemblies
- Support for nested namespaces

For details, see the product-specific release notes.

MATLAB® Builder for Excel

- New Deployment Tool graphical user interface for creating and building projects

For details, see the product-specific release notes.

MATLAB® Compiler

- New Deployment Tool graphical user interface for creating and building projects
- Support for converting MATLAB M-files to Java components, in conjunction with the optional MATLAB Builder for Java product
- Support for Windows x64

For details, see the product-specific release notes.

MATLAB® Distributed Computing Engine

- Support for MPI (parallel jobs) with third-party scheduler or MathWorks job manager
- Enhanced error detection for MPI (parallel jobs)
- Support for Windows Compute Cluster Server (CCS)
- Support for Win64
- Interactive Parallel Mode (pmode)
- New distributed array structure to support parallel algorithms

For details, see the product-specific release notes.

Model-Based Calibration Toolbox

Optimization

- Gradient constraints for controlling table smoothness in optimization-based and feature-based table filling
- Contour plot view for optimization results
- Point-by-point variable bounds for finer control of optimization problems
- Ability to enable and disable optimization constraints for “what-if” studies

- Support for starting optimizations from results of a previous run, to solve related “what-if” problems faster and to refine calibration results by tweaking algorithm settings
- Enhanced user interface for comparing results of different calibration optimization problems

Modeling

- Improved integration of validation metrics within model selection process
- Enhanced local modeling at each operating point (a complementary approach to building a single, global model)
- Ability to alter variable ranges and refit models automatically

For details, see the product-specific release notes.

Optimization Toolbox

- New Optimization Tool (`optimtool`), a graphical user interface for performing common optimization tasks, including
 - Select a solver and define your optimization problem
 - Set and inspect optimization options
 - Run problems and visualize results
 - Import and export your problem definitions, algorithm options, and results between the MATLAB workspace and Optimization Tool
 - Automatically generate M-code to capture your work and automate tasks
- New plot option for displaying predefined or customized measures of progress as the solver is executing

For details, see the product-specific release notes.

Real-Time Windows Target

- Support for National Instruments M-series boards
- Support for new Humusoft AD622 and MF624 boards

For details, see the product-specific release notes.

Real-Time Workshop®

- Support for Simulink Report Generator to create code generation reports
- New packNGo function for relocating static and generated code files for a model
- Support for new `Simulink.SubSystem.getChecksum` method for determining why subsystem code is not reused
- Merge block enhancement that supports storage classes for input signals
- Consistency improvements to formatting of generated code
- Support for Simulink Legacy Code Tool, which speeds creation of S-functions from legacy C/C++ code

For details, see the product-specific release notes.

Real-Time Workshop® Embedded Coder

- Element-by-element optimized code for vector operations
- Improved efficiency and readability of for loops generated for wide signals that have multiple sources
- Unnecessary temporary variables no longer generated for muxed signals
- Fixed-point code generation support for enhanced N-D lookup tables
- Ability to control use of parentheses in generated code
- Enhanced HTML code report performance and content
- New general-purpose OSEK/VDX® real-time operating system (RTOS) target example

For details, see the product-specific release notes.

RF Toolbox

- New objects and functions to support signal integrity engineering
- Support for converting S-parameters into the transfer function
- Support for fitting a rational function to component or network data
- Support for computing the frequency response and impulse response of the component or network

- Support for exporting a Verilog-A model for use in a time-domain circuit simulator
- Three new demos to show the use of the objects and functions for modeling backplanes

For details, see the product-specific release notes.

Robust Control Toolbox

- New `ltiarray2uss` function to construct an uncertain state-space system for any LTI array

For details, see the product-specific release notes.

Signal Processing Blockset

- New Filter Design Toolbox block library for design and implementation of single-rate and multirate FIR and IIR filters
- Enhanced audio processing support
- New diagnostic output port added to Pseudoinverse, Singular Value Decomposition, and SVD Solver blocks to report failure to converge
- Fixed-point support added to Backward Substitution, Forward Substitution, LDL Factorization, and LU Factorization blocks
- 2-D support added to Matrix Product, Matrix Sum, Maximum, and Minimum blocks
- Multichannel support added to the Autocorrelation LPC, Levinson-Durbin, LSF/LSP to LPC, Yule-Walker AR Estimator, and Zero Crossing blocks

For details, see the product-specific release notes.

Signal Processing Toolbox

- Spectral analysis functions support frequency vector input
- FFT Length parameter moved from spectrum objects to spectrum estimation methods
- `sosfilt` and `dfilt` filter methods support multidimensional array input
- `dfilt` block method support for specifying simulink subsystem destination and linking command-line filter to Simulink blocks

For details, see the product-specific release notes.

SimBiology™

Version 2.0 of SimBiology, released via the Web after R2006a, includes the following features. No additional new features were added in Version 2.0.1, released as part of R2006b.

- Block diagram editor for constructing pathways
- Functionality for performing sensitivity analysis
- Functionality for performing parameter estimation
- Ability to run ensemble runs
- Ability to determine conserved moieties

For details, see the product-specific release notes.

SimEvents™

- New Event-Based Sequence Generator block
- New history options and significant acceleration of scope blocks
- Variable delay in event translation block
- Unlimited-port routing
- Compatibility with Simulink Accelerator
- New demos related to real-time operating systems, Ethernet-based network modeling, CAN communication workflow modeling, and Automatic Repeat Request based communications using Stateflow

For details, see the product-specific release notes.

SimHydraulics™

SimHydraulics was released via the Web after R2006a, but before R2006b.

Version 1.1 of SimHydraulics, included in R2006b, includes these new features:

- Linearization command support
- New solver option to start transient analysis from a steady state
- Extended functionality for working with physical units
- 13 new blocks

For details, see the product-specific release notes.

SimMechanics

- Computer-aided design (CAD) translator for Pro/ENGINEER

For details, see the product-specific release notes.

SimPowerSystems

- New Mechanical Input parameter for the AC Electric Drive blocks, for connecting SimMechanics or SimDriveline to electric drives by specifying motor speed as block input

For details, see the product-specific release notes.

Simulink® Fixed Point

- New Prelookup block and Interpolation Using Prelookup block support for fixed-point data types
- SQRT operation in Math Function block support for fixed-point data types

For details, see the product-specific release notes.

Simulink® Report Generator

- Two new components for Real-Time Workshop: Code Generation Summary and Import Generated Code

For details, see the product-specific release notes.

Simulink® Response Optimization

- For response optimization in Simulink Control Design, support for specifying rise-time, overshoot, and settling-time requirements on step response plots

For details, see the product-specific release notes.

Simulink® Verification and Validation

- Requirement consistency checks added to Model Advisor, ensuring that requirement links are valid
- Modeling standards checks added to Model Advisor, ensuring Simulink block diagrams adhere to particular style guidelines

For details, see the product-specific release notes.

Spline Toolbox

- Significant speedup in spline evaluation

For details, see the product-specific release notes.

Stateflow®

- Support for interfacing Simulink bus signals with Stateflow charts and Truth Table blocks
- Support for structure data types specified as `Simulink.Bus` objects
- New chart property for constraining finite state machines to use either Mealy or Moore semantics
- MATLAB language support
 - Support for structures as inputs, outputs, and persistent variables in Embedded MATLAB functions
 - Continued expansion of the Embedded MATLAB runtime library with 36 new functions added for code generation, including `std`, `cov`, `rand`, and `histc`

For details, see the product-specific release notes.

Stateflow® Coder

- Support for code generation of structures to represent Simulink.Bus objects

For details, see the product-specific release notes.

Statistics Toolbox

- New visualization capabilities — main-effects plots, two-factor interaction plots, and multivariate charts for Design of Experiment results
- Expanded statistical process control functionality — gage repeatability and reproducibility studies, eight new control charts, control rules for control chart data, and enhanced capability analysis functions
- New regression functionality — multivariate regressions with missing data and multinomial regressions of discrete multicategory data, including multinomial logistic regression
- Expanded probability distributions — multinomial PDF and random number generator
- Enhanced hypothesis testing — new function to calculate sample size necessary for a hypothesis test to have a specified power, and improvements to Lilliefors and Jarque-Bera distribution tests to accommodate a wider range of sample sizes, significance levels, and probability distributions

For details, see the product-specific release notes.

System Identification Toolbox

- Support for MATLAB Compiler

For details, see the product-specific release notes.

Video and Image Processing Blockset

- New Contrast Adjustment, Demosaic, and Trace Boundaries blocks
- Draw Markers block support for antialiasing
- Resize block support for region-of-interest (ROI) processing
- Shear and Translate block support for specifying intensity of background pixels

- From Multimedia File block support for AVI files on UNIX platforms
- New demos: Abandoned Object Detection and Concentricity Inspection
- Enhanced demos: Pattern Matching and fixed-point versions of Lane Departure Warning and People Tracking demos

For details, see the product-specific release notes.

Virtual Reality Toolbox

- Text Output Block added

For details, see the product-specific release notes.

Wavelet Toolbox

- Multivariate denoising graphical user interface (GUI) and functions added
- Multiscale principal component analysis GUI and functions added
- Several new demos

For details, see the product-specific release notes.

xPC Target

Version 3.0 of the xPC Target, released via the Web after R2006a, includes these features:

- Support for additional Ethernet chips: National Semiconductor DP83815, 3Com 3C90x, Realtek RTL8139D
- Enhanced support of To/From Blocks and Target PCs
- Support for new I/O boards
- New xPC Target demos
- Support for general-purpose counter I/O option, xPC TargetBox® IO309, based on MPC-555 TPU

Version 3.1, part of R2006b, includes these features:

- Support for Texas Instruments DM642 EVM, in conjunction with Embedded Target for TI C600 DSP
- Support for additional Ethernet chip: Intel I82547EI

- Support for Microsoft Visual C/C++ Version 8.0
- New demos

For details, see the product-specific release notes.

New Products

Six new products will be available in R2006b.

Aerospace Toolbox	Aerospace reference standards, environment models, and aerodynamic coefficient importing
Link for TASKING®	Build, test, and verify embedded code using TASKING
MATLAB® Builder for Java™	Create MATLAB® based Java classes
SimHydraulics™	Model and simulate hydraulic systems
Simulink® HDL Coder	Generate synthesizable HDL code and test-benches from Simulink® models and Stateflow® diagrams
SystemTest	Manage tests and analyze results for system verification and validation

Aerospace Toolbox

The Aerospace Toolbox provides reference standards, environment models, and aerodynamic coefficient importing for performing advanced aerospace analysis to develop and evaluate your designs. An interface to FlightGear flight simulator lets you visualize flight data in a three-dimensional environment and reconstruct behavioral anomalies in flight-test results. To ensure design consistency, Aerospace Toolbox provides utilities for unit conversions, coordinate transformations, and quaternion math, as well as standards-based environment models for the atmosphere, gravity, and magnetic fields. You can import aerodynamic coefficients from the U.S. Air Force Digital Data Compendium (Datcom) to carry out preliminary control design and vehicle performance analysis.

Key Features

- Provides standards-based environment models for atmosphere, gravity, and magnetic fields
- Converts units and transforms coordinate systems and spatial representations
- Implements predefined utilities for aerospace parameter calculations, time calculations, and quaternion math
- Imports aerodynamic coefficients directly from the U.S. Air Force Digital Data Compendium (Datcom)
- Interfaces to FlightGear flight simulator, enabling visualization of vehicle dynamics in a three-dimensional environment

For details, see the product-specific release notes.

Link for TASKING®

Note Link for TASKING 1.0 was released via the Web after R2006a, but before R2006b.

Link for TASKING lets you build, test, and verify embedded code using MATLAB and Simulink with TASKING (from Altium). Link for TASKING makes it easy to use a test harness model in Simulink to verify object code executing within the TASKING integrated development environment. This processor-in-the-loop (PIL) test operation works with code automatically generated from Simulink models.

Key Features

- Links MATLAB and Simulink to a TASKING environment through high-speed bidirectional links
- Supports Infineon, STMicroelectronics, Renesas, ARM, Freescale, and other microprocessors
- Enables processor-in-the-loop testing for object code verification
- Supports automatic code generation, building, and downloading to an instruction-set simulator or embedded hardware

- Provides MATLAB APIs to analyze and debug code generated automatically or by hand
- Includes customizable templates for configuring hardware variants, automating MISRA C-code checking, and controlling the build process

For details, see the product-specific release notes.

MATLAB® Builder for Java™

MATLAB Builder for Java extends the MATLAB Compiler with tools for automatically generating Java classes from your MATLAB algorithms. You can run MATLAB based classes created using Builder for Java outside the MATLAB environment, referencing them the same way as any other Java class.

Key Features

- Converts MATLAB algorithms into Java classes via a graphical user interface or the command line
- Supports conversion between Java data types and MATLAB array data types using data conversion classes
- Enables royalty-free desktop and Web deployment of independent components

For details, see the product-specific release notes.

SimHydraulics™

Note SimHydraulics 1.0 was released via the Web after R2006a, but before R2006b.

SimHydraulics extends Simulink with tools for modeling and simulating hydraulic power and control systems. It enables you to describe multidomain systems containing connected hydraulic and mechanical components as physical networks. It provides a representative library of hydraulic components and building blocks that lets you implement other components. SimHydraulics can be used for a variety of automotive, aerospace, defense, and

industrial equipment applications, such as modeling automatic transmissions, actuating flight control surfaces, and actuating heavy equipment.

Key Features

- Modeling environment for building hydraulic and hydromechanical systems as physical networks
- More than 75 hydraulic and mechanical components, including pumps, valves, accumulators, and pipelines
- Foundation library of hydraulic building blocks and fundamental mechanical and mathematical elements
- Customizable library of common hydraulic fluids

For details, see the product-specific release notes.

Simulink® HDL Coder

Simulink HDL Coder generates bit-true and cycle-accurate, synthesizable Verilog and VHDL code from Simulink models and Stateflow diagrams. The automatically generated HDL code is target independent.

You can simulate and synthesize the automatically generated HDL using industry standard tools, and then map it into field-programmable gate arrays (FPGAs) or application-specific integrated circuits (ASICs). You can also use the automatically generated HDL code to verify existing HDL code using formal or functional verification tools.

Simulink HDL Coder also generates test benches, enabling rapid verification of the generated HDL code using HDL simulation tools.

Key Features

- Generates synthesizable HDL code from Simulink models for data path implementations
- Generates synthesizable HDL code from Stateflow diagrams for finite state machines and control logic implementations
- Generates IEEE -1076 compliant VHDL code and IEEE -1364-2001 compliant Verilog code
- Lets you create bit-true and cycle-accurate models that match your Simulink design specifications

- Incorporates multirate Simulink models in one HDL clock domain
- Lets you select from multiple HDL architectural implementations for commonly used blocks
- Lets you specify the subsystem for HDL code generation
- Enables you to reuse existing IP HDL code (with Link for ModelSim®, available separately)
- Generates simulation and synthesis scripts

For details, see the product-specific release notes.

SystemTest

Note SystemTest 1.0 was released via the Web after R2006a, but before R2006b.

SystemTest provides a software framework for developing tests that exercise MATLAB algorithms and Simulink models. It includes predefined test elements that let you build and maintain standard test routines. You can save and share tests throughout a development project, ensuring standard and repeatable test verification from research and development through preproduction. You can map test variables into a result set for analysis.

Key Features

- Develops, manages, and edits test structures
- Uses predefined test elements to test MATLAB algorithms and Simulink models
- Incorporates MATLAB scripts directly into a test
- Maps MATLAB and Simulink workspace variables to test variables
- Visualizes and analyzes multidimensional test results

For details, see the product-specific release notes.

R2006b Products with License-Related Changes

The following products will undergo restructuring in R2006b:

- Aerospace Blockset
- Embedded Target for Infineon C166® Microcontrollers
- Embedded Target for OSEK/VDX

Aerospace Blockset

With the introduction of R2006b, Aerospace Blockset no longer requires Control System Toolbox, but does require a new product, Aerospace Toolbox.

Aerospace Toolbox will be provided at no initial cost to Aerospace Blockset customers whose maintenance is up-to-date.

Embedded Target for Infineon C166® Microcontrollers

With the introduction of R2006b, Embedded Target for Infineon C166® Microcontrollers requires Link for TASKING®.

Link for TASKING will be provided at no initial cost to Embedded Target for Infineon C166® Microcontrollers customers whose maintenance is up-to-date.

Embedded Target for OSEK/VDX®

With the introduction of R2006b, a subset of the Embedded Target for OSEK/VDX® functionality is incorporated into a demonstration model for Real-Time Workshop Embedded Coder. The demonstration model can be used as a starting point for targeting specific OSEK/VDX implementations.

Embedded Target for OSEK/VDX is no longer available for purchase as a separate product.

Discontinued Products

The following products are being discontinued and are not included with R2006b:

- CDMA Reference Blockset
- MATLAB® Web Server

CDMA Reference Blockset

With the introduction of R2006b, CDMA Reference Blockset is no longer available for purchase and no new feature development or further bug fixes will occur. Technical support will be offered on a limited basis, as available. Contact your sales representative with additional questions.

MATLAB® Web Server

With the introduction of R2006b, MATLAB Web Server is no longer available for purchase and no new feature development or further bug fixes will occur. Technical support will be offered on a limited basis, as available. Contact your sales representative with additional questions.

Compatibility Considerations

These topics summarize the potential compatibility considerations when upgrading from R2006a to R2006b:

- “R2006b Compatibility Assessment” on page 33
- “Compatibility for Each Product” on page 35
- “Compatibility Considerations for New Functions and Models” on page 35

R2006b Compatibility Assessment

Upgrading from R2006a to R2006b is not expected to cause any significant compatibility problems for most users. The compatibility considerations likely to be noticed by the broadest cross-section of users are summarized in the following table. Refer to product-specific release notes for details.

Product and Area	Compatibility Consideration	Potential Impact	Recommended Actions
MATLAB— Creating Graphical User Interfaces	Obsolete functions: axlimd1g, edtext, menubar, paged1g, umtoggle.	Usage generates a warning in Version 7.3 (R2006b), but will error in a future release.	Replace occurrences of the functions.
MATLAB— Creating Graphical User Interfaces	Background color of user interface control (uicontrol) push buttons and toggle buttons on Windows XP are now flat colored buttons.	The background color of a push button or toggle button that was set to any color other than the factory color on Windows XP changes from a colored border around the button.	None.
MATLAB— External Interfaces	Internal storage of sparse arrays on 64-bit machines changed.	Existing MEX-files that use certain sparse functions will not work on 64-bit systems in Version 7.3 (R2006b).	Change declaration statements in your source code and recompile MEX-files that interact with sparse arrays.

Product and Area	Compatibility Consideration	Potential Impact	Recommended Actions (Continued)
MATLAB— Programming	save function no longer supports compress, nocompress, unicode, and nunicode options.	Code that uses save with any of these options will error.	Modify the code to use the v6 option instead.
MATLAB— External Interfaces	actxcontrol command now validates ProgID.	Some server ProgIDs that worked with actxcontrol now return an error.	Correct the ProgID or disable the validation checking.
MATLAB— External Interfaces	-v5 option to mex to be removed.	In a future version, you will not be able to build a MEX-file that is compatible with MATLAB Version 5.	Migrate to the current version.
Simulink	Integrator block: level reset option changed.	Model simulations that specify this option could produce results different from those of previous releases.	To reproduce results from previous releases, change this option to level hold.
Real-Time Workshop	Changed default settings for certain multitasking diagnostic options to error.	For models created with a version of Simulink before V6.5 (R2006b), using a value other than error for certain situations could generate code that might corrupt data or produce unpredictable behavior.	A dialog box reports the issue and suggests that you change the diagnostic setting to error.

Compatibility for Each Product

To see if any known compatibility considerations exist for your product, see “Release Summary” on page 37. If the table shows that there are compatibility considerations for any of your products, follow the link to the product’s release notes and refer to the summary table of compatibility considerations. From this table you can link to details. If you are upgrading from a release before R2006a, also see the entries in the table for earlier versions.

Compatibility Considerations for New Functions and Models

The introduction of new functions and models could cause a conflict with any of your own M-files, models, and variables having the same names.

Example

If you created a function named `ld1` in a previous release, it might conflict with the new MATLAB `ld1` function (introduced in R2006b). Another conflict that might arise is when you load a MAT-file that contains a variable named `ld1`.

To identify and address name conflicts, see these topics in the MATLAB Programming documentation:

- Don’t Use Function Names for Variables
- Naming a Function Uniquely

System Requirements

No Major Supported Platform Changes

There are no major supported platform changes since R2006a, although in some cases more recent versions of operating systems and other software are now supported.

For information on system requirements, visit
<http://www.mathworks.com/support/sysreq/r2006b/>

Matrix Size Limitations

Historically, MATLAB matrices have been limited in size to those that would fit in a 32-bit address space. Some of those limitations have been lifted, but some remain. Specifically, the data type used to index into an `mxArray` is still a 32-bit signed integer. This limits the number of elements in any one array to `INT_MAX - 1`, or 2147483646 (approximately 2×10^9). With this limit, you can create matrices up to 16 GB (for doubles). You can create as many of these as your machine has memory to support.

For Linux: CDF Writing Not Supported

CDF writing is not supported, but you can read CDF files.

Sun Solaris Run-Time Libraries

MATLAB dynamically links against the Solaris C++ Run-Time Library, which must reside on the same machine.

To determine whether this library is already installed, type the following line in a UNIX shell. (If you get a `No match` response, then you need to install it from your Solaris system CD.)

```
ls -l /usr/lib/libCstd.so.*
```

Release Summary

Note An asterisk (*) after a product name indicates the product has had a Web release since R2006a.

Product (Links to Release Notes)	New Features	Major Bug Fixes	Compatibility Considerations
MATLAB®	Yes	Yes	Yes
Simulink®*	Yes	Yes	Yes
Aerospace Blockset (see “R2006b Products with License-Related Changes” on page 31)	Yes	Yes	No
Aerospace Toolbox (see “New Products” on page 26)	N/A	N/A	N/A
Bioinformatics Toolbox*	Yes	Yes	Yes
CDMA Reference Blockset (see “Discontinued Products” on page 32)	N/A	N/A	N/A
Communications Blockset	Yes	Yes	Yes
Communications Toolbox	Yes	No	Yes
Control System Toolbox	Yes	Yes	No
Curve Fitting Toolbox	No	Yes	No
Data Acquisition Toolbox	No	Yes	Yes

Product (Links to Release Notes)	New Features	Major Bug Fixes	Compatibility Considerations
Database Toolbox	Yes	No	No
Datafeed Toolbox	Yes	No	No
Distributed Computing Toolbox	Yes	Yes	Yes
Embedded Target for Infineon® C166 Microcontrollers (see "R2006b Products with License-Related Changes" on page 31)	Yes	Yes	Yes
Embedded Target for Motorola® HC12	No	Yes	No
Embedded Target for Motorola® MPC555	No	Yes	Yes
Embedded Target for OSEK/VDX (see "R2006b Products with License-Related Changes" on page 31)	N/A	N/A	N/A
Embedded Target for TI C2000™ DSP	Yes	Yes	No
Embedded Target for TI C6000™ DSP	Yes	Yes	Yes
Excel Link	Yes	No	No
Extended Symbolic Math Toolbox	No	Yes	Yes
Filter Design HDL Coder	Yes	Yes	Yes
Filter Design Toolbox	Yes	Yes	Yes

Product (Links to Release Notes)	New Features	Major Bug Fixes	Compatibility Considerations
Financial Derivatives Toolbox	No	No	No
Financial Toolbox	Yes	No	No
Fixed-Income Toolbox	No	No	No
Fixed-Point Toolbox	Yes	No	No
Fuzzy Logic Toolbox (no release notes)	No	Yes	No
GARCH Toolbox	Yes	No	No
Gauges Blockset	No	No	No
Genetic Algorithm and Direct Search Toolbox	No	Yes	Yes
Image Acquisition Toolbox	Yes	Yes	No
Image Processing Toolbox	Yes	Yes	Yes
Instrument Control Toolbox	No	Yes	No
Link for Code Composer Studio™ Development Tools*	Yes	Yes	Yes
Link for ModelSim®	Yes	No	No
Link for TASKING®* (see “New Products” on page 26)	No	No	Yes
Mapping Toolbox	Yes	Yes	Yes

Product (Links to Release Notes)	New Features	Major Bug Fixes	Compatibility Considerations
MATLAB® Builder for Excel	Yes	Yes	No
MATLAB® Builder for Java™ (see “New Products” on page 26)	N/A	N/A	N/A
MATLAB® Builder for .NET	Yes	Yes	No
MATLAB® Compiler	Yes	Yes	No
MATLAB® Distributed Computing Engine	Yes	Yes	Yes
MATLAB® Report Generator	No	Yes	No
MATLAB® Web Server (see “Discontinued Products” on page 32)	N/A	N/A	N/A
Model-Based Calibration Toolbox	Yes	Yes	No
Model Predictive Control Toolbox	No	No	No
Neural Network Toolbox	No	No	No
OPC Toolbox	No	Yes	No
Optimization Toolbox	Yes	Yes	No
Partial Differential Equation Toolbox (no release notes)	No	No	No
Real-Time Windows Target	Yes	No	No

Product (Links to Release Notes)	New Features	Major Bug Fixes	Compatibility Considerations
Real-Time Workshop®*	Yes	Yes	Yes
Real-Time Workshop® Embedded Coder*	Yes	Yes	Yes
RF Blockset	No	No	No
RF Toolbox	Yes	No	No
Robust Control Toolbox	Yes	No	No
Signal Processing Blockset	Yes	Yes	Yes
Signal Processing Toolbox	Yes	Yes	Yes
SimBiology™*	Yes	Yes	No
SimDriveline	No	Yes	No
SimEvents™	Yes	Yes	Yes
SimHydraulics™* (see “New Products” on page 26)	Yes	Yes	No
SimMechanics	Yes	Yes	No
SimPowerSystems	Yes	Yes	No
Simulink® Accelerator*	No	No	No
Simulink® Control Design	No	No	No
Simulink® Fixed Point*	Yes	No	No
Simulink® HDL Coder (see “New Products” on page 26)	N/A	N/A	N/A

Product (Links to Release Notes)	New Features	Major Bug Fixes	Compatibility Considerations
Simulink® Parameter Estimation	No	No	No
Simulink® Report Generator*	Yes	Yes	No
Simulink® Response Optimization	Yes	Yes	No
Simulink® Verification and Validation	Yes	Yes	No
Spline Toolbox	Yes	No	No
Stateflow® and Stateflow® Coder*	Yes	Yes	No
Statistics Toolbox	Yes	Yes	Yes
Symbolic Math Toolbox	No	Yes	Yes
System Identification Toolbox	Yes	Yes	No
SystemTest* (see "New Products" on page 26)	No	Yes	N/A
Video and Image Processing Blockset	Yes	Yes	Yes
Virtual Reality Toolbox	Yes	Yes	No
Wavelet Toolbox	Yes	Yes	No
xPC Target*	Yes	Yes	Yes
xPC TargetBox®	No	No	No

