MATLAB®
Desktop Tools and Development Environment

R2012b
## Revision History

<table>
<thead>
<tr>
<th>Date</th>
<th>Type</th>
<th>Updates</th>
</tr>
</thead>
<tbody>
<tr>
<td>June 2004</td>
<td>First printing</td>
<td>New for MATLAB 7.0 (Release 14). Formerly part of Using MATLAB.</td>
</tr>
<tr>
<td>October 2004</td>
<td>Online only</td>
<td>Revised for MATLAB 7.0.1 (Release 14SP1)</td>
</tr>
<tr>
<td>March 2005</td>
<td>Online only</td>
<td>Revised for MATLAB 7.0.4 (Release 14SP2)</td>
</tr>
<tr>
<td>March 2005</td>
<td>Second printing</td>
<td>Revised for MATLAB 7.0.4 (Release 14SP2)</td>
</tr>
<tr>
<td>June 2005</td>
<td>Third printing</td>
<td>Minor revision for MATLAB 7.0.4 (Release 14SP2)</td>
</tr>
<tr>
<td>September 2005</td>
<td>Online only</td>
<td>Revised for MATLAB 7.1 (Release 14SP3)</td>
</tr>
<tr>
<td>March 2006</td>
<td>Online only</td>
<td>Revised for MATLAB 7.2 (Release 2006a)</td>
</tr>
<tr>
<td>September 2006</td>
<td>Online only</td>
<td>Revised for MATLAB 7.3 (Release 2006b)</td>
</tr>
<tr>
<td>March 2007</td>
<td>Online only</td>
<td>Revised for MATLAB 7.4 (Release 2007a)</td>
</tr>
<tr>
<td>September 2007</td>
<td>Online only</td>
<td>Revised for MATLAB 7.5 (Release 2007b)</td>
</tr>
<tr>
<td>March 2008</td>
<td>Online only</td>
<td>Revised for MATLAB 7.6 (Release 2008a)</td>
</tr>
<tr>
<td>October 2008</td>
<td>Online only</td>
<td>Revised for MATLAB 7.7 (Release 2008b)</td>
</tr>
<tr>
<td>March 2009</td>
<td>Online only</td>
<td>Revised for MATLAB 7.8 (Release 2009a)</td>
</tr>
<tr>
<td>September 2009</td>
<td>Online only</td>
<td>Revised for MATLAB 7.9 (Release 2009b)</td>
</tr>
<tr>
<td>March 2010</td>
<td>Online only</td>
<td>Revised for MATLAB 7.10 (Release 2010a)</td>
</tr>
<tr>
<td>September 2010</td>
<td>Online only</td>
<td>Revised for MATLAB Version 7.11 (Release 2010b)</td>
</tr>
<tr>
<td>April 2011</td>
<td>Online only</td>
<td>Revised for MATLAB Version 7.12 (Release 2011a)</td>
</tr>
<tr>
<td>September 2011</td>
<td>Online only</td>
<td>Revised for MATLAB Version 7.13 (Release 2011b)</td>
</tr>
<tr>
<td>March 2012</td>
<td>Online only</td>
<td>Revised for MATLAB Version 7.14 (Release 2012a)</td>
</tr>
<tr>
<td>September 2012</td>
<td>Online only</td>
<td>Revised for MATLAB Version 8.0 (Release 2012b)</td>
</tr>
</tbody>
</table>
## Startup and Shutdown

### Starting and Exiting the MATLAB Program

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Starting the MATLAB Program on Windows Platforms</td>
<td>1-2</td>
</tr>
<tr>
<td>Starting the MATLAB Program on Linux Platforms</td>
<td>1-6</td>
</tr>
<tr>
<td>Starting the MATLAB Program on Macintosh Platforms</td>
<td>1-6</td>
</tr>
<tr>
<td>Exiting the MATLAB Program</td>
<td>1-7</td>
</tr>
</tbody>
</table>

### Startup Folder for the MATLAB Program

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>What Is the Startup Folder?</td>
<td>1-12</td>
</tr>
<tr>
<td>Startup Folder on Windows Platforms</td>
<td>1-13</td>
</tr>
<tr>
<td>Startup Folder on Linux Platforms</td>
<td>1-14</td>
</tr>
<tr>
<td>Startup Folder on Macintosh Platforms</td>
<td>1-14</td>
</tr>
<tr>
<td>Changing the Startup Folder</td>
<td>1-14</td>
</tr>
</tbody>
</table>

### Startup Options

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specifying MATLAB Startup Options</td>
<td>1-18</td>
</tr>
<tr>
<td>Commonly Used Startup Options</td>
<td>1-20</td>
</tr>
<tr>
<td>Passing Perl Variables on Startup</td>
<td>1-21</td>
</tr>
<tr>
<td>Startup and Calling Java Software from the MATLAB Program</td>
<td>1-22</td>
</tr>
</tbody>
</table>

### Toolbox Path Caching in the MATLAB Program

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>About Toolbox Path Caching in the MATLAB Program</td>
<td>1-23</td>
</tr>
<tr>
<td>Using the Cache File Upon Startup</td>
<td>1-23</td>
</tr>
<tr>
<td>Updating the Cache and Cache File</td>
<td>1-23</td>
</tr>
<tr>
<td>Additional Diagnostics with Toolbox Path Caching</td>
<td>1-26</td>
</tr>
</tbody>
</table>

## Desktop

### Adjust Desktop Appearance

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fonts</td>
<td>2-2</td>
</tr>
</tbody>
</table>
Color Settings ........................................ 2-7
Access Frequently Used Features ...................... 2-10
Optimize Desktop Layout for Limited Screen Space ... 2-11

Define Keyboard Shortcuts .......................... 2-15
  Keyboard Shortcuts ................................ 2-15
  Choose a Set of Keyboard Shortcuts ................. 2-16
  Compare Sets of Keyboard Shortcuts ................. 2-19
  Display Keyboard Shortcuts ......................... 2-21
  Customize Keyboard Shortcuts ...................... 2-24
  Evaluate and Resolve Keyboard Shortcut Conflicts ... 2-29
  Examples of Creating, Modifying, and Deleting Keyboard
    Shortcuts ...................................... 2-31
  Delete a Set of Keyboard Shortcuts .................. 2-35
  Use Keyboard Shortcuts Settings Files Created on Other
    Systems ....................................... 2-35
  Keyboard Shortcut Restrictions ..................... 2-36

Set Print Options .................................... 2-39
  Page Setup Options ................................ 2-39
  Layout Options for Page Setup ...................... 2-39
  Header Options for Page Setup ...................... 2-40
  Fonts Options for Page Setup ...................... 2-40

Web Browsers and MATLAB .......................... 2-42
  About Web Browsers and MATLAB ................. 2-42
  Display Pages in Web Browsers ..................... 2-44
  Specify Proxy Server Settings for Connecting to the
    Internet ..................................... 2-44
  Specify the System Browser for UNIX Platforms .... 2-45

License Management and Software Updates .......... 2-47
  Manage Your Licenses ................................ 2-47
  Check for Software Updates ....................... 2-48

Macintosh Platform Conventions .................... 2-50
  Mouse Instructions and Macintosh Platforms ....... 2-50
  Navigating Within the MATLAB Root Folder on Macintosh
    Platforms .................................. 2-50

Preferences ......................................... 2-52
Set Preferences for MATLAB ........................................ 2-52
Where MATLAB Stores Preferences .......................... 2-53
Preferences Folder and Files MATLAB Uses When Multiple
MATLAB Releases Are Installed ............................ 2-54
General Preferences ........................................ 2-55
MAT-Files Preferences ....................................... 2-56
Confirmation Dialogs Preferences ............................ 2-57
Source Control Preferences .................................... 2-59
Java Heap Memory Preferences ............................... 2-59
Keyboard Shortcuts Preferences ............................... 2-60
Fonts Preferences ............................................ 2-62
Fonts Custom Preferences .................................... 2-63
Colors Preferences ............................................. 2-64
Colors Programming Tools Preferences ....................... 2-65
Toolbars Preferences ........................................... 2-66
Web Preferences .................................................. 2-67

Entering Commands

3

Enter Statements in Command Window ................................. 3-2

Find Functions Using the Function Browser .................. 3-4
What Is the Function Browser? ................................ 3-4
Steps for Using the Function Browser ............................ 3-4
Interpreting Search Results in the Function Browser ........ 3-7
Customizing the Function Browser ............................ 3-7

Format Output in Command Window .......................... 3-9
Format Line Spacing in Output ................................ 3-9
Format Floating-Point Numbers ................................ 3-10
Wrap Lines of Code to Fit Window Width ..................... 3-10
Suppress Output .................................................. 3-10
View Output by Page .......................................... 3-11
Clear the Command Window ..................................... 3-11

Stop Execution .................................................. 3-12

Find Text in Command Window or History ................. 3-13
Find Text in the Command Window ........................ 3-13
Find Text in the Command History Window .......... 3-15

Create Shortcuts to Rerun Commands .................. 3-17

Set Command Window Preferences ........................ 3-19

Set Keyboard Preferences ................................. 3-21

Check Syntax As You Type ............................... 3-23
Syntax Highlighting ...................................... 3-23
Delimiter Matching ...................................... 3-24
Tab Completion .......................................... 3-25
Function Syntax Hints ................................. 3-27

Command History ......................................... 3-29
What Is the Command History? ......................... 3-29
Using Command History Commands .................. 3-29
Changing the Command History Date Format ........... 3-31
Command History Preferences .......................... 3-31

Help and Product Information

Ways to Get Function Help ............................... 4-2

Run Examples .............................................. 4-3
Code Excerpts .......................................... 4-3
Example Scripts ........................................ 4-3

Search Syntax and Tips ................................... 4-5

Bookmark and Share Page Locations .................... 4-7
Bookmark Favorite Pages ............................... 4-7
View Page Locations .................................... 4-8
Workspace Browser and Variable Editor

What Is the MATLAB Workspace? ........................................... 5-2

View, Edit, and Copy Variables ........................................... 5-3
  View and Edit Variables .................................................. 5-3
  Copy, Paste, and Rename Variables ................................... 5-6

Keyboard Shortcuts for Navigating Variable Elements ............... 5-8

Save, Load, and Delete Workspace Variables ......................... 5-9

Statistical Calculations in the Workspace Browser ................... 5-12
  Improve Workspace Browser Performance during Statistical
  Calculations ......................................................................... 5-12
  Include or Exclude NaN Values in Statistical
  Calculations ......................................................................... 5-12

Set Workspace and Variable Preferences ............................... 5-14
  Workspace Browser Preferences .......................................... 5-14
  Variables Preferences ......................................................... 5-15
Managing Files in MATLAB

Understanding File Locations in MATLAB ............. 6-2
   Important MATLAB Folders .......................... 6-2
   Path Names in MATLAB .............................. 6-5

Working with Files and Folders ......................... 6-11
   Viewing Folder Contents ............................ 6-11
   Using the Current Folder Browser .................... 6-16

Finding Files and Folders ............................. 6-24
   Finding Files and Folders by Name in the Current
      Folder ............................................ 6-24
   Simple Search for File and Folder Names in the Current
      Folder Browser .................................... 6-24
   Advanced Search for Files — Find Files Tool ......... 6-25
   Locating a File or Folder in the Operating System
      Browser ............................................. 6-29
   Finding Files and Folders Using Functions .......... 6-30
   Additional Ways to Find Files ....................... 6-30

Creating, Opening, Changing, and Deleting Files and
Folders ............................................... 6-31
   Creating New Files and Folders ..................... 6-31
   Copying, Renaming, and Deleting Files and Folders 6-36
   Opening and Running Files .......................... 6-40
   Running External Commands, Scripts, and Programs 6-43

Comparing Files and Folders ........................... 6-47
   Comparing Files and Folders ......................... 6-47
   Comparing Folders and Zip Files ..................... 6-49
   Comparing Text Files ............................... 6-53
   Comparing Files with Autosave Version or Version on
      Disk ................................................... 6-58
   Comparing MAT-Files ............................... 6-59
   Comparing Binary Files ............................. 6-61
   Using Comparison Tool Features ...................... 6-62
   Function Alternative for Comparing Files and Folders 6-64

Making Files and Folders Accessible to MATLAB ....... 6-65
Files and Folders That MATLAB Can Access 6-65
How to Make Files Accessible 6-65
Determining if MATLAB Can Access a File 6-67
Ensuring MATLAB Uses the File You Want 6-68

Using the MATLAB Search Path 6-71
What Is the Search Path? 6-71
Viewing Files and Folders on the Search Path 6-73
Changing the Search Path 6-74
Using the Search Path with Different MATLAB Installations 6-79
Recovering from Problems with the Search Path 6-80
Handling Errors and Unexpected Behavior When Updating Folders 6-82

Editor Preferences 7

Editor/Debugger Preferences 7-2
General Preferences for the Editor/Debugger 7-2
Editor/Debugger Display Preferences 7-3
Editor/Debugger Tab Preferences 7-4
Editor/Debugger Language Preferences 7-5
Editor/Debugger Code Folding Preferences 7-8
Editor/Debugger Autosave Preferences 7-9

Code Analyzer Preferences 7-11
Code Analyzer Preferences 7-11
Searching Messages in the Code Analyzer Preferences Dialog Box 7-12

Internationalization 8

How the MATLAB Process Uses Locale Settings 8-2
Windows Platform-Specific Behavior 8-3
Macintosh Platform-Specific Behavior ................................. 8-3

Setting the Locale .................................................. 8-4
  Setting Locale on Windows Platforms ......................... 8-4
  Setting Locale on Linux Platforms .............................. 8-7
  Setting Locale on Macintosh Platforms ....................... 8-8

Troubleshooting I18n Messages and Settings .............. 8-9
  Asian Characters Incorrectly Displayed on Linux Systems ...................................................... 8-9
  Characters Incorrectly Displayed on Windows Systems .... 8-10
  datenum Might Not Return Correct Value ................... 8-10
  Numbers Display Period for Decimal Point ................. 8-10
  MATLAB Displays Messages in English ...................... 8-11
  File or Folder Names Incorrectly Displayed ............... 8-11

Index
The way you start the MATLAB® program depends on which platform you use. The following topics describe how to start up and shut down MATLAB software on all supported platforms, including information about customizing startup and shutdown.

- “Starting and Exiting the MATLAB Program” on page 1-2
- “Startup Folder for the MATLAB Program” on page 1-12
- “Startup Options” on page 1-18
- “Toolbox Path Caching in the MATLAB Program” on page 1-23
Starting and Exiting the MATLAB Program

In this section...

- “Starting the MATLAB Program on Windows Platforms” on page 1-2
- “Starting the MATLAB Program on Linux Platforms” on page 1-6
- “Starting the MATLAB Program on Macintosh Platforms” on page 1-6
- “Exiting the MATLAB Program” on page 1-7

Starting the MATLAB Program on Windows Platforms

There are several ways to start the MATLAB program on a Microsoft® Windows® platform:

- Use the shortcut on the Windows Start Menu.
  - On Windows 7 systems, if you chose to have the installer put a shortcut to the MATLAB program on the Windows Start menu, select Start > MATLAB R2012b
  - On Windows XP systems, select Start > Programs > MATLAB > R2012b > MATLAB R2012b

- If you chose to have the installer create a shortcut, double-click the MATLAB shortcut on your Windows desktop.

- Double-click a file with any of a number of file extensions in the Windows Explorer tool. The installer sets up associations between these file types and MathWorks® products during installation. For example, double-clicking a file with a .m extension in the starts MATLAB and opens the file in the MATLAB Editor. For more information, see “Associating Files with MATLAB on Windows Platforms” on page 1-3.

After starting MATLAB, the desktop opens. Desktop components that were open when you last shut down MATLAB will be opened on startup. For more information, see “Adjust Desktop Appearance” on page 2-2. You can
specify other startup options, such the current folder upon startup—for more information, see “Startup Options” on page 1-18 and “Startup Folder for the MATLAB Program” on page 1-12.

If you have trouble starting MATLAB, see Troubleshooting topics in the Installation Guide.

**Associating Files with MATLAB on Windows Platforms**

When you install MATLAB software on Windows platforms, the installer sets up associations between certain file types and MathWorks products. When you double-click a particular file type, identified by its file extension, Windows starts MATLAB and opens the file in the appropriate tool. The following table lists some of the file extensions the installer associates with MathWorks products and the behavior that results from this association. To learn how to change this behavior, see “Managing File Associations for MATLAB on Windows Systems” on page 1-5.

**File Extension and Resulting Action**

<table>
<thead>
<tr>
<th>File Extension</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>.fig</td>
<td>Opens file in figure window</td>
</tr>
<tr>
<td>.m</td>
<td>Opens file in Editor</td>
</tr>
<tr>
<td>.mat</td>
<td>Opens Import Wizard to load the data into the MATLAB workspace.</td>
</tr>
<tr>
<td>.mdl</td>
<td>Opens file in a Simulink® model window</td>
</tr>
<tr>
<td>.mex¹</td>
<td>Displays icon for MATLAB in Windows Explorer tool</td>
</tr>
<tr>
<td>.p</td>
<td>Displays icon for MATLAB in Windows Explorer tool</td>
</tr>
</tbody>
</table>

File associations for the Windows Explorer tool do not affect what happens when you open one of these file types from within MATLAB. MATLAB acts on the file using the MATLAB tool associated with that file type. For example, even if your system associates .mat files with the Access™ application, when

---

¹ MEX-file extensions are platform specific. See “Using Binary MEX-Files”. 

---
you open a MAT-file from within MATLAB, it opens the Import Wizard to load the data.
Managing File Associations for MATLAB on Windows Systems.

You can associate any file types with MATLAB on computers running the Windows operating system. For example, you can associate the .xml extension with MATLAB so that when you double-click an XML file, it opens in the MATLAB Editor.

Occasionally, another program may already own the association with a particular file type. For example, the Microsoft Access program might own the association with files having a .mat extension. To reset this file association, use the Windows Default Programs control panel.

**Note** These instructions might not exactly apply to the version of the Windows operating system you are running on your computer. If you encounter differences or problems, see your Windows documentation.

1. Click the Windows Start menu.
2. Select Control Panel.
3. In the Control Panel window, select Default Programs.
4. In the Default Programs window, select Associate a file type or protocol with a program.
5. In the Set Associations window, select a file name extension to view which program currently opens it by default. In this example, select .mat.
6. To change the default association, click Change Program. This opens the Open with dialog box which lists other programs that might be recommended for this file extension. If it’s a file extension associated with MATLAB, such as .mat, the list includes all the versions of MATLAB you have installed. Select the version of MATLAB you want to associate with the file extension and click OK. You create file associations with particular versions of MATLAB.
7. Click Close to close the Set Associations dialog box.

After associating a file type with MATLAB, you can open other applications that have the same extension via the context menu. For example, if you want to open a MAT-file with the Access application, right-click myfile.mat, and
from the context menu, select **Open With**. The Access application should be one of the options.

**Speeding Up MATLAB Start Up on Windows Systems**

On Windows systems, the MathWorks Installer installs and configures a utility program that can speed-up MATLAB startup, called the MATLAB Startup Accelerator. For information about this program, including information about how to modify the configuration, see “Post-Installation Tasks” on the MathWorks Web site.

**Starting the MATLAB Program on Linux Platforms**

To start the MATLAB program on Linux® platforms, type `matlab` at the operating system prompt.

If you did not set up symbolic links in the installation procedure, you must enter the full path name, `matlabroot/bin/arch`, where `matlabroot` is the name of the folder in which you installed MATLAB and `arch` is an architecture-specific subfolder, such as `glnxa64`.

After starting MATLAB, the desktop opens. Desktop components that were open when you last shut down MATLAB will be opened on startup. For more information, see “Adjust Desktop Appearance” on page 2-2.

If the `DISPLAY` environment variable is not set or is invalid, the desktop will not display. If you have trouble starting MATLAB, see Troubleshooting topics in the Installation Guide.

You can specify the current folder upon startup as well as other options—for more information, see “Startup Folder for the MATLAB Program” on page 1-12 and “Startup Options” on page 1-18.

**Starting the MATLAB Program on Macintosh Platforms**

There are several ways to start the MATLAB program on Macintosh computers:

- Double-click the MATLAB icon in the Applications folder.
Starting and Exiting the MATLAB® Program

- Open a Terminal window, navigate to your MATLAB installation folder, and type `matlab` at the operating system prompt.

  `Applications/MATLAB_R2012b.app/bin/matlab`

After starting MATLAB, the desktop opens. Desktop components that were open when you last shut down MATLAB will be opened on startup. For more information, see “Adjust Desktop Appearance” on page 2-2. If the `DISPLAY` environment variable is not set or is invalid, the desktop will not display.

If MATLAB fails to start due to a problem with required system components such as Java™ software, diagnostics run automatically and advise you of the problem, along with suggestions to correct it.

If you have trouble starting MATLAB, see Troubleshooting topics in the Installation Guide.

You can specify the current folder upon startup as well as other options—for more information, see “Startup Folder for the MATLAB Program” on page 1-12 and “Startup Options” on page 1-18.

**Limitation**

On Macintosh platforms, if you run MATLAB remotely, for example using `rlogin`, you must run with `nodisplay`, `noawt`, and `nojvm` startup options—for more information, see “Startup Options” on page 1-18.

**Exiting the MATLAB Program**

- “Ways to Exit the MATLAB Program” on page 1-7
- “Confirm Exiting the MATLAB Program” on page 1-8
- “Running a Script When Exiting the MATLAB Program” on page 1-9
- “Abnormal Termination” on page 1-9

**Ways to Exit the MATLAB Program**

To exit the MATLAB program at any time, do one of the following:

- Click the Close box \[\times\] in the MATLAB desktop.
• Click 🔄 on the left side of the MATLAB desktop title bar and select Close.
• Type quit or exit at the Command Window prompt.

MATLAB closes after

• Prompting you to confirm exiting, if that preference is specified (see “Confirm Exiting the MATLAB Program” on page 1-8.)
• Prompting you to save any unsaved files
• Running the finish.m script, if it exists in the current folder or on the search path (see “Running a Script When Exiting the MATLAB Program” on page 1-9.)

Confirm Exiting the MATLAB Program
To set a preference that displays a confirmation dialog box when you exit MATLAB:

1 On the Home tab, in the Environment section, select Preferences.
2 Select General > Confirmation Dialogs.
3 Select the Confirm before exiting MATLAB check box and click OK.

MATLAB then displays the following dialog box when you exit.

For more information, see “Confirmation Dialogs Preferences” on page 2-57.

You can also display your own exit confirmation dialog box using a finish.m script, as described in the following section.
Running a Script When Exiting the MATLAB Program

When MATLAB exits, it runs the script `finish.m`, if `finish.m` exists in the current folder or anywhere on the search path. You create the file `finish.m`. It contains statements to run when MATLAB terminates, such as saving the workspace or displaying a confirmation dialog box. There are two sample files in `matlabroot/toolbox/local` that you can use as the basis for your own `finish.m` file:

- `finishsav.m` — Includes a `save` function so the workspace is saved to a MAT-file when MATLAB quits.
- `finishdlg.m` — Displays a confirmation dialog box that allows you to cancel quitting.

For more information, see the `finish` reference page.

Abnormal Termination

- “When the MATLAB Program Terminates Unexpectedly” on page 1-9
- “Error Log Reporting” on page 1-10
- “Recovering Data After an Abnormal Termination” on page 1-11

When the MATLAB Program Terminates Unexpectedly. In the event MATLAB experiences a segmentation violation (segv) or other serious problem, the MATLAB System Error dialog box opens to notify you about the problem. When this occurs, the internal state of MATLAB is unreliable and not suitable for further use. You should exit as soon as possible and then restart. However, you might want to first try to save your work in progress.

To exit and restart without trying to save your work, follow these steps:

1. If you want to view the stack trace for the problem, click Details.

2. Click Close to terminate MATLAB.

3. Restart MATLAB. If the Error Log Reporter dialog box opens, select the option to send a report to MathWorks.
To try to save your work in progress before exiting and restarting MATLAB, follow these steps:

1. If you want to view the stack trace for the problem, click **Details**.

2. Click **Attempt to Continue**. MATLAB tries to return to the Command Window or tool you were using.

   The Command Window displays the message **Please exit and restart MATLAB** to the left of the prompt, which reminds you to discontinue use.

3. From the Command Window or tool, try to save the workspace and unsaved files.

   **Caution** Because the internal state of MATLAB might be corrupted, do not save existing files to the same file name. Instead, specify a new file name. The information in the new file might be corrupted or incomplete.

4. Exit MATLAB immediately after saving because any further usage would be unreliable.

5. Restart MATLAB. If the Error Log Reporter dialog box opens, select the option to send a report to MathWorks.

**Error Log Reporting.** Upon startup, if MATLAB detects an error log generated by a serious problem during the *previous* session, an Error Log Reporter prompts you to email the log to MathWorks for analysis. The error log contains the stack trace and information about the MATLAB software configuration. If the problem occurs repeatedly, make note of what seems to cause it, look for information about it in the MathWorks Bug Reports database, and if the problem is reproducible, submit a Service Request via http://www.mathworks.com/support/contact_us/ts/help_request_1.html.

**Emailing Error Log Reports**

There are some situations where the Error Log Reporter will not open, for example, when you start MATLAB with a `-r` option or run in deployed mode.
It also will not open if you selected the option to never send error reports the last time the Error Log Reporter opened. If you experience abnormal termination but do not see the Error Log Reporter on subsequent startups, you can instead email the reports.

Send email to segv@mathworks.com with this file attached: C:\Temp\matlab_crash_dump.####. After you send the log file, delete it or move it to another location. If you do not delete the log file, the Error Log Reporter can detect it on the next startup and prompt you to send it, even though you already emailed it.

**Recovering Data After an Abnormal Termination.** If MATLAB terminates unexpectedly, you might lose information. After you start MATLAB again, you can try these suggestions to recover some of the information.

- Use the Command History or the file on which it is based, history.m, to run statements from the previous session. You might be able to approximately recreate data as it was prior to the termination.
- If you used the diary function or -logfile startup option for the session in which MATLAB terminated unexpectedly, you might be able to recover output.
- If you saved the workspace to a MAT-file during the session, you can recover it by loading the MAT-file.
- If you were editing a file in the Editor when MATLAB terminated unexpectedly, and you had the autosave enabled, you should be able to recover changes you made to files you had not saved.
- If you were in a Simulink session when a segmentation violation occurred, and you have the Simulink Autosave Options preference selected, note that the last autosave file for the model reflects the state of the autosave data prior to the segmentation violation. Because Simulink models might be corrupted by a segmentation violation, a model is not autosaved after a segmentation violation occurs.

Some of the above suggestions refer to actions you might have needed to take during the session when MATLAB terminated. If you did not take those actions, consider regularly performing them to help you recover from any future abnormal terminations you might experience.
Startup Folder for the MATLAB Program

In this section...

- “What Is the Startup Folder?” on page 1-12
- “Startup Folder on Windows Platforms” on page 1-13
- “Startup Folder on Linux Platforms” on page 1-14
- “Startup Folder on Macintosh Platforms” on page 1-14
- “Changing the Startup Folder” on page 1-14

What Is the Startup Folder?

The startup folder is the current folder in the MATLAB application when it starts. It is convenient if you make the current folder upon startup be a folder that you frequently use. On Windows and Apple Macintosh platforms, a folder called userpath is added automatically to the search path upon startup, and is the default startup folder. On Linux platforms, you can set the userpath as the startup folder. The default value for userpath is, for example, Documents/MATLAB on Microsoft Windows Vista™ platforms. You can specify a different default value for userpath, or specify a different startup folder.

Accepting the default value for userpath and using it as the startup folder offers these benefits:

- You can store the MATLAB files you work with in one, appropriately-named location, such as Documents/MATLAB.
- Your MATLAB files are readily available upon startup, because the current folder is always the same, for example, Documents/MATLAB.
- You can always run your files because MATLAB automatically adds the userpath folder to the top of the search path upon startup.
- The first time you run a new version of MATLAB, MATLAB automatically creates the userpath folder if it does not exist.
- When you upgrade to a newer version of MATLAB, MATLAB automatically continues to use the same MATLAB folder and your existing files, with all of its other benefits.
• The default userpath also utilizes the benefits provided by the standard location in the Windows and Macintosh environments for storing personal files. Files in the Documents/MATLAB folder (or My Documents/MATLAB on Windows platforms other than Windows Vista) are available to you when you use other machines. Because each user has their own Documents/MATLAB folder, other users, even those using your machine, cannot access files in your Documents/MATLAB folder.

To view the userpath value, run the userpath function. To specify a location other than the default for userpath, or if you do not want to take advantage of userpath, make changes with the userpath function.

There are other ways to change the startup folder as well as the folders on your search path. For more information, see “Changing the Startup Folder” on page 1-14 and “Viewing Files and Folders on the Search Path” on page 6-73.

### Startup Folder on Windows Platforms

The startup folder on Windows platforms depends on any startup options you specified and the way you started MATLAB:

<table>
<thead>
<tr>
<th>How Started</th>
<th>Startup Folder</th>
</tr>
</thead>
<tbody>
<tr>
<td>Double-click the MATLAB shortcut on your Windows desktop</td>
<td>The startup folder is set to the userpath value, whose default value is My Documents\MATLAB, or Documents\MATLAB on Windows Vista platforms. The userpath folder is automatically added to the search path. If there is a value specified in the Start in field of the Properties dialog box for the MATLAB program, that value is the startup folder, although the userpath is added to the search path. If MATLAB does not find a valid userpath value upon startup, and the</td>
</tr>
</tbody>
</table>
## Startup and Shutdown

### How Started

<table>
<thead>
<tr>
<th>How Started</th>
<th>Startup Folder</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start in field is empty, the startup folder is</td>
<td>Start in field is empty, the startup folder is the Windows desktop.</td>
</tr>
<tr>
<td>the Windows desktop.</td>
<td></td>
</tr>
<tr>
<td>Double-click a file type associated with MATLAB</td>
<td>The folder in which the file resides is the startup folder. The userpath folder is automatically added to the search path.</td>
</tr>
<tr>
<td>In a DOS window</td>
<td>The folder in which you ran the <code>matlab</code> command is the startup folder. The userpath folder is automatically added to the search path.</td>
</tr>
</tbody>
</table>

### Startup Folder on Linux Platforms

On Linux platforms, the default startup folder is the folder from which you started MATLAB.

You can specify that the `userpath` be the startup folder by setting the value of the environment variable `MATLAB_USE_USERPATH` to 1 prior to startup. By default, `userpath` is `userhome/Documents/MATLAB`, and MATLAB automatically adds the userpath folder to the top of the search path upon startup. To specify a different folder for `userpath`, and for other options, use the MATLAB `userpath` function.

### Startup Folder on Macintosh Platforms

When you start MATLAB on Apple Macintosh platforms by double-clicking the MATLAB application, the startup folder is the value returned when you enter `userpath`, which by default is `userhome/Documents/MATLAB`. MATLAB automatically adds the userpath folder to the top of its search path upon startup. To specify a different folder for `userpath`, and for other options, use the `userpath` function.

When you start MATLAB in a shell, the default startup folder is the folder from which you started MATLAB.

### Changing the Startup Folder

You can start MATLAB in a folder other than the default in one of these ways:
Changing the Startup Folder Via the `userpath` Function

Use the `userpath` function to change the startup folder as well as to add the startup folder to the search path upon startup. For more information, see the `userpath` reference page and “Startup Folder for the MATLAB Program” on page 1-12.

Changing the Startup Folder Using the Shortcut — Windows Platforms Only

To change the startup folder on Windows platforms using the shortcut,

1. Right-click the shortcut icon for MATLAB and select Properties from the context menu.

   The Properties dialog box for MATLAB opens to the Shortcut pane.

2. The Target field contains the full path to start MATLAB.

   By default, the startup folder is My Documents\MATLAB or Documents\MATLAB on Windows Vista platforms; for more information, see “Startup Folder on Windows Platforms” on page 1-13.

   In the Start in field, specify the full path to the folder in which you want MATLAB to start, and click OK.
The next time you start MATLAB using that shortcut icon, the current folder will be the one you specified in step 2.
You can make multiple shortcuts to start MATLAB, each with its own startup folder, and with each startup folder having different startup options.

**Changing the Startup Folder Using the `startup.m` File**

Use the `startup.m` file to specify the startup folder as well as other startup options—for details, see “Specifying Startup Options in the MATLAB Startup File” on page 1-19.
Startup Options

In this section...

“Specifying MATLAB Startup Options” on page 1-18
“Commonly Used Startup Options” on page 1-20
“Passing Perl Variables on Startup” on page 1-21
“Startup and Calling Java Software from the MATLAB Program” on page 1-22

Specifying MATLAB Startup Options

You can specify startup options (also called command flags or command line switches) that instruct the MATLAB program to perform certain operations when you start it. On all platforms, you specify the options as arguments to the `matlab` command when you start is at the operating system prompt. For example, the following starts MATLAB and suppresses the display of the splash screen.

```
matlab -nosplash
```

On Windows platforms, you can precede a startup option with either a hyphen (-) or a slash (/). For example, `-nosplash` and `/nosplash` are equivalent.

On all platforms, you can also specify startup options using a MATLAB startup file—see “Specifying Startup Options in the MATLAB Startup File” on page 1-19

On Windows platforms, you can specify startup options in the MATLAB shortcut—see “Including Startup Options in a Shortcut on Windows Systems” on page 1-18.

Including Startup Options in a Shortcut on Windows Systems

You can add selected startup options (also called command flags or switches for the command line) to the target path for your shortcut in the Windows environment for MATLAB. For more information about the options, see “Commonly Used Startup Options” on page 1-20.
To use startup options for the MATLAB shortcut icon in a Windows environment, follow these steps:

1. Right-click the shortcut icon for MATLAB and select **Properties** from the context menu. The Properties dialog box for MATLAB opens to the **Shortcut** pane.

2. In the **Target** field, after the target path for `matlab.exe`, add the startup option, and click **OK**. For example, adding `-r "filename"` runs the MATLAB code file `filename` after startup.

This example instructs MATLAB to automatically run the file `results` after startup, where `results.m` is in the startup folder or on the search path for MATLAB. The statement in the **Target** field might appear as

C:\Program Files\MATLAB\R2010b\bin\matlab.exe -r "results"

Include the statement in double quotation marks ("statement"). Use only the file name, not the file extension or path name. For example, MATLAB produces an error when you run

... matlab.exe -r "D:\results.m"

Use semicolons or commas to separate multiple statements. This example changes the format to **short**, and then runs the MATLAB code file `results`:

... matlab.exe -r "format('short');results"

Separate multiple options with spaces. This example starts MATLAB without displaying the splash screen, and then runs the MATLAB code file `results`:

... matlab.exe -nosplash -r "results"

**Specifying Startup Options in the MATLAB Startup File**

At startup, MATLAB automatically executes the file `matlabrc.m` and, if it exists, `startup.m`. The file `matlabrc.m`, which is in the `matlabroot/toolbox/local` folder, is reserved for use by MathWorks and by the system manager on multiuser systems.
The file `startup.m` is for you to specify startup options. For example, you can modify the default search path, predefine variables in your workspace, or define defaults for Handle Graphics® objects. Use the following statements in a `startup.m` file to add the specified folder, `/home/username/mytools`, to the search path, and to change the current folder to `mytools` upon startup.

```
addpath /home/username/mytools
cd /home/username/mytools
```

Place the `startup.m` file in the default or current startup folder, which is where MATLAB first looks for it. For more information, see “Startup Folder for the MATLAB Program” on page 1-12.

### Commonly Used Startup Options

The following table provides a list of some commonly used startup options for both Windows and UNIX® platforms. For more information, including a complete list of startup options, see the `matlab` (Windows) reference page or the `matlab` (UNIX) reference page.

<table>
<thead>
<tr>
<th>Platform</th>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>All</td>
<td><code>-c licensefile</code></td>
<td>Set <code>LM_LICENSE_FILE</code> to <code>licensefile</code>. It can have the form <code>port@host</code>.</td>
</tr>
<tr>
<td>All</td>
<td><code>-h</code> or <code>-help</code></td>
<td>Display startup options (without starting MATLAB).</td>
</tr>
<tr>
<td>All</td>
<td><code>-logfile &quot;logfilename&quot;</code></td>
<td>Automatically write output from MATLAB to the specified log file.</td>
</tr>
<tr>
<td>Windows platforms</td>
<td><code>-minimize</code></td>
<td>Start MATLAB with the desktop minimized. Any desktop tools or documents that were undocked when MATLAB was last closed will not be minimized upon startup.</td>
</tr>
</tbody>
</table>
### Startup Options

<table>
<thead>
<tr>
<th>Platform</th>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNIX platforms</td>
<td>-nojvm</td>
<td>Start MATLAB without loading the Sun Microsystems™ JVM™ software. This minimizes memory usage and improves initial startup speed, but restricts functionality. With nojvm, you cannot use the desktop, figures, or any tools that require Java software. For example, you cannot set preferences if you start MATLAB with the -nojvm option. However, you can start MATLAB once without the -nojvm option, set the preference, and quit MATLAB. MATLAB remembers that preference when you start it again, even if you use the -nojvm option.</td>
</tr>
<tr>
<td>All</td>
<td>-nosplash</td>
<td>Start MATLAB without displaying its splash screen.</td>
</tr>
<tr>
<td>All</td>
<td>-r &quot;statement&quot;</td>
<td>Automatically run the specified statement immediately after MATLAB starts. This is sometimes referred to as calling MATLAB in batch mode. Files you run must be in the startup folder for MATLAB or on the search path. Do not include path names or file extensions. Enclose the statement in double quotation marks (&quot;statement&quot;). Use semicolons or commas to separate multiple statements.</td>
</tr>
<tr>
<td>All</td>
<td>-singleCompThread</td>
<td>Limit MATLAB to a single computational thread. By default, Windows makes use of the multithreading capabilities of the computer on which it is running.</td>
</tr>
</tbody>
</table>

### Passing Perl Variables on Startup

You can pass Perl variables to MATLAB on startup by using the -r option of the `matlab` function. For example, assume a MATLAB function `test` that takes one input variable:

```matlab
function test(x)
    % code here
end
```

To start MATLAB with the function `test`, use the command

```
matlab -r "test(10)"
```

On some platforms, you might need to use double quotation marks:

```
matlab -r "test(10)"
```
This command starts MATLAB and runs test with the input argument 10.

To pass a Perl variable instead of a constant as the input parameter, follow these steps.

1. Create a Perl script such as

```bash
#!/usr/local/bin/perl
$val = 10;
system('matlab -r "test(' . ${val} . ')"');
```

2. Invoke the Perl script at the prompt using a Perl interpreter.

For more information, see the `matlab` (Windows) or `matlab` (UNIX) reference page.

**Startup and Calling Java Software from the MATLAB Program**

When the MATLAB program starts, it constructs the class path for Java software using `librarypath.txt` as well as `classpath.txt`. If you call Java software from MATLAB, see more about this in “The Java Class Path” and “Locating Native Method Libraries” in the MATLAB External Interfaces documentation.
Toolbox Path Caching in the MATLAB Program

In this section...

“About Toolbox Path Caching in the MATLAB Program” on page 1-23
“Using the Cache File Upon Startup” on page 1-23
“Updating the Cache and Cache File” on page 1-23
“Additional Diagnostics with Toolbox Path Caching” on page 1-26

About Toolbox Path Caching in the MATLAB Program
For performance reasons, the MATLAB program caches toolbox folder information across sessions. The caching features are mostly transparent to you. However, if MATLAB does not see the latest versions of your MATLAB code files or if you receive warnings about the toolbox path cache, you might need to update the cache.

Using the Cache File Upon Startup
Upon startup, MATLAB gets information from a cache file to build the toolbox folder cache. Because of the cache file, startup is faster, especially if you run MATLAB from a network server or if you have many toolbox folders. When you end a session, MATLAB updates the cache file.

MATLAB does not use the cache file at startup if you clear the Enable toolbox path cache check box in File > Preferences > General. Instead, it creates the cache by reading from the operating system folders, which is slower than using the cache file.

Updating the Cache and Cache File

How the Toolbox Path Cache Works
MATLAB caches (essentially, stores in a known files list) the names and locations of files in \texttt{matlabroot/toolbox} folders. These folders are for files provided with MathWorks products that should not change except for product installations and updates. Caching those folders provides better performance during a session because MATLAB does not actively monitor those folders.
We strongly recommend that you save any MATLAB code files you create and any files provided by MathWorks that you edit in a folder that is not in the `matlabroot/toolbox` folder tree. If you keep your files in `matlabroot/toolbox` folders, they may be overwritten when you install a new version of MATLAB.

**When to Update the Cache**

When you add files to `matlabroot/toolbox` folders, the cache and the cache file need to be updated. MATLAB updates the cache and cache file automatically when you install toolboxes or toolbox updates using the installer for MATLAB. MATLAB also updates the cache and cache file automatically when you use MATLAB tools, such as when you save files from the MATLAB Editor to `matlabroot/toolbox` folders.

When you add or remove files in `matlabroot/toolbox` folders by some other means, MATLAB might not recognize those changes. For example, when you

- Save new files in `matlabroot/toolbox` folders using an external editor
- Use operating system features and commands to add or remove files in `matlabroot/toolbox` folders

MATLAB displays this message:

Undefined function or variable

You need to update the cache so MATLAB will recognize the changes you made in `matlabroot/toolbox` folders.

**Steps to Update the Cache**

To update the cache and the cache file,

1. Select **File > Preferences > General**.

   The **General Preferences** pane is displayed.

2. Click **Update Toolbox Path Cache** and click **OK**.
Function Alternative
To update the cache, use `rehash toolbox`. To also update the cache file, use `rehash toolboxcache`. For more information, see `rehash`.
Additional Diagnostics with Toolbox Path Caching
To display information about startup time when you start MATLAB, select the Enable toolbox path cache diagnostics check box in General Preferences.
Desktop

- “Adjust Desktop Appearance” on page 2-2
- “Define Keyboard Shortcuts” on page 2-15
- “Set Print Options” on page 2-39
- “Web Browsers and MATLAB” on page 2-42
- “License Management and Software Updates” on page 2-47
- “Macintosh Platform Conventions” on page 2-50
- “Preferences” on page 2-52
Adjust Desktop Appearance

<table>
<thead>
<tr>
<th>In this section...</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Fonts” on page 2-2</td>
</tr>
<tr>
<td>“Color Settings” on page 2-7</td>
</tr>
<tr>
<td>“Access Frequently Used Features” on page 2-10</td>
</tr>
<tr>
<td>“Optimize Desktop Layout for Limited Screen Space” on page 2-11</td>
</tr>
</tbody>
</table>

Fonts
You can specify the font characteristics (type, style, and size) for:

- Sets of MATLAB desktop tools, based on their predominant content — code or text
- Each desktop tool individually
- A combination of sets and individual tools

You can set some font options differently for printing — see “Set Print Options” on page 2-39.

Setting Fonts for Sets of Tools Based on Content
You can specify fonts for tools based on their predominant content — text or code. For an example of how fonts affect text in the Editor, see “Monospace Font Compared to Proportional Font” on page 2-4.

To set fonts for a tool:


2. Under Desktop code font select a font type, style, and size from the drop-down menus.

   For font size, you can type a size, including a size not shown as a choice in the drop-down menu.
3 Under **Desktop text font**, do one of the following:
   - Select **Use system font**.
   - Select the font type, style, and size from the drop-down menus.
     For font size, you can type a size, including a size not shown as a choice in the drop-down menu.

4 Review the list of tools currently using each font group—**Desktop code font**, **Desktop text font**, and **Custom fonts**.

   If you want to move a tool from one group to another, or if you want to customize fonts for a particular tool, follow the steps in “Setting Fonts for Individual Tools” on page 2-3. Otherwise, click **OK**.

---

**Note** If you are running MATLAB on a Linux\(^2\) or UNIX\(^3\) platform, consider selecting **Use antialiasing to smooth desktop fonts**. Note however, that this option requires you to restart MATLAB.

---

**Setting Fonts for Individual Tools**

Custom font preferences enable you to change the font for individual desktop tools. For an example of how fonts affect text in the Editor, see “Monospace Font Compared to Proportional Font” on page 2-4.

To set custom fonts:

1 On the **Home** tab, in the **Environment** section, click **Preferences > Fonts > Custom**.

2 Select the tool you want to customize from the **Desktop tools** list.

   The type of font the tool currently uses appears under **Font to Use**.

3 Under **Font to Use**, select one of the following:

---

2. Linux is a registered trademark of Linus Torvalds.

3. UNIX is a registered trademark of The Open Group in the United States and other countries.
• **Desktop code.**
• **Desktop text**
• **Custom** (Then, specify the font characteristics.)

4 Click **OK**.

The font that you select for the Profiler and Comparison Tool also applies to:

• Function Browser function help
• Extended Code Analyzer messages
• The content pane of the Supplemental Software help browser

For the Profiler and associated tools, you can change the font size, but not the style (for example, bold or italic).

**Monospace Font Compared to Proportional Font.** A monospace font is useful when you care about alignment, but it does take more space than a proportional font.

With a monospaced font, all characters are the same width.

```
1234567890
abcdefghij
ABCDEFghij
```

With a proportional font, characters are different widths.

```
1234567890
abcdefghij
ABCDEFghij
```

**Factory Default Font Settings**
This table lists the factory default code and text font settings, and the tools that use those font settings. If you previously changed settings, and now want to revert to the default values, update the settings using the values in the table.
<table>
<thead>
<tr>
<th><strong>Font Type</strong></th>
<th><strong>Factory Default Characteristics</strong></th>
<th><strong>Tools Using Font Type by Factory Default</strong></th>
</tr>
</thead>
</table>
| Desktop code font          | Monospaced, Plain, 10 point         | • Command History  
                              • Command Window  
                              • Editor (which also applies to the Shortcuts Editor) |
| Desktop text font          | Your system’s current font.         | • Current Folder browser (which also applies to the Path browser)  
                              • Workspace browser  
                              • Variables editor  
                              • Function Browser in the Command Window and Editor |
| Custom fonts               | SansSerif, Plain, 10 point          | • Profiler and Comparison Tool                                                  |

**Changing the Font**
This example changes the settings for the desktop code font; changes the Command History font preference so that it uses the desktop text font instead of the code font; and specifies a custom font for the Current Folder browser.

1. On the **Home** tab, in the **Environment** section, click **Preferences > Fonts**.
2. Under **Desktop code font**, select **Times New Roman, Plain, 14 point**.
3. Under **Desktop text font**, select **Use system font**.
4. Click **Apply**.
5. Make the Command History window use the desktop text font:
   a. Click the **Custom Fonts** link.
   b. Under **Desktop tools**, select **Command History**.
   c. Under **Font to use**, select the **Desktop text** radio button.
   d. Click **Apply**.
6. Apply a custom font to the Current Folder browser:
a Under **Desktop tools**, select **Current Folder**.
b Under **Font to use**, select the **Custom** radio button.
c Select Arial Narrow and **Plain**, and then type 11 in the size field.
d Click **OK**.

The following table details the results of the changes.

<table>
<thead>
<tr>
<th><strong>Font Type</strong></th>
<th><strong>Font Characteristics</strong></th>
<th><strong>Tools Assigned Font</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Desktop code</td>
<td>Times New Roman® font, Plain, 14 point</td>
<td>Command Window Editor</td>
</tr>
<tr>
<td>Desktop text</td>
<td>Same as your current system font, which appears in the dimmed fields below the <strong>Use system font</strong> check box.</td>
<td>Command History Workspace browser Variables editor Function Browser</td>
</tr>
<tr>
<td>Custom</td>
<td>Monotype Corporation Arial® Narrow font, Plain, 11 point</td>
<td>Current Folder browser</td>
</tr>
</tbody>
</table>

**Font Troubleshooting — Document Displays Boxes and Odd Symbols Instead of Text**

If you open a file created by someone else, and you see boxes or meaningless symbols instead of text, it is probably because you are using different language fonts from the file creator. This situation can occur, for example, if you open a file created by someone whose native language is Japanese and your native language is English. The Japanese user is probably using fonts for East Asian languages and you are not.

To resolve this problem, in the Windows Control Panel, find the region and language options, and then install the supplemental files for East Asian languages.

For more information, refer to the Windows help.
Where MATLAB Gets Fonts

Unless you change it, MATLAB gets fonts from the operating-system font list. Any fonts on the system list that MATLAB cannot display are excluded from its list. The system fonts are installed in one of the following locations:

- The operating system’s standard location
  
  Ask your system administrator where this is on your system.

- The /jre/lib/fonts folder where Java software is installed on your system.

If a new compatible font becomes available to MATLAB, consider updating fonts on your Windows platform, as follows:

1. Use the Windows Control Panel to install the font.
   
   For more information, refer to the Windows help.

2. Restart MATLAB.

A compatible font on Windows platforms for desktop components (such as the Command Window), figure windows, and uicontrols is one compatible with TrueType and Microsoft OpenType® fonts. A bitmapped font is compatible font for graphics objects, such as xlabel, ylabel, title, and text.

Color Settings

You can change any of the colors that the desktop uses for text, background, or denoting links and different types of case-specific highlighting.

Changing Text, Background, and Hyperlink Colors in Desktop Tools

To change the colors that MATLAB uses for text and background in desktop tools follow these steps:

Note  The colors you specify also apply to the Import Wizard, but do not apply to the Help display pane or the Web browser.
1 On the **Home** tab, in the **Environment** section, click **Preferences > Colors**.

2 **Clear Use system colors.**

   System colors are the text and background colors that your platform (for example, Microsoft Windows) uses for other applications.

3 **Select the colors you want to use from the Text and Background color palettes.**

   When you choose a color, the **Sample** area in the dialog box updates to show you how it looks.

   **Tip** If you use a gray background color, a selection in an inactive window is not visible.

4 **Under Other colors**, select the color you want to use for hyperlinks.

5 **Click OK.**

---

**Changing Syntax Highlighting Colors**

In the Command Window, Command History, Editor, and Shortcuts callback area, MATLAB conveys syntax information using different colors. This feature, known as syntax highlighting, helps you to identify syntax elements, such as `if/else` statements at a glance. The Editor also provides syntax highlights colors for other languages.

In the Command Window, only the MATLAB input you type is highlighted. The output from running MATLAB functions is not highlighted.

To change syntax highlighting colors, follow these steps:

1 On the **Home** tab, in the **Environment** section, click **Preferences > Editor/Debugger > Language**.

2 From the **Language** drop-down menu, select the language for which you want to change syntax highlighting colors.
3 In the **Syntax highlighting** section, select **Enable syntax highlighting**.

4 Change the colors.
   - If you set the **Language** to MATLAB, click the **Set syntax colors** link, and then change the colors under **MATLAB syntax highlighting colors**.
   - If you did not set the **Language** to MATLAB, change the colors under **Syntax highlighting**.

5 Click OK.

**Changing Code Analyzer Colors**

Code Analyzer helps you to identify potential problems and refine your MATLAB code. By default, the Editor indicates:

- Code for which there are warnings, by underlining that code with an orange wavy line and placing an orange line in the message bar.
- Code for which there are errors, by underlining that code with an red wavy line and placing a red line in the message bar.
- Code that MATLAB can fix automatically (autofix), by highlighting that code in tan.

To change code analyzer colors:

1 On the **Home** tab, in the **Environment** section, click **Preferences > Colors > Programming Tools**.

2 Under **Code analyzer colors**, select the colors you want for warnings, autofix highlighting, or both.

3 Decide if you want autofix highlights to appear in the Editor.
   - Clear **Autofix highlight** if you do not want autofix highlights to appear in the Editor; select **Autofix highlight** if you do.

4 Click **Apply**.

5 Decide if you want to change the color that the code analyzer uses for errors.
   - If you do not, go to step 6.
• If you do, then:
  a In the left navigation pane, click Colors.
  b Under MATLAB syntax highlighting colors, change the color for Errors.

    In addition to changing the color of code analyzer indicators for errors, this action also changes the color for errors in the Command Window, Command History window, Editor, and Shortcuts callback area.

6 Click OK.

For more information, see “Automatically Check Code in the Editor — Code Analyzer”.

Access Frequently Used Features

The quick access toolbar provide access to frequently used operations. This toolbar is always visible, even when you navigate between different MATLAB Toolstrip tabs.

You can change the location of the quick access toolbar. On the Home tab, in the Environment section, click Layout, and then select an option for the Quick Access Toolbar.

To add a Toolstrip button to the quick access toolbar, right-click the button, and then select Add to Quick Access Toolbar.

To add, remove, or rearrange buttons on the quick access toolbar, follow these steps:

1 On the Home tab, in the Environment section, click Preferences > Toolbars.

2 From the Toolbar drop-down menu, select Quick Access.

    The controls for the selected toolbar appear in the Layout and Controls sections of the Toolbars Preferences pane.
3 In the Controls list, select or clear the check box for controls that you want to display or remove from the toolbar, respectively.

4 Under Layout, rearrange the order of the controls and separator bars on the selected toolbar, by doing either of the following:
   - Drag the icon for a control or separator bar to another position.
   - Select a Layout icon, and then click one of the Layout buttons below the layout icons.

   For instance, to move the MATLAB desktop cut icon to the beginning of the toolbar, select the Cut icon, and then click.

5 Click Apply or OK.

Optimize Desktop Layout for Limited Screen Space
You can close, minimize, and undock desktop tools to optimize the desktop layout. Once you design a layout you like, you can save it for reuse.

- “Desktop Layout” on page 2-11
- “Document Layout” on page 2-13

Desktop Layout
This table shows how you can optimize the MATLAB desktop layout on your screen.
<table>
<thead>
<tr>
<th><strong>Action</strong></th>
<th><strong>Procedure</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimize or restore the MATLAB Toolstrip</td>
<td>At the upper right corner of the Toolstrip, click ▲ or ▼.</td>
</tr>
<tr>
<td>Open or hide a tool</td>
<td>On the <strong>Home</strong> tab, in the <strong>Environment</strong> section, click <strong>Layout</strong>. Then, under <strong>Show</strong>, select or deselect desktop tools you want to show or hide. You also can use a function to open desktop tools. For example, to open the Editor, use <code>edit</code>. To open the Profiler, use <code>profile</code> with the <code>viewer</code> option.</td>
</tr>
<tr>
<td>Maximize a tool</td>
<td>Do one of the following:</td>
</tr>
<tr>
<td></td>
<td>• Double-click the title bar in that tool.</td>
</tr>
<tr>
<td></td>
<td>• On the title bar of a docked tool, click ◀, and then select <strong>Maximize</strong>.</td>
</tr>
<tr>
<td>Minimize a tool</td>
<td>On the title bar of a docked tool, click ◀, and then select <strong>Minimize</strong>.</td>
</tr>
<tr>
<td>Use a minimized tool</td>
<td>Click the button for the tool to temporarily open the tool on the desktop. When you</td>
</tr>
<tr>
<td>Restore a tool as it appeared before maximizing or</td>
<td>Do one of the following:</td>
</tr>
<tr>
<td>Move a tool</td>
<td>Drag a tool by its title bar to a new location. The status bar indicates where the tool</td>
</tr>
<tr>
<td>Close a tool</td>
<td>On the title bar of a docked tool, click ◀, and then select <strong>Close</strong>.</td>
</tr>
<tr>
<td>Show or hide title bars</td>
<td>On the <strong>Home</strong> tab, in the <strong>Environment</strong> section, click <strong>Layout</strong>. Then, under <strong>Show</strong>,</td>
</tr>
<tr>
<td>Show or hide a toolbar in a figure window</td>
<td>From the <strong>View</strong> menu, select the toolbar of interest.</td>
</tr>
<tr>
<td>Undock tools to move them outside the desktop</td>
<td>Do one of the following:</td>
</tr>
<tr>
<td></td>
<td>• Drag a tool by its title bar to a new location outside of the MATLAB desktop.</td>
</tr>
<tr>
<td></td>
<td>• On the title bar of the tool, click ◀, and then select <strong>Undock</strong>.</td>
</tr>
</tbody>
</table>
Adjust Desktop Appearance

**Move undocked tools back to the desktop**
At the upper right of the tool panel, click 📦, and then select **Dock**.

**Manage a desktop arrangement**
On the **Home** tab, in the **Environment** section, click **Layout**, and then select an option.

When you end a session, MATLAB saves the current desktop arrangement. The next time you start MATLAB, the desktop appears as you left it. However, tools such as the Help browser, Web browser, and Variables editor do not reopen automatically, even if they were open when you ended the last session. You can use startup options to specify tools that you want to open on startup. For more information, see “Startup Options” on page 1-13.

**Document Layout**
When you open MATLAB documents, they open in the associated tool, such as the Editor or Variables editor, and a document bar lists all the open documents. The Editor and Variables editor appear in the position they occupied when last used. Entries for undocked documents appear on the Windows task bar, or the equivalent for your platform. Click the task bar entry for a document to make that document active.

This table shows how to optimize the layout of documents and document bar within a tool.

<table>
<thead>
<tr>
<th>Action</th>
<th>Procedure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Move or hide document bar</td>
<td>On the <strong>View</strong> tab, in the <strong>Document Bar</strong> section, click <strong>Bar Position</strong>, and then select an option.</td>
</tr>
<tr>
<td>Reorder document names on the document bar</td>
<td>Drag a document name to a different position on the document bar. To alphabetize names of documents on the document bar, in the <strong>Document Bar</strong> section of the <strong>View</strong> tab, select <strong>Alphabetize</strong>.</td>
</tr>
<tr>
<td>Action</td>
<td>Procedure</td>
</tr>
<tr>
<td>------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| Arrange or tile documents          | In the Editor and Variables editor, select the View tab. In the Tiles section, click a tile option.  
In a Figure panel, Help browser, or Web browser, select a tile option, ☐, ☐, ☐, or ☐, on the right side of the toolbar. |
| Move a tiled document             | Drag the title bar of the document to another tile. If you drag it to a tile that already contains a document, the document you are dragging covers up the other document. |
| Undock a document                 | Right-click the document name in the document bar, and then select Undock.                                                                  |
| Close and save the document       | Click ❌.                                                                                                                                  |
| currently displaying              |                                                                                                                                 |
| Close a document in the Editor    | Click Ctrl + ❌.                                                                                                                           |
| without saving                    |                                                                                                                                 |
Define Keyboard Shortcuts

Keyboard Shortcuts

To quickly access desktop features, you can use keyboard shortcuts. Press \texttt{Alt} to display tooltips on MATLAB Toolstrip buttons, indicating what keys to press to access those features. For example, pressing \texttt{Alt} followed by \texttt{H} accesses the \texttt{Home} tab and displays tooltips for the features available on that tab. You cannot customize these shortcuts.

An action can have multiple keyboard shortcuts. All defined shortcuts work, but only one appears on the desktop Toolstrip tooltip.

You can:

- Choose from a set of shortcuts that install with MATLAB.
- Create customized sets of shortcuts.
- Use a set of shortcuts copied from another system.
Choose a Set of Keyboard Shortcuts

By default, MATLAB uses keyboard shortcut settings that correspond to the platform on which you are running. To choose different keyboard shortcut settings, follow these steps:

1 On the **Home** tab, in the **Environment** section, click **Preferences > Keyboard > Shortcuts**.

2 Click the down arrow in the **Active settings** field, and make a selection from the drop-down list, as summarized in this table.

<table>
<thead>
<tr>
<th>Settings File</th>
<th>Option to Select</th>
<th>Details</th>
</tr>
</thead>
</table>
| Installed with MATLAB | • On Mac, Mac Default Set  
                        • On all other systems, Windows Default Set or Emacs Default Set | For a description of the files that install with MATLAB, see “Installed Settings Files for Keyboard Shortcuts” on page 2-17. |
| Previously added | The file name | No additional information. |
| On your system, but not in the drop-down list | Browse | “Browse to Keyboard Shortcuts Settings Files” on page 2-17. |

3 Click **Apply**.
Installed Settings Files for Keyboard Shortcuts

The following table lists the keyboard shortcuts settings files installed with MATLAB.

<table>
<thead>
<tr>
<th>Operating System</th>
<th>Keyboard Shortcut Settings Files Installed with MATLAB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows</td>
<td>• Windows Default Set (Default)</td>
</tr>
<tr>
<td></td>
<td>• Emacs Default Set</td>
</tr>
<tr>
<td>UNIX</td>
<td>• Emacs Default Set (Default)</td>
</tr>
<tr>
<td></td>
<td>• Windows Default Set</td>
</tr>
<tr>
<td>Macintosh</td>
<td>• Macintosh Default Set (Default)</td>
</tr>
</tbody>
</table>

Browse to Keyboard Shortcuts Settings Files

Browse to use a keyboard shortcuts settings file that is on your system, but not an Active settings choice in the Keyboard Shortcuts Preferences dialog box. This situation typically arises when you copy a settings file from another system to a folder other than the `prefdir` directory. To browse to a settings file and make it your active settings file, follow these steps:

2. In the Active settings field, click the down arrow, and then select Browse.
3. In the Open dialog box, navigate to the folder containing the settings file.
4. Select the settings file, and then click Open.
5. In the Keyboard Shortcuts preferences pane, click OK.

The settings file you selected in step 4 is now the active settings file for MATLAB.

Future MATLAB sessions will provide this settings file as a choice in the Active settings drop-down menu.
Download Keyboard Shortcut Settings Files from File Exchange

Download keyboard shortcut settings files from File Exchange when you want to do either of the following:

- Restore the MATLAB default keyboard shortcuts that were in place for MATLAB Version 7.9 (R2009a) and earlier releases.
- Find and download keyboard shortcuts that others created and uploaded to File Exchange.

Follow these steps:


2. In the Active settings field, click the down arrow, and then select Search File Exchange for Downloadable Shortcut Sets.

   MATLAB opens File Exchange.

3. Enter your MathWorks account information, and then click Submit.

   If you do not have a MathWorks account, create one by clicking Create an account, and then completing the form that opens in your Web browser.

4. Search File Exchange for the keyboard shortcut set that you want to use.

   When you follow the steps presented so far, File Exchange lists all files tagged with keyboard shortcuts configurable including:
   - MATLAB Desktop R2009a Non-Default Keyboard Shortcut sets
   - MATLAB Desktop R2009a Default Keyboard Shortcut sets

   For a description, click the file name in the File Summary column of File Exchange. Click the Back button to return to the list of files.

5. Click the download button next to the file you want to download.

6. Respond to the Download MATLAB Desktop confirmation dialog boxes as follows:
   - Click Download.
b Click **Change Current Folder to Download Location**.

The downloaded .ZIP file appears in the Current Folder Browser. Expand it to preview its contents.

7 In the Current Folder browser, right-click the downloaded .ZIP file, and then select **Extract**.

MATLAB creates a subfolder with the same name as the .ZIP file and extracts the files from that .ZIP file into the newly created folder.

8 In the Current Folder browser, expand the newly created folder, and then double-click the settings file you want to use.

A keyboard key icon preceding a file name indicates a valid keyboard shortcut settings file.

9 In the Keyboard Shortcuts Preferences dialog box, review the settings, and then click **OK**.

The newly downloaded settings file is now in effect.

**Compare Sets of Keyboard Shortcuts**

Compare sets of keyboard shortcuts to:

- Upgrade MATLAB from a version before Version 7.9 (R2009b).
  
  MATLAB 7.9 made keyboard shortcuts consistent across the desktop. Therefore, you might find that shortcuts you used before Version 7.9 are different.

- See how a set of keyboard shortcuts you found on File Exchange differs from your current set of keyboard shortcuts.

- See how a set of keyboard shortcuts differs from the default set.

**Steps for Comparing Keyboard Shortcuts**

To compare your current set of keyboard shortcuts to another set:
1 On the Home tab, in the Environment section, click Preferences > Keyboard > Shortcuts.

2 Click the Actions button 🔄.

3 From the drop-down menu, choose the set of keyboard shortcuts to which you want to compare the current set.

4 The Comparison Tool opens and displays the two keyboard shortcut sets side-by-side.

**Read the Results of Comparing Sets of Keyboard Shortcuts**

When you compare keyboard shortcut sets, they appear in the Comparison Tool as follows:

- One set displays on the left side of the tool and the other set displays on the right side of the tool.

- Each column header displays the name of the keyboard shortcut set contained within the column.

- Highlighting identifies rows that differ:
  - Rows that exist in one file, but not the other, appear in green highlighting.
  - Rows that appear in both files, but that differ in content appear in pink highlighting.

- When multiple desktop tools support the same keyboard shortcut for a single desktop action, there is a row for each tool. For example, if both the MATLAB desktop and the Editor support the keyboard shortcut Ctrl+W for closing a selected window, a column of the Comparison Tool might appear like this:

```
<table>
<thead>
<tr>
<th>Close</th>
<th>MATLAB Desktop</th>
<th>Ctrl+W</th>
<th>Closes the selected window</th>
</tr>
</thead>
<tbody>
<tr>
<td>Close</td>
<td>MATLAB Editor</td>
<td>Ctrl+W</td>
<td>Closes the selected window</td>
</tr>
</tbody>
</table>
```

- When there are multiple keyboard shortcuts for the same action in a single tool, there is a row for each keyboard shortcut. For example, if there are two different keyboard shortcuts in the Editor for applying a code analyzer autofix, a column of the Comparison Tool might appear like this:
On Macintosh platforms, the textual format of keyboard shortcuts is slightly different from other platforms, and also differs from the representation shown on MATLAB desktop menus. These differences are due to the Macintosh platform displaying shortcuts using symbols. For instance, the Macintosh platform uses the symbol ⌘ for a keyboard key. Because the Comparison Tool represents symbols as text strings; it specifies the symbol ⌘ as CMD.

See also “Using Comparison Tool Features” on page 6-62.

**Display Keyboard Shortcuts**

The following sections describe the various ways you can display keyboard shortcuts:

- “List All Keyboard Shortcuts in a Set” on page 2-21
- “Display Keyboard Shortcuts on Menus” on page 2-22
- “Display Keyboard Shortcuts in the Preferences Dialog Box” on page 2-22

**List All Keyboard Shortcuts in a Set**

You can copy all the keyboard shortcuts from a keyboard shortcuts set and paste them in a text file or spreadsheet application, such as Microsoft Excel®. To create a list of keyboard shortcuts for easy browsing and future reference, follow these steps:


2. Click the Actions button ⌘.

3. From the drop-down menu, choose Copy to Clipboard.

4. Open a spreadsheet application or a text editor.

   For the best formatting use a spreadsheet application.
5 Paste in the data from the clipboard.

In Microsoft Excel, for example, choose **Home > Paste**.

**Display Keyboard Shortcuts on Menus**

If no keyboard shortcut appears on the menu, one does not currently exist for that action. To create a keyboard shortcut for an action, follow the steps in “Customize Keyboard Shortcuts” on page 2-24.

**Display Keyboard Shortcuts in the Preferences Dialog Box**

To identify a keyboard shortcut when there is no menu option for an action, use the **Keyboard Shortcuts Preferences** pane:

1 On the **Home** tab, in the **Environment** section, click **Preferences > Keyboard > Shortcuts**.

2 In the filter field, type the name of the tool for which you want to list the keyboard shortcuts. For example, type **Editor** to see the keyboard shortcuts currently defined for actions you can perform in the Editor.
3 Narrow the list of **Action names** that the preferences pane displays by adding a string describing the action. For example, add `clear`, if you want to find the keyboard shortcut for clearing selected text in the Editor. Type a short string to increase the likelihood of the filter returning the action you seek.

![Keyboard Shortcuts Preferences](image)

4 Select the action name of interest. In this example, select `Clear Selection`.

5 View the table labeled **Shortcuts for Clear Selection**. It indicates that the `Escape` key is the current keyboard shortcut for the `Clear Selection` action in the Editor.

![Shortcuts for Clear Selection](image)
Customize Keyboard Shortcuts

You can customize or view keyboard shortcuts for MATLAB desktop tools. On the Home tab, in the Environment section, click Preferences > Keyboard > Shortcuts. If you have an active Internet connection, you can watch the Customizable Keyboard Shortcuts video for an overview.

The following sections provide details:

- “Steps for Customizing Keyboard Shortcuts” on page 2-24
- “Filter Keyboard Shortcut Actions” on page 2-27
- “Specify Keystrokes for a Keyboard Shortcut” on page 2-28
- “Evaluate and Resolve Keyboard Shortcut Conflicts” on page 2-29
- “Examples of Creating, Modifying, and Deleting Keyboard Shortcuts” on page 2-31
- “Display Keyboard Shortcuts” on page 2-21

Steps for Customizing Keyboard Shortcuts

1 On the Home tab, in the Environment section, click Preferences > Keyboard > Shortcuts.

2 In the Active settings field, choose the file that contains the set of keyboard shortcuts that you want to customize.

Typically, the first time you modify keyboard shortcuts, you begin with the default settings for your platform. For details, see “Choose a Set of Keyboard Shortcuts” on page 2-16.

3 Under Action name, select the action for which you want to define or modify a keyboard shortcut. An action is the operation for which you want to customize the shortcut, such as Clear Command History.

For tips on finding the action you want, see “Filter Keyboard Shortcut Actions” on page 2-27.

4 Click the Add button.
An editable field opens under the **Shortcut** column.

5 Type the shortcut that you want to use for the action you selected in Step 3. Alternatively, you can choose a shortcut from the drop-down menu.

For details, see “Specify Keystrokes for a Keyboard Shortcut” on page 2-28.

6 Assign the shortcut to the tool or tools with which you want to use it. For example, in the **Tools with shortcut** column:

   a Click the down arrow for the list of desktop tools to which you can assign a shortcut. Not all actions are available with all desktop tools.

   b Select a check box to assign the shortcut to a tool. Clear a check box to remove it.

7 Evaluate and resolve any conflicts, indicated by the informational and error icons.

   For more information, see “Evaluate and Resolve Keyboard Shortcut Conflicts” on page 2-29.

8 Click **Apply**.
   - The keyboard shortcut becomes available immediately.
If a changed shortcut corresponds to a menu option that previously displayed no keyboard shortcut, MATLAB reflects the new keyboard shortcut on the menu.

**Restore Default Keyboard Shortcut Sets**

If you modify keyboard shortcuts, and then decide you do not want to keep the changes, you can restore the default shortcuts. To restore the default state of a keyboard shortcut:

1. Click the **Actions** button.
2. Select **Undo Modifications to Windows Default Set (modified)** or **Undo Modifications to Emacs Default Set (modified)**, as appropriate for your system.
3. Click **OK**.

**Note** Undoing modifications reverts all keyboard shortcuts changes that you made to the set. You cannot undo modifications on a shortcut-by-shortcut basis.

**Save Keyboard Shortcuts to a Settings File**

Save keyboard shortcuts to a settings file to:

- Save changes you make to a default settings file, such as the Windows default set, to a new set.

  MATLAB preserves changes you make to the default sets across sessions. However, if you undo modifications to a default keyboard shortcut set (as described in “Restore Default Keyboard Shortcut Sets” on page 2-26) you lose all changes, unless you first save them to a new set.

- Copy the keyboard shortcuts settings file to another system running MATLAB and use it there.

- Overwrite a settings file that you previously saved.

  You cannot overwrite the default settings files that install with MATLAB. MATLAB saves modifications that you make to a default set using the
Define Keyboard Shortcuts

name of the default set appended with the text (modified). For instance, Windows default (modified).

- Share a keyboard shortcuts settings file with others.

For example, you can submit your file to the File Exchange repository. Click this link to go directly to the page where you can submit your file: MATLAB Central File Exchange — Submit New File.

To save a keyboard shortcuts settings file, follow these steps:


2. Click the Actions button, and then select Save As.

3. In the Save dialog box, navigate to the folder where you want to save the file, specify the file name, and then click Save.

   MATLAB saves the file as an .xml file in the folder that you specified.

Filter Keyboard Shortcut Actions

Use the filter field to see the list of actions for which you can customize or define a keyboard shortcut:

1. Type all or part of any one of the following:
   - An action name, for example, Delete.
     MATLAB displays only the action names or desktop menus that contain the text you specify.
   - The name of a desktop tool or menu, for example, File or Command Window.
     MATLAB displays a list of the action names associated with the tool or menu you specify. In addition, the list includes any action names that contain the name of the tool or menu. For example, if you specify Command History, the list of action names includes Next History Command, which is a Command Window action.
   - A keyboard shortcut, for example, Ctrl+R
MATLAB displays only the action names that have the shortcut you specify. Be aware of the following:

- You can enter most keyboard shortcuts by either pressing keystrokes or typing the key names.

  For example, to enter Ctrl+S, use the keystroke (by pressing the Ctrl key and the S key). Or, type Ctrl+S character by character (C-t-r-l-+Y).

- If using keystrokes for a keyboard shortcut does not work, try typing the characters instead. You must type some keyboard shortcuts character by character, such as shortcuts including the Tab, Backspace, or Delete keys.

- Type numpad to refer to the number pad that is on the far right of some keyboards.

- Type Up or Down to refer to the Up arrow or Down arrow keypad keys, respectively.

2 Verify that an Action name performs the action you expect:

a Hover the mouse pointer over the Action name. For example, Remove Next Word.

b View the tooltip that appears.

Specify Keystrokes for a Keyboard Shortcut

A keystroke can be a single key or the combination of a modifier (Alt, Shift, or Ctrl) and another key. When you create a keyboard shortcut, specify the keystrokes for the shortcut as follows:
1 Click the Add button 📌.

2 Specify the number of keystrokes you want to use for the shortcut:
   - To use the default number of keystrokes, which is one keystroke, skip to step 3.
   - To specify multiple keystrokes, or to specify explicitly one keystroke follow these steps:
     a Click the down arrow next to the key icon 🔄 in the Shortcuts field.
     b Choose Limit to 1 keystroke, Limit to 2 keystrokes, or Limit to 3 keystrokes.

     For instance, Ctrl+F is one keystroke, Ctrl+Y, Shift+Z is two keystrokes, and Ctrl+Y, Shift+Z, F9 is three keystrokes.

3 Specify the keystrokes by doing one of the following:
   - Type the keystrokes, by pressing the keys, not by typing the key names character by character.
     For example, press the Ctrl key and the Y key. Do not type C-t-r-l-+-Y.
   - Choose a keystroke, such as the Tab key, by clicking the down arrow next to the key icon 🔄 in the Shortcuts field. Then, choose the key name.

     The listed keys already have a defined action within dialog boxes. For example, the Tab key navigates from one field to the next in dialog boxes.

**Evaluate and Resolve Keyboard Shortcut Conflicts**

Conflicts arise when two or more different actions have the same shortcut. There is no requirement that you resolve keyboard shortcut conflicts. However, if the same shortcut specifies two different actions, the shortcuts can be confusing to use.

View keyboard shortcut conflicts — On the Home tab, in the Environment section, click Preferences > Keyboard > Shortcuts.

The Keyboard Shortcuts preferences pane indicates conflicts using informational 🔄 and error 📚 icons.
• 🔄—An informational icon indicates that two different actions in two different tools have the same shortcut. For information on resolving these conflicts, see “Actions in Different Tools Have the Same Shortcut — Evaluating Conflicts” on page 2-30.

• 🟥—An error icon indicates that two different actions within the same tool have the same shortcut. For information on resolving these conflicts, see “Actions in the Same Tool Have the Same Shortcut — Evaluating Conflicts” on page 2-30.

**Actions in Different Tools Have the Same Shortcut — Evaluating Conflicts**

Typically, you want to resolve conflicts indicated by the informational icon 🔄 when all the following are true:

- You use both tools frequently.
- You perform both actions frequently.
- You have difficulty remembering the action that the shortcut performs in each tool.

For instance on Microsoft Windows platforms, by default, Ctrl+Shift+U undocks a tool from the MATLAB desktop. However if you select text in the Editor, and then press Ctrl+Shift+U, it changes the selected text to uppercase. If you frequently use both of these actions, you can specify a different keyboard shortcut for one or both actions.

**Actions in the Same Tool Have the Same Shortcut — Evaluating Conflicts**

Typically, you want to resolve conflicts indicated by the error icon 🟥.

It can be *unnecessary* to resolve these conflicts if one or more of the following are true:

- The situation is temporary.

For instance, you are performing a two-step procedure. In the first step, you assign the keyboard shortcut to an action that results in a conflict. Then, in the second step, you remove the shortcut from the original action.
• The two actions are associated with different modes of the same tool.

By default, when the MATLAB Editor is in cell mode, Ctrl+Up and Ctrl+Down move the cursor to the Next and Previous cell, respectively. When the Editor is not in cell mode, those keyboard shortcuts scroll up and scroll down, respectively. The shortcuts are in conflict, but the behavior probably is expected, for the given MATLAB Editor mode.

Although not evident from the preferences pane, Ctrl+C presents a similar situation on Windows systems. Ctrl+C is the keyboard shortcut for interrupting MATLAB execution. However, the default keyboard shortcut for the copy action is also Ctrl+C. Therefore, if you:

- Select an item, and then press Ctrl+C, it copies the selected item to the clipboard, — regardless of whether MATLAB is busy.
- Do not select an item and press Ctrl+C, it interrupts MATLAB execution.

If you change the default keyboard shortcut for the copy action from Ctrl+C to another keystroke, then Ctrl+C interrupts MATLAB execution, regardless of whether you have selected an item.

**Resolve Keyboard Shortcut Conflicts**

To resolve a conflict, change or delete shortcuts such that there is a one-to-one correspondence between a shortcut and a frequently used action. For examples, see “Changing a Keyboard Shortcut” on page 2-32 and “Deleting a Keyboard Shortcut” on page 2-34.

**Examples of Creating, Modifying, and Deleting Keyboard Shortcuts**

• “Creating a New Keyboard Shortcut” on page 2-32
• “Changing a Keyboard Shortcut” on page 2-32
• “Deleting a Keyboard Shortcut” on page 2-34
Creating a New Keyboard Shortcut

By default, no keyboard shortcut is available for adding a Help topic to the list of favorites. If you frequently mark topics as favorites, you can define a keyboard shortcut for this action, as follows:

1 On the Home tab, in the Environment section, click Preferences > Keyboard > Shortcuts.

2 In the filter field, type Help.

3 Scroll through the Action name list, and select Add to Favorites.

4 Click the plus button.

MATLAB adds a row to the table above the plus button.

5 In the Shortcut field, click the down arrow, and then change Limit to 1 keystroke to Limit to 2 keystrokes.

6 In the Shortcut field, press Ctrl+S, and then Alt+V.

Notice that the All possible conflicts table is empty, which indicates that no other desktop action is currently using this combination of keystrokes.

7 Click Apply.

Notice that:

- The Add to Favorites dialog box opens when you press Ctrl+S, Alt+V in the Help browser.
- Ctrl+S, Alt+V appears next to Add to Favorites when you click the Favorites menu in the Help browser.

Changing a Keyboard Shortcut

Suppose you frequently adjust indenting in the MATLAB Editor. However, you have difficulty remembering the default keyboard shortcut of Ctrl+[ for decreasing the indent. So, you decide to change it to something that is easier to remember.
This example changes the keyboard shortcut for **Decrease Indent** in the MATLAB Editor from Ctrl+[ to Ctrl+Backspace:

1. On the **Home** tab, in the **Environment** section, click **Preferences > Keyboard > Shortcuts**.

2. Under **Active settings**, choose **Windows Default Set**.

3. In the filter field, press Ctrl+[.

4. Under **Action name**, select **Decrease Indent**.

5. In the table labeled **Shortcuts for Decrease Indent**, under **Shortcut**, click Ctrl+[. MATLAB makes the field editable.

6. In the **Shortcut** field, press **Ctrl+Backspace** twice.

   The first time you press the key combination, it deletes Ctrl+[. The second time you press it, Ctrl+Backspace appears in the field.
7 Click Apply.

MATLAB saves your changes to the Windows Default Set (modified) settings.

Deleting a Keyboard Shortcut
Suppose you find yourself frequently pressing the wrong keyboard shortcut. For example, on Windows, you press Alt+Enter (to apply a code analyzer autofix) instead of Ctrl+Enter (to evaluate the current cell in the MATLAB Editor). To avoid accidentally applying an autofix, delete the Alt+Enter shortcut by following these steps:

1 On the Home tab, in the Environment section, click Preferences > Keyboard > Shortcuts.

2 Under Active settings, choose Windows Default Set or Windows Default Set (modified).

3 In the filter field, press Alt+Enter.

4 Under Action name, select the row containing Autofix Message.

5 In the next table, under Shortcuts for Autofix Message, select the row containing Alt+Enter.

6 Click the remove button.

7 Click Apply.

If it does not exist, MATLAB creates a Windows Default Set (modified) keyboard shortcut set. This set consists of the Windows Default Set
Define Keyboard Shortcuts

of keyboard shortcuts, less the shortcut for Alt+Enter. If the Windows Default Set (modified) settings file exists, then MATLAB deletes the Alt+Enter keyboard shortcut from that set of keyboard shortcuts.

See also “Delete a Set of Keyboard Shortcuts” on page 2-35.

Delete a Set of Keyboard Shortcuts

If you previously saved or copied a set of keyboard shortcuts to your system and you no longer want it, delete it as follows:

1 On the Home tab, in the Environment section, click Preferences > Keyboard > Shortcuts.

2 Under Active settings, choose the set of keyboard shortcuts that you want to delete.

You cannot delete default keyboard shortcut sets, such as Windows Default Set.

3 Click the Actions button and choose Delete filename, where filename is the name of a keyboard shortcut set you previously saved or copied to your system.

For information on deleting a single keyboard shortcut from a set that you want to keep, see “Deleting a Keyboard Shortcut” on page 2-34.

Use Keyboard Shortcuts Settings Files Created on Other Systems

If you find a keyboard shortcuts settings file that is useful to you, or if you want to use one you created on a different system, make it the active settings file as follows:

1 Copy the settings file to a folder on your system, such as:

   I:\my_matlab_files\active_settings_files\new_settings.xml

2 On the Home tab, in the Environment section, click Preferences > Keyboard > Shortcuts.
3 In the **Active settings** field, click the down arrow, and then click **Browse**.

4 In the Open dialog box, navigate to the folder where you copied the settings file.

5 Select the settings file, and then click **Open**.

6 In the Keyboard Shortcuts preferences pane, click **Apply**. The settings file you specified is now the active settings file for MATLAB.

**Keyboard Shortcut Restrictions**
These sections describe the tools, portions of tools, and actions for which you cannot change keyboard shortcuts:

- “Tools for Which You Cannot Customize Keyboard Shortcuts” on page 2-36
- “Actions for Which You Cannot Customize Keyboard Shortcuts” on page 2-36

**Tools for Which You Cannot Customize Keyboard Shortcuts**
You cannot change the keyboard shortcuts associated with the following tools or portions of tools:

- Figure windows—For example, you cannot modify the keyboard shortcut, **Ctrl+S** for saving a MATLAB **.fig** file.
- Toolboxes—For example, you cannot modify keyboard shortcuts in the SimBiology® desktop.
- Incremental search—You can modify the keyboard shortcuts for initiating a forward or backward incremental search. However, you cannot change the keyboard shortcuts that you use within incremental search mode, such as **Ctrl+Shift+S** to search forward.
- Dialog boxes—For example, you cannot create a keyboard shortcut for the **OK** button.

**Actions for Which You Cannot Customize Keyboard Shortcuts**
The following table describes some frequently used actions for which you cannot customize keyboard shortcuts.
<table>
<thead>
<tr>
<th><strong>Action</strong></th>
<th><strong>Keyboard Shortcut</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Cancel the current action.</td>
<td><strong>Esc</strong> (escape)</td>
</tr>
<tr>
<td>For example, if you select the <strong>Edit</strong> menu,</td>
<td></td>
</tr>
<tr>
<td>the menu items display.</td>
<td></td>
</tr>
<tr>
<td>Pressing <strong>Esc</strong> retracts the menu items.</td>
<td></td>
</tr>
<tr>
<td>In the Function Browser, pressing <strong>Esc</strong> up to</td>
<td></td>
</tr>
<tr>
<td>three times has the following effects:</td>
<td></td>
</tr>
<tr>
<td>1 Dismisses the search history</td>
<td></td>
</tr>
<tr>
<td>2 Clears the search field</td>
<td></td>
</tr>
<tr>
<td>3 Closes the Function Browser</td>
<td></td>
</tr>
<tr>
<td>Interrupt MATLAB execution on all supported</td>
<td><strong>Ctrl+C</strong></td>
</tr>
<tr>
<td>platforms.</td>
<td></td>
</tr>
<tr>
<td>Interrupt MATLAB execution on Windows and UNIX</td>
<td><strong>Ctrl+Cancel</strong></td>
</tr>
<tr>
<td>systems.</td>
<td></td>
</tr>
<tr>
<td>Interrupt MATLAB execution on Macintosh</td>
<td><strong>Cmd+.</strong> (period)</td>
</tr>
<tr>
<td>systems.</td>
<td></td>
</tr>
<tr>
<td>Open context menu on Windows and UNIX systems.</td>
<td><strong>Ctrl+Shift+F10</strong></td>
</tr>
<tr>
<td>Close the desktop and consequently shut down</td>
<td><strong>Alt+F4</strong></td>
</tr>
<tr>
<td>the MATLAB program.</td>
<td></td>
</tr>
<tr>
<td>Outside the desktop, close the active window</td>
<td></td>
</tr>
<tr>
<td>(except on Macintosh platforms).</td>
<td></td>
</tr>
<tr>
<td>Action</td>
<td>Keyboard Shortcut</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>----------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Accessibility affordances</td>
<td><strong>Tab</strong> for navigating through fields in dialog boxes, for example.</td>
</tr>
</tbody>
</table>
| Make an open tool the active tool          | • Command Window: **Ctrl+0**  
• Command History: **Ctrl+1**  
• Current Folder: **Ctrl+2**  
• Workspace: **Ctrl+3**  
• Profiler: **Ctrl+4**  
• Figure Palette: **Ctrl+6**  
• Plot Browser: **Ctrl+7**  
• Property Editor: **Ctrl+8**  
• Editor: **Ctrl+Shift+0**  
• Figures: **Ctrl+Shift+1**  
• Web browser: **Ctrl+Shift+2**  
• Variables Editor: **Ctrl+Shift+3**  
• Comparison Tool: **Ctrl+Shift+4**  
• Help browser: **Ctrl+Shift+5**          |
Set Print Options

MATLAB provides special page setup options for printing from the Command Window and Editor. The setup options are essentially the same for both tools, with minor variations. This section covers their use:

**Page Setup Options**
To specify page setup options, perform these steps:

1. In the tool you want to print from, for example, the Command Window, select **Page Setup**.

   The Page Setup dialog box opens for that tool.

2. Click the **Layout**, **Header**, or **Fonts** tab in the dialog box and set those options for that tool, as detailed in subsequent sections.

3. Click **OK**.

4. After specifying the options, select **Print** in the tool you want to print from, for example, the Command Window.

   The contents from the tool print, using the options you specified in Page Setup.

**Layout Options for Page Setup**
You can specify the following layout options. A preview area shows you the effects of your selections.

- **Print header** — Print the header specified in the **Header** pane.
- **Print line numbers** — Print line numbers.
• **Wrap lines** — Wrap any lines that are longer than the printed page width.

• **Syntax highlighting** — For keywords and comments that are highlighted in the Command Window, specify how they are to appear in print. Options are black and white text (that is, no highlighting), colored text (for use with a color printer), or styled text. For styled text, keywords appear in bold, comments appear in italics, and all other text appears in the normal style. Only keywords and comments you input in the Command Window are highlighted; output is not highlighted.

**Header Options for Page Setup**

If you want to print a header, select the **Layout** tab and then select **Print header**. Next, select the **Header** tab and specify how the elements of the header are to appear. A preview area shows you the effects of your selections:

• **Page number** — Format for the page number, for example # of n

• **Border** — Border style for the header, for example, Shaded box

• **Layout** — Layout style for the header. For example, **Standard one line** includes the date, time, and page number all on one line

**Fonts Options for Page Setup**

Specify the font to use for the printed contents:

1. From **Choose font**, select the element, either **Body** or **Header**, where **Body** text is everything except the **Header**.

2. Select the font to use for the element.

   For example, if you access this dialog box while using the Command Window, you can select **Use Command Window font** for **Body** text. The printed text matches the Command Window font.

3. Repeat for the other element.

   If you did not select **Print header** on the **Layout** pane, you do not need to specify the **Header** font.
As an example, for Header text, select **Use custom font** and then specify the font characteristics—type, style, and size. After you specify a custom font, the Sample area shows how the font will look.

**Tip** You can change the font that a desktop tool uses. On the Home tab, in the Environment section, click Preferences > Fonts > Custom.
Web Browsers and MATLAB

From MATLAB, Web sites and documents can display in any of the following browsers:

- MATLAB Web browser
- Help browser
- Your system Web browser, such as Mozilla® Firefox®

MATLAB uses the different browsers to display different types of information:

- Web sites display in your system browser.
- Documentation displays in the Help browser.
- Other HTML files display in the MATLAB Web browser. For example, after publishing a MATLAB program file to HTML, the HTML file displays in the MATLAB Web browser:
MATLAB Web and Help Browsers

The MATLAB Web and Help browsers may not support all the features that a particular Web site or HTML page uses. For example, the MATLAB Web browser does not display .bmp (bitmap) image files. Instead use .gif or .jpeg formats for image files in HTML pages.
System Browser
The system browser that MATLAB uses depends on your platform:

- On Microsoft Windows and Apple Macintosh platforms, MATLAB uses the default browser for your operating system.
- On UNIX platforms, MATLAB uses the Mozilla Firefox browser. You can specify a different system browser for MATLAB using Web preferences.

Display Pages in Web Browsers
To display an HTML document in the MATLAB Web browser, double-click the document name in the Current Folder browser.

To display a Web page or any file type in the MATLAB Web browser:

1. Open the browser using the `web` command.
2. Type a URL or full path to a filename in the Location field.

Specify Proxy Server Settings for Connecting to the Internet
If your network uses a firewall or another method of protection that restricts Internet access, provide information about your proxy server to MATLAB. Be aware that:

- MATLAB supports non-authenticated, basic, digest, and NTLM proxy authentication types.
- You cannot specify the proxy server settings using a script.
- There is no automated way to provide the proxy server settings your system browser uses to MATLAB.

To specify the proxy server settings:

2. Select the Use a proxy server to connect to the Internet check box.
3. Specify values for Proxy host and Proxy port.
Examples of acceptable formats for the host are: 172.16.10.8 and ourproxy. For the port, enter an integer only, such as 22. If you do not know the values for your proxy server, ask your system or network administrator for the information.

If your proxy server requires a user name and password, select the **Use a proxy with authentication** check box. Then enter your proxy user name and password.

**Note** MATLAB stores the password without encryption in your `matlab.prf` file.

4 Ensure that your settings work by clicking the **Test connection** button.

MATLAB attempts to connect to `http://www.mathworks.com`:

- If MATLAB can access the Internet, **Success!** appears next to the button.
- If MATLAB cannot access the Internet, **Failed!** appears next to the button. Correct the values you entered and try again. If you still cannot connect, try using the values you used when you authenticated your MATLAB license.

5 Click **OK** to accept the changes.

**Specify the System Browser for UNIX Platforms**

To specify the system browser:

1 On the **Home** tab, in the **Environment** section, click **Preferences > Web**.

2 Under **System Web browser**, in the **Command** field, specify the system command to open the browser, for example, `opera`, which opens the Opera Web browser.

**Note** The **System Web browser** preference is for UNIX platforms (excluding Macintosh) and does not appear in the preferences pane for other platforms.
3 Add options for opening your system browser in the **Options** field. For example, **geometry 1064x860** specifies the size of the window for Opera.

4 Click **OK**.
License Management and Software Updates

In this section...

“Manage Your Licenses” on page 2-47
“Check for Software Updates” on page 2-48

Manage Your Licenses

You can use the MATLAB licensing features to perform license management activities, such as activating licenses, deactivating licenses, or updating licenses. You also can visit the License Center at the MathWorks Web site to perform other license-related activities.

To access the licensing feature:

1 On the Home tab, in the Resources section, click Licensing.

2 Select a Licensing option. The following table describes the Licensing options. Depending on your license type, your system might not include all of these options.

Note Some options require an Internet connection. If your Internet connection requires a proxy server, use MATLAB Web preferences to specify the server host and port. See “Specify Proxy Server Settings for Connecting to the Internet” on page 2-44 for more information.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Update Current Licenses</td>
<td>Displays a list of all your MathWorks licenses on this computer, with their current status. When you select a license and click Update Selected License, MATLAB contacts MathWorks to retrieve the most current version of the License File for the license. The update process</td>
</tr>
</tbody>
</table>

2-47
<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Activate Software</strong></td>
<td>Starts the activation application, which walks you through the activation process. Answer the questions on each dialog box, select the license you want to activate, and click <strong>Activate</strong>.</td>
</tr>
<tr>
<td><strong>Deactivate Software</strong></td>
<td>Displays a list of all your MathWorks licenses on this computer, with their current status. When you select a license and click <strong>Deactivate Selected License</strong>, MATLAB deactivates all releases on this computer associated with the license, and updates the licensing information at the MathWorks Web site. You will not be able to use MathWorks software with that license on this computer. If you are not connected to the Internet, MATLAB deactivates the licences on your computer but cannot update the corresponding license information stored at the MathWorks Web site. In this scenario, MATLAB returns a deactivation string. To complete deactivation, save a copy of this string, go to a computer with an Internet connection, and visit the License Center at the MathWorks Web site. There you can log in to your MathWorks Account and enter the deactivation string.</td>
</tr>
<tr>
<td><strong>Manage Licenses</strong></td>
<td>Starts a Web browser, opening the My Licenses page associated with your MathWorks Account. You can use this page, called the License Center, to perform many licensing activities.</td>
</tr>
</tbody>
</table>

**Check for Software Updates**

To determine if more recent versions of your MathWorks products are available, and to view latest version numbers for all MathWorks products, follow these steps:

1. Make sure you have an active Internet connection.
2 Select **Help > Check for Updates**. The Check for Updates dialog box displays.

3 From the **Select View** list, choose to view the latest version numbers for all MathWorks products installed on your system, or all MathWorks products.

   The latest versions display.

4 Click any column heading to sort or reverse the sort order by that column.

5 Use the What’s New column to access the release notes for a product.

   Release notes document new features and changes, bug reports, and compatibility considerations.

6 Decide whether you want to upgrade to the most recent version.

   • If you do, click **Download Products at MathWorks.com**
   • If you do not, go to step 7.

7 Click **Close**.
Macintosh Platform Conventions

In this section...

“Mouse Instructions and Macintosh Platforms” on page 2-50
“Navigating Within the MATLAB Root Folder on Macintosh Platforms” on page 2-50

Mouse Instructions and Macintosh Platforms
The documentation typically presents conventions for Microsoft Windows platforms. Therefore, some conventions and operations differ on the Macintosh platform from those that appear in the rest of the documentation. The intended action for the Macintosh platform is typically obvious. Mouse operations follow Macintosh conventions.

Make the following replacements to adjust documented mouse instructions for Macintosh platforms if you are using a one-button mouse:

- Replace right-click with Ctrl+click
- Replace middle-click with Command+click

Navigating Within the MATLAB Root Folder on Macintosh Platforms
On Macintosh platforms, MATLAB is installed as an application bundle. The root folder, the string returned by the `matlabroot` command, has a .app extension.

To view the contents of the MATLAB root folder in the Mac Finder, right-click the MATLAB application bundle, and then select Show Package Contents from the context menu.

To view the content of the MATLAB root folder from within MATLAB:

1. On the Home tab, in the File section, click 🗃️.

2. In the File Browser dialog box, press Command+Shift+G to open the Go To Folder dialog box.
3 Enter the full path to the MATLAB folder, for example, 
\(/\text{Applications/MATLAB\_R2012a.app}\).

4 Press Go.

To open a file with a MATLAB command, such as edit, specify the full path of the MATLAB root folder. For example:

\texttt{edit(\texttt{fullfile(matlabroot,'/toolbox/matlab/demos/lotka.m'))}}
Preferences

In this section...

“Set Preferences for MATLAB” on page 2-52
“Where MATLAB Stores Preferences” on page 2-53
“Preferences Folder and Files MATLAB Uses When Multiple MATLAB Releases Are Installed” on page 2-54
“General Preferences” on page 2-55
“MAT-Files Preferences” on page 2-56
“Confirmation Dialogs Preferences” on page 2-57
“Source Control Preferences” on page 2-59
“Java Heap Memory Preferences” on page 2-59
“Keyboard Shortcuts Preferences” on page 2-60
“Fonts Preferences” on page 2-62
“Fonts Custom Preferences” on page 2-63
“Colors Preferences” on page 2-64
“Colors Programming Tools Preferences” on page 2-65
“Toolbars Preferences” on page 2-66
“Web Preferences” on page 2-67

Set Preferences for MATLAB
MATLAB provides a variety of options called preferences for customizing MATLAB. To access and set preferences:

1. On the Home tab, in the Environment section, click Preferences.
2. From the left pane of the Preferences dialog box, select a tool, product, or an entry revealed when you click a plus sign (+) preceding a tool or product name.
3. Change settings in the right pane of the Preferences dialog box.
4. Click Apply or OK.
Preferences take effect immediately. They remain persistent across sessions of MATLAB.

**Function Alternative**
Open the Preferences dialog box using the `preferences` function.

**Where MATLAB Stores Preferences**
MATLAB and other MathWorks products store their preferences in the file `matlab.prf`. This file loads when you start MATLAB. The folder containing this file is called the preferences folder. The preference folder also contains other related files.

**The Path to and File Name for the Preferences Folder**
To see the full path for the folder where `matlab.prf` and related files are located, type `prefdir` in the MATLAB Command Window.

On Apple Macintosh platforms, the folder can be in a hidden folder, for example, `myname/.matlab/R2009b`. If so, to access the hidden folder:

1. In the Apple Mac OS Finder tool, select Go > Go to Folder.
2. In the resulting dialog box, type the path returned by `prefdir`, and then press Enter.

The name of the preferences folder, matches the name of the release. For instance, for MATLAB R2010b, the name of the preferences folder is `R2010b`.

**Effects of Changing Preferences**
When you change preferences using the MATLAB Desktop, it updates `matlab.prf`. When you close MATLAB, it saves those changes to `matlab.prf`.

**Effects of Installation and Deinstallation on the Preferences Folder**
Installing MATLAB has no effect on the preferences folder. That is, MATLAB creates, checks, copies, and writes to the preferences folder when you start up MATLAB, not when you install it. When you uninstall MATLAB, there
is an option in the uninstaller to remove the preferences folder. However, this option is not selected by default.

Preferences Folder and Files MATLAB Uses When Multiple MATLAB Releases Are Installed

The files in the preferences folder that MATLAB uses depends on the version of MATLAB you are starting up. How and if MATLAB migrates (reuses) preferences files from one version to the next also depends on the version.

Process MATLAB Uses to Create and Migrate the Preferences Folder and its Files

When you start it up, MATLAB looks for a preferences folder name that matches the release starting up, and then does one of the following:

- If MATLAB finds a preferences folder name matching the release starting up, it uses that folder and the files within it.
  
  If that folder is empty, MATLAB recreates the default files for the release starting up.

- If MATLAB does not find a preferences folder name matching the release starting up, it creates one. Then, MATLAB checks to see if the release of MATLAB that immediately precedes the one you are starting up is installed.
  
  - If that previous release is not installed, MATLAB recreates the folder and default files for the version starting up.
    
    For example, if you start up R2010b and R2010a is not installed, then MATLAB recreates the default files for the R2010b preferences folder. This is true even if R2009b or earlier is installed.
  
  - If that previous release is installed, MATLAB migrates the files from the preferences folder corresponding to that previous release to the preferences folder for the release starting up.
    
    For example, if you start up R2010b and R2010a is installed, then MATLAB migrates the files from R2010a preferences folder to the R2010b preferences folder.
Control the Preferences Files MATLAB Uses
This table describes how to control which versions of preferences files MATLAB uses.

<table>
<thead>
<tr>
<th>To Use:</th>
<th>Do This:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default preference files for a given release of MATLAB</td>
<td>Make sure the preferences folder for that release exists, but is empty before starting up that MATLAB version.</td>
</tr>
<tr>
<td>All the preference files from the release of MATLAB immediately preceding the release you plan to start up</td>
<td>Ensure that the preferences folder exists for that preceding release. If so, delete the entire preferences folder for the release of MATLAB you plan to start up.</td>
</tr>
<tr>
<td>The release-specific default for just a particular file in the preferences folder</td>
<td>Delete just that file from the preferences folder for the release of MATLAB you plan to start up. One file to consider keeping is history.m. For more information, see “Command History” on page 3-29.</td>
</tr>
</tbody>
</table>

General Preferences
You can set preferences for toolbox path caching, figure window printing, and deleting files.

On the Home tab, in the Environment section, click Preferences > General. Then, adjust preference options as described in this table.
<table>
<thead>
<tr>
<th>Preference</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Toolbox path caching</td>
<td>Select <strong>Enable toolbox path caching</strong> to have MATLAB cache toolbox folder information across sessions for quicker startup performance. Select <strong>Enable toolbox path cache diagnostics</strong> to display information about startup time when you start MATLAB. Click <strong>Update Toolbox Path Cache</strong> to add files to the toolbox folders under the <code>matlabroot</code> folder. (Use after you use tools not provided with MATLAB to create MATLAB files.) For details, see “Toolbox Path Caching in the MATLAB Program” on page 1-23.</td>
</tr>
<tr>
<td>Figure window printing</td>
<td>Select an option to specify how colored lines and text appear in printed output.</td>
</tr>
<tr>
<td>Deleting Files</td>
<td>Select an option to specify what MATLAB does with files you delete using the <code>delete</code> function. Selecting <strong>Delete permanently</strong> makes the <code>delete</code> function run faster. For details, see “Deleting Files and Folders Using Functions” on page 6-38.</td>
</tr>
</tbody>
</table>

**MAT-Files Preferences**

You can set the default MATLAB version for MAT-files and FIG-files. These preferences apply to both the `save` function and the **Save** menu options. However, the `matfile` function creates only Version 7.3 MAT-files. On the **Home** tab, in the **Environment** section, click **Preferences > General > MAT-Files**. Then, adjust preference options as described in the table below.

For more details on the features supported in each version, see the `save` reference page.
Option | Use to:
---|---
MATLAB Version 7.3 or later (save -v7.3) | Load or save parts of variables, or save variables larger than 2 GB on 64-bit systems. As with Version 7, files are compressed and use Unicode® character encoding.

MATLAB Version 7 or later (save -v7) | Save compressed MAT-files that use Unicode character encoding. This is the default on new installations of MATLAB software and upgrades from versions earlier than 7.3.

MATLAB Version 5 or later (save -v6) | Save MAT-files for use with versions prior to MATLAB Version 7, or create uncompressed files.

**Confirmation Dialogs Preferences**

You can specify whether or not MATLAB displays specific confirmation dialog boxes.

On the Home tab, in the Environment section, click Preferences > General > Confirmation Dialogs. Then, adjust preference options as described in the table below.

This table summarizes the core MATLAB confirmation dialog boxes. There might be additional confirmation dialog boxes for other products you install.

<table>
<thead>
<tr>
<th>Option</th>
<th>Confirmation Dialog Box Appears</th>
</tr>
</thead>
<tbody>
<tr>
<td>Warn before deleting Command History items</td>
<td>When you delete entries from the Command History window. For details, see “Using Command History Commands” on page 3-29.</td>
</tr>
<tr>
<td>Warn before clearing the Command Window</td>
<td>When, on the Home tab, in the Code section, you click Clear Commands. Does not appear when you use the clc function.</td>
</tr>
<tr>
<td>Option</td>
<td>Confirmation Dialog Box Appears</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Confirm when overwriting variables in MAT-files</strong></td>
<td>When you save variables by dragging them from the Workspace browser onto a MAT-file in the Current Folder browser. For details, see “Creating and Updating MAT-Files with the Current Folder Browser” on page 6-32.</td>
</tr>
<tr>
<td><strong>Confirm when overwriting workspace variables via drag-and-drop</strong></td>
<td>When you load variables by dragging them from the Details Panel of the Current Folder browser to the Workspace browser or Command Window.</td>
</tr>
<tr>
<td><strong>Prompt when editing files that do not exist</strong></td>
<td>When you type <code>edit</code> <code>filename</code> and <code>filename</code> does not exist in the current folder or on the search path.</td>
</tr>
<tr>
<td><strong>Prompt to exit debug mode when saving file</strong></td>
<td>When you try to save a modified file while in debug mode. For details, see “End Debugging”.</td>
</tr>
<tr>
<td><strong>Prompt to save on activate</strong></td>
<td>When you have unsaved changes to a figure and program file and you activate the GUI by clicking the Run button, for example. For details, see “GUIDE Preferences”.</td>
</tr>
<tr>
<td><strong>Prompt to save on export</strong></td>
<td>When you have unsaved changes to a figure and program file and you select File &gt; Export. For details, see “GUIDE Preferences”.</td>
</tr>
<tr>
<td><strong>Confirm changing default callback implementation</strong></td>
<td>When you have modified a callback signature in GUIDE. For details, see “GUIDE Preferences”.</td>
</tr>
<tr>
<td><strong>Confirm before exiting MATLAB</strong></td>
<td>When you quit MATLAB.</td>
</tr>
<tr>
<td><strong>Confirm when deleting variables</strong></td>
<td>When you delete variables from the workspace using menu items. Does not appear with the clear function. For details, see “Save, Load, and Delete Workspace Variables” on page 5-9.</td>
</tr>
</tbody>
</table>
Source Control Preferences
You can select which previously installed and configured source control system to use with MATLAB.

On the Home tab, in the Environment section, click Preferences > General > Source Control. Then, select an option from the list.

For detailed information on setting up and using a source control system with MATLAB, see “Set Up Source Control (Microsoft Windows)” and “Source Control Interface on UNIX Platforms”.

Java Heap Memory Preferences
You can adjust the amount of memory that MATLAB software allocates for Java objects.

Note The default heap size is sufficient for most cases.

To adjust the Java heap size:

1 On the Home tab, in the Environment section, click Preferences > General > Java Heap Memory.

2 Select a Java heap size value using the slider or spin box.

Note Increasing the Java heap size decreases the amount of memory available for storing data in arrays.

3 Click OK.

4 Restart MATLAB.

If the amount of memory you specified is not available upon restart, MATLAB resets the value to the default, and displays an error dialog box. To readjust the value, repeat the previous steps.
If increasing the heap size does not eliminate memory errors, check your Java code for memory leaks. Eliminate references to objects that are no longer useful. For more information, see the Java SE Troubleshooting guide at http://www.oracle.com/technetwork/java/javase/index-138283.html.

**Keyboard Shortcuts Preferences**

You can set keyboard shortcuts for actions you perform using MathWorks software. You can specify or import sets of predefined keyboard shortcuts, set individual shortcuts on an action-by-action basis, or use a combination of both approaches.

On the Home tab, in the Environment section, click Preferences > Keyboard > Shortcuts. Then, adjust preference options as described in the table below.

For step-by-step instructions, see “Customize Keyboard Shortcuts” on page 2-24.

<table>
<thead>
<tr>
<th>Preference</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active settings</td>
<td>Select or import a set of predefined keyboard shortcuts. For details, see “Choose a Set of Keyboard Shortcuts” on page 2-16 and “Use Keyboard Shortcuts Settings Files Created on Other Systems” on page 2-35.</td>
</tr>
<tr>
<td></td>
<td>Select any one of these options:</td>
</tr>
<tr>
<td></td>
<td>• <strong>Save As</strong>—Save active settings to a file. For details, see, “Display Keyboard Shortcuts” on page 2-21.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Copy to clipboard</strong>— so you can import into Microsoft Excel, for example.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Compare</strong> active settings to another set. For details, see, “Compare” on page 2-21.</td>
</tr>
<tr>
<td>Preference</td>
<td>Usage</td>
</tr>
<tr>
<td>------------</td>
<td>-------</td>
</tr>
<tr>
<td>For details, see, “Compare Sets of Keyboard Shortcuts” on page 2-19.</td>
<td></td>
</tr>
<tr>
<td><strong>Undo Modifications</strong> to a default keyboard shortcut set.</td>
<td></td>
</tr>
<tr>
<td><strong>Delete</strong> a set of keyboard shortcuts you previously saved or added.</td>
<td></td>
</tr>
<tr>
<td>For details, see “Delete a Set of Keyboard Shortcuts” on page 2-35.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Search by action name or shortcut</th>
<th>Search the list of displayed actions.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Shortcuts for</strong> &lt;action-name&gt;</td>
<td>View the keyboard shortcut assigned to a selected action.</td>
</tr>
<tr>
<td>+ −</td>
<td>Add or delete a keyboard shortcut to a selected action.</td>
</tr>
<tr>
<td></td>
<td>For details, see, “Examples of Creating, Modifying, and Deleting Keyboard Shortcuts” on page 2-31.</td>
</tr>
</tbody>
</table>
### Preference Usage

**All possible conflicts**
- Display conflicts when two or more different actions have the same shortcut.
- For details, see “Evaluate and Resolve Keyboard Shortcut Conflicts” on page 2-29.

**Unassign**
- Remove the keyboard shortcut from the selection in the All possible conflicts list.
- For details, see “Evaluate and Resolve Keyboard Shortcut Conflicts” on page 2-29.

### Fonts Preferences
You can set and specify different fonts for desktop tools.

On the **Home** tab, in the **Environment** section, click **Preferences > Fonts**. Then, set options as described in the table below.

For step-by-step instructions on setting these preferences, see “Fonts” on page 2-2.

<table>
<thead>
<tr>
<th>Preference</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Desktop code font</strong></td>
<td>Specify the font (type, style, and size in points) for tools assigned to use the desktop code font.</td>
</tr>
<tr>
<td></td>
<td>To change the list of tools that use the desktop code font, see “Fonts Custom Preferences” on page 2-63.</td>
</tr>
<tr>
<td><strong>Desktop text font</strong></td>
<td>Specify the font (type, style, and size in points) for tools assigned to use the desktop text font.</td>
</tr>
<tr>
<td></td>
<td>To have the desktop text font use the same font as your system, select <strong>Use system font</strong>. Otherwise, clear <strong>Use system font</strong></td>
</tr>
<tr>
<td></td>
<td>and specify the type, style, and size in points in the fields provided. To change the list of tools that use the desktop text font,</td>
</tr>
<tr>
<td>Preference</td>
<td>Usage</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>tools that use the desktop text font, see “Fonts Custom Preferences” on page 2-63.</td>
<td>See which tools currently do not use the desktop text font or the desktop code font. By default, the Profiler and Comparison Tool use a custom font.</td>
</tr>
<tr>
<td>Custom fonts</td>
<td>Give the desktop a smoother appearance.</td>
</tr>
<tr>
<td>Use antialiasing to smooth desktop fonts</td>
<td>This option is not provided on Microsoft Windows or Apple Macintosh platforms, because MATLAB follows the operating system’s font settings on these platforms.</td>
</tr>
</tbody>
</table>

**Fonts Custom Preferences**

You can override font settings for individual desktop tools, as described in the table that follows. Desktop tools otherwise use the settings that the “Fonts Preferences” on page 2-62 specify.

On the Home tab, in the Environment section, click Preferences > Fonts > Custom. Then, set options as described in the table below.

For step-by-step instructions on setting these preferences, see “Fonts” on page 2-2.

---

4. Linux is a registered trademark of Linus Torvalds.

5. UNIX is a registered trademark of The Open Group in the United States and other countries.
<table>
<thead>
<tr>
<th>Preference</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Desktop tools</td>
<td>Select the desktop tool for which you want to view or customize fonts, such as the Command Window or Editor.</td>
</tr>
<tr>
<td>Font to use</td>
<td>Indicates the font currently being used in the selected desktop tool. Use one of these fonts to change it.</td>
</tr>
<tr>
<td></td>
<td>• Desktop code</td>
</tr>
<tr>
<td></td>
<td>Uses the characteristics of the desktop code font, as described in “Fonts Preferences” on page 2-62.</td>
</tr>
<tr>
<td></td>
<td>• Desktop text</td>
</tr>
<tr>
<td></td>
<td>Uses the characteristics of the desktop text font, as described in “Fonts Preferences” on page 2-62.</td>
</tr>
<tr>
<td></td>
<td>• Custom</td>
</tr>
<tr>
<td></td>
<td>Uses the type, style, and size you specify in the fields.</td>
</tr>
<tr>
<td></td>
<td>For the Profiler and Comparison Tool, you can change the font size, but changes to the font style (for example, bold or italic) have no effect.</td>
</tr>
</tbody>
</table>

**Colors Preferences**

You can specify the text and background color for desktop tools, as well as colors for highlighting syntax elements of MATLAB code.

On the **Home** tab, in the **Environment** section, click **Preferences > Colors**. Then, set options as described in the table below.
Preferences

Preference Usage

Desktop tool colors
Specify that desktop tools use the same text and background colors that your platform uses for other applications by selecting **Use system colors**.

Customize colors by clearing **Use system colors**, and then choose **Text** and **Background** colors from the drop-down menus.

These colors do not apply to the Help display pane, nor to the Web browser.

For details, see “Changing Text, Background, and Hyperlink Colors in Desktop Tools” on page 2-7.

MATLAB syntax highlighting colors
Set colors to help you quickly identify elements of MATLAB syntax in the Editor, Command Window, Command History window, and the MATLAB shortcuts callback area.

For details, see “Changing Syntax Highlighting Colors” on page 2-8.

MATLAB Command Window colors
Set colors to help you quickly identify errors, warnings, and hyperlinks in the Command Window.

Colors Programming Tools Preferences
You can specify options used for editing and debugging code, including code analysis colors, variable and function colors, and cell display options.

On the **Home** tab, in the **Environment** section, click **Preferences > Colors > Programming**. Then, set options as described in the table below.
<table>
<thead>
<tr>
<th>Preference</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Code analyzer colors</strong></td>
<td>• <strong>Warnings</strong>—Specifies the color Code Analyzer uses to identify code in the Editor for which there are warning messages.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Autofix highlight</strong>—Specifies the color Code Analyzer uses to identify code in the Editor for which there is an automatic fix.</td>
</tr>
<tr>
<td></td>
<td>For details, see “ Automatically Check Code in the Editor — Code Analyzer”.</td>
</tr>
<tr>
<td><strong>Variable and function colors</strong></td>
<td>• <strong>Automatically highlight</strong>—Specifies the color the Editor uses to highlight all occurrences of a specific variable or function. For details, see “Find and Replace Functions or Variables in the Current File”.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Variables with shared scope</strong>—Specifies the color of variables with shared scope. The text is colored, not shaded. For details, see “Check Variable Scope in Editor”</td>
</tr>
<tr>
<td><strong>Section display options</strong></td>
<td><strong>Highlight sections</strong>—Specifies the color the Editor uses to shade code sections.</td>
</tr>
<tr>
<td></td>
<td><strong>Show lines between sections</strong>—Specifies that code section divisions appear with a gray line between each section in the Editor. These lines do not appear in the published or printed file.</td>
</tr>
<tr>
<td></td>
<td>See also “Run Code Sections”.</td>
</tr>
</tbody>
</table>

**Toolbars Preferences**

You can customize some toolbars in the MATLAB application.

On the **Home** tab, in the **Environment** section, click **Preferences > Toolbars**. Then, set options as described in the table below.

For step-by-step instructions on setting these preferences, see “Access Frequently Used Features” on page 2-10.
Preferences

Preference Usage

Toolbar Select the toolbar you want to customize.

Layout Rearrange the order of controls in the toolbar by dragging and dropping them to a new location in the Layout.

Controls Select which buttons appear on the selected toolbar.

Web Preferences
Web preferences enable you to specify Internet connection information to MATLAB.

Note

Limitations

- MATLAB supports nonauthenticated, basic, digest, and NTLM proxy authentication types.
- You cannot specify proxy server settings using a script.
- There is no automated way to provide MATLAB with the proxy server settings that your system browser uses.

You can set Web preferences, on the Home tab, in the Environment section. Click Preferences > Web, and then adjust preference options as described in the table below.

<table>
<thead>
<tr>
<th>Preference</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use a proxy server to connect to the Internet</td>
<td>Provide information that MATLAB needs to access the internet when your network uses a firewall or another method of protection that restricts Internet access.</td>
</tr>
<tr>
<td>Proxy host</td>
<td>Specify a value for the Proxy host. For example, 172.16.10.8 or ourproxy. If you do not know the values for your proxy server, ask your system or network administrator for the information.</td>
</tr>
<tr>
<td>Preference</td>
<td>Usage</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Proxy port</td>
<td>Specify an integer value for the <strong>Proxy port</strong>. For example, 22. If you do not know the values for your proxy server, ask your system or network administrator for the information.</td>
</tr>
<tr>
<td>Use a proxy with authentication</td>
<td>Specifies that your proxy server requires a user name and password.</td>
</tr>
<tr>
<td>Proxy username</td>
<td>Specify the proxy server user name.</td>
</tr>
<tr>
<td>Proxy password</td>
<td>Specify the proxy server password.</td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>MATLAB stores the password without encryption in your <em>matlab.prf</em> file.</td>
</tr>
<tr>
<td>Test connection</td>
<td>Ensure that your settings work.</td>
</tr>
<tr>
<td></td>
<td>If MATLAB cannot access the Internet, <strong>Failed!</strong> appears next to the button. Correct the values you entered and try again. If you still cannot connect, try using the values you used when you authenticated your MATLAB license.</td>
</tr>
<tr>
<td>System Web browser</td>
<td><strong>UNIX</strong> platforms only — excluding Macintosh</td>
</tr>
<tr>
<td></td>
<td>• <strong>Command</strong>—Specifies the system command to open the browser.</td>
</tr>
<tr>
<td></td>
<td>For example, <code>opera</code>, opens the Opera Web browser.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Options</strong>—Specifies options for the system browser.</td>
</tr>
<tr>
<td></td>
<td>For example, <code>geometry 1064x860</code> specifies the size of the window for Opera.</td>
</tr>
</tbody>
</table>

---

6. UNIX is a registered trademark of The Open Group in the United States and other countries.
Entering Commands

- “Enter Statements in Command Window” on page 3-2
- “Find Functions Using the Function Browser” on page 3-4
- “Format Output in Command Window” on page 3-9
- “Stop Execution” on page 3-12
- “Find Text in Command Window or History” on page 3-13
- “Create Shortcuts to Rerun Commands” on page 3-17
- “Set Command Window Preferences” on page 3-19
- “Set Keyboard Preferences” on page 3-21
- “Check Syntax As You Type” on page 3-23
- “Command History” on page 3-29
Enter Statements in Command Window

As you work in MATLAB, you can enter individual statements in the Command Window. For example, create a variable named a by typing this statement at the command line:

\[ a = 1 \]

MATLAB immediately adds variable a to the workspace and displays the result in the Command Window.

\[ a = 1 \]

When you do not specify an output variable, MATLAB uses the variable ans, short for answer, to store the results of your calculation.

\[ \sin(a) \]

\[ \text{ans} = 0.8415 \]

The value of \text{ans} changes with every command that returns an output value that is not assigned to a variable.

If you end a statement with a semicolon, MATLAB performs the computation, but suppresses the display of output in the Command Window.

\[ b = 2; \]

To enter multiple statements on multiple lines before running any of the statements, use \text{Shift+Enter} between statements. This action is unnecessary when you enter a paired keyword statement on multiple lines, such as \text{for} and \text{end}.

You also can enter more than one statement on the same line by separating statements. To distinguish between commands, end each one with a comma or semicolon. Commands that end with a comma display their results, while
commands that end with a semicolon do not. For example, enter the following three statements at the command line:

\[ A = \text{magic}(5), \quad B = \text{ones}(5) \times 4.7; \quad C = A./B \]

\[
A = \\
\begin{array}{cccc}
17 & 24 & 1 & 8 & 15 \\
23 & 5 & 7 & 14 & 16 \\
4 & 6 & 13 & 20 & 22 \\
10 & 12 & 19 & 21 & 3 \\
11 & 18 & 25 & 2 & 9
\end{array}
\]

\[
C = \\
\begin{array}{cccc}
3.6170 & 5.1064 & 0.2128 & 1.7021 & 3.1915 \\
4.8936 & 1.0638 & 1.4894 & 2.9787 & 3.4043 \\
0.8511 & 1.2766 & 2.7660 & 4.2553 & 4.6809 \\
2.1277 & 2.5532 & 4.0426 & 4.4681 & 0.6383 \\
2.3404 & 3.8298 & 5.3191 & 0.4255 & 1.9149
\end{array}
\]

MATLAB displays only the values of \( A \) and \( C \) in the Command Window.

To recall previous lines in the Command Window, press the up- and down-arrow keys, \( \uparrow \) and \( \downarrow \). Press the arrow keys either at an empty command line or after you type the first few characters of a command. For example, to recall the command \( b=2 \), type \( b \), and then press the up-arrow key.

You can evaluate any statement already in the Command Window. Select the statement, right-click, and then select Evaluate Selection.

In the Command Window, you also can execute only a portion of the code currently at the command prompt. To evaluate a portion of the entered code, select the code, and then press Enter.

For example, select a portion of the following code:

\[ \text{\texttt{disp('hello')}, disp('world')}} \]

hello
Find Functions Using the Function Browser

In this section...

“What Is the Function Browser?” on page 3-4
“Steps for Using the Function Browser” on page 3-4
“Interpreting Search Results in the Function Browser” on page 3-7
“Customizing the Function Browser” on page 3-7

What Is the Function Browser?
The Function Browser provides quick access to the syntax and description for a function by using the reference pages for installed MathWorks products. Using the Function Browser, you can browse or search for a MATLAB function.

Be aware that:

- You cannot use the Function Browser for blocks. Instead use the doc function or the Help browser.
- You cannot use the Function Browser to find functions you created or that other users provided. Instead, use “Finding Files and Folders” on page 6-24.

Steps for Using the Function Browser

1 Open the Function Browser by clicking the Browse for functions button, $f_x$, in the Command Window or Editor.

   The Function Browser closes when you move the pointer outside of it. To keep the Function Browser open, drag it by the top edge to a different location.

2 Find functions by browsing in the list of categories or by typing a search term.

   The following illustration shows how you browse for functions by expanding categories of interest.
The following illustration shows results when you search, for example, for `filter`.

![Illustration of function browser search results for 'filter'](image_url)
3 View more information about a function by moving the pointer over a function. A brief description for each of the syntax options displays in a yellow pop-up window.

The pop-up window automatically closes when you move your pointer to a new item in the results list. To keep the pop-up window open, drag it by the top edge to a different location.

4 Choose a function you want to use and then:

- Add a function name after the cursor in the Command Window or Editor by double-clicking the name in the Function Browser results list.
• Drag the function name from the Function Browser into any tool or application.
• Right-click the function name in the Function Browser to display other options.

Interpreting Search Results in the Function Browser

Parentheses Indicate Location of Function
For results in products other than MATLAB, the product folder appears in parentheses.

When more than one function in MATLAB has the same name, the folder for the overloaded function appears in parentheses.

For example:
• filter is in MATLAB
• filter (signal) is in the Image Processing Toolbox™
• filter (timeseries) is in the timeseries folder in MATLAB

Highlights in Search Results
In the results, the search term is highlighted in blue.

When a result does not include any blue highlighting, the matching term is not part of the name, syntax, or brief description. The term is somewhere else in the reference page.

Results for a Two-Letter Search Term
For faster performance, when you type only two letters, the Function Browser looks only for exact matches.

Customizing the Function Browser
• You can show or hide the Browse for functions button in the Command Window. On the Home tab, in the Environment section, select
Preferences > Command Window, and then select or clear Show Function Browser button.

- Change the font that the Function Browser uses by selecting Preferences > Fonts. By default, the Function Browser uses the desktop text font and the pop-up window uses the HTML proportional text font.

- You can restrict the products the Function Browser looks in. Click the product area at the bottom of the Function Browser, and then set the Selected Products options in the Help Preferences dialog box that opens.
Format Output in Command Window

Format Line Spacing in Output
By default, MATLAB displays blank lines in command output.

You can select one of two numeric display options in MATLAB.

- **loose**—Keeps the display of blank lines (default)
  ```
  >> x = [4/3 1.2345e-6]
  x =
    1.3333    0.0000
  ```

- **compact**—Suppresses the display of blank lines
  ```
  >> x = [4/3 1.2345e-6]
  x =
    1.3333    0.0000
  ```

To format the output display, do one of the following:

- On the **Home** tab, in the **Environment** section, select **Preferences > Command Window**, and then choose a **Numeric format** option.
- Use the **format** function at the command line, for example:
format loose
format compact

**Format Floating-Point Numbers**
You can change the way numbers display. By default, MATLAB uses the short format (5-digit scaled, fixed-point values).

For example, suppose you enter \( x = \frac{4}{3} \ 1.2345 \times 10^{-6} \) in the Command Window. The MATLAB output display depends on the format you selected.

<table>
<thead>
<tr>
<th>Output Display Format</th>
<th>Example Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>short (default)</td>
<td>( x = 1.3333 \ 0.0000 )</td>
</tr>
<tr>
<td>short e</td>
<td>( x = 1.3333 \times 10^0 \ 1.2345 \times 10^{-6} )</td>
</tr>
<tr>
<td>+</td>
<td>( x = + + )</td>
</tr>
</tbody>
</table>

**Note** The text display format affects only how numbers are shown, not how MATLAB computes or saves them.

**Wrap Lines of Code to Fit Window Width**
A line of code or its output can exceed the width of the Command Window, requiring you to use the horizontal scroll bar to view the entire line. To break a single line of input or output into multiple lines to fit within the current width of the Command Window:


2. Select Wrap Lines.

3. Click OK.

**Suppress Output**
To suppress code output, add a semicolon (\;\) to the end of a command. This is particularly useful when code generates large matrices.
Running the following code creates A, but does not show the resulting matrix in the Command Window:

\[
A = \text{magic}(100);
\]

**View Output by Page**

Output in the Command Window might exceed the visible portion of the window. You can view the output, one screen at a time:

1. In the Command Window, type `more on` to enable paged output.
2. Type the command that generates large output.
3. View the output:
   - Advance to the next line by pressing **Enter**.
   - Advance to the next page by pressing **Space Bar**.
   - Stop displaying the output by pressing **q**.

To disable paged output, type `more off`.

**Clear the Command Window**

If the Command Window seems cluttered, you can clear all the text (without clearing the workspace) by doing one of the following:

- On the **Home** tab, in the **Code** section, select **Clear Commands > Clear Command Window** to clear the Command Window scroll buffer.
- Use the `clc` function to clear the Command Window scroll buffer.
- Use the `home` function to clear your current view of the Command Window, without clearing the scroll buffer.

**See Also**

`clc` | `format` | `home` | `more`
Stop Execution

To stop execution, press Ctrl+C or Ctrl+Break

On Apple Macintosh platforms, you also can use Command+. (the Command key and the period key).

Ctrl+C does not always stop execution for files that run a long time, or that call built-ins or MEX-files that run a long time. If you experience this problem, include a drawnow, pause, or getframe function in your file, for example, within a large loop.

Also, Ctrl+C might be less responsive if you start MATLAB with the -nodesktop option.

**Note** For certain operations, stopping the program might generate errors in the Command Window.

**See Also** drawnow | getframe | pause
Find Text in Command Window or History

In this section...

| “Find Text in the Command Window” on page 3-13 |
| “Find Text in the Command History Window” on page 3-15 |

Find Text in the Command Window

You can search text currently in the Command Window. This includes text that is currently visible on the screen, as well as text that is in the scroll buffer.

- “Search Using Find Dialog” on page 3-13
- “Incremental Search Using Keyboard Shortcuts” on page 3-14

Search Using Find Dialog

To search for specified text in the Command Window, on the Command Window title bar, click , and then select Find. The Find dialog box opens. The search begins at the current cursor position. MATLAB finds the text you specified and highlights it.

MATLAB beeps when a search for Find Next reaches the end of the Command Window, or when a search for Find Previous reaches the top of the Command Window. If you have Wrap around selected, MATLAB continues searching after beeping.

To search for the specified text in other MATLAB desktop tools, change the selection in the Look in field.

You can increase the amount of information available in the Command Window so that more text is available for searching. Be aware that doing so requires more memory. On the Home tab, in the Environment section, select Preferences > Command Window, and then increase the setting for Number of lines in the command window scroll buffer.

Clearing the command window (for example, with the clc function), empties the scroll buffer. The cleared text is no longer available for searching. To
clear your display in the Command Window without clearing the buffer, use the `home` function.

**Incremental Search Using Keyboard Shortcuts**

This topic shows how to perform an incremental search in the Command Window.

With the incremental search feature, you do not have to leave the Command Window while you perform your search.

1 Begin an incremental search using one of the following keyboard shortcuts, depending on your operating system’s active settings file.

<table>
<thead>
<tr>
<th>Action</th>
<th>Windows Default Active Settings</th>
<th>Macintosh or Emacs Default Active Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initiate a forward incremental search.</td>
<td>Ctrl+Shift+S</td>
<td>Ctrl+S</td>
</tr>
<tr>
<td>Initiate a backward incremental search.</td>
<td>Ctrl+Shift+R</td>
<td>Ctrl+R</td>
</tr>
</tbody>
</table>

An incremental search field appears at the bottom of the Command Window. For a forward search, the text `incSearch` appears. The `F` indicates a forward search.

2 Begin typing your search term.

When you enter lowercase letters in the `Inc Search` field, incremental search looks for both lowercase and uppercase instances of the letters. For example, if you enter `b`, incremental search looks for `b` and `B`. However, if you enter uppercase letters, incremental search only looks for instances that match the case you entered.

3 Perform incremental search actions using the following keyboard shortcuts.
Find Text in Command Window or History

<table>
<thead>
<tr>
<th>Action</th>
<th>Keyboard Shortcut</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complete a partially highlighted string of characters.</td>
<td>Ctrl+W</td>
</tr>
<tr>
<td>Find the next occurrence of a string of characters.</td>
<td>Ctrl+S</td>
</tr>
<tr>
<td>Remove characters from the <strong>Inc Search</strong> field, back to the last successful search</td>
<td>Ctrl+G</td>
</tr>
</tbody>
</table>

If you search for a string that does not appear in the Command Window text, **Failing** appears in the incremental search field.

4 End incremental searching by pressing **Esc** (escape), **Enter**, or any other key that is not a character or number.

The **Inc Search** field disappears. The cursor remains at the position where the text was last found, with the search text highlighted.

**Find Text in the Command History Window**

This topic shows two methods to find text in the Command History Window.

- “Quick Search Using Keyboard Controls” on page 3-15
- “Search Partial or Whole Word, or Match Case Using Find Menu” on page 3-16

**Quick Search Using Keyboard Controls**

To quickly find entries in the Command History Window based on the first few letters or numbers in the entry:

1 Type the first few letters or numbers of the entry you want to find in the Command History window.

The Command History window searches backwards and selects the previous entry that begins with the letters you typed. A tooltip with the text: **Search history for:**:, appears at the top of the Command History window.

2 Search for additional instances by doing one of the following:
• Find the previous or next occurrence of the entry with the up and down arrow keys, respectively.

• Highlight each occurrence of the entry found, while you search for additional instances, press the Ctrl key with the up or down arrow key.

• Highlight all instances of the entry, press Ctrl+A.

Search Partial or Whole Word, or Match Case Using Find Menu
You can find text in the Command History Window that matches the case, whole word, or partial word.

On the Command History title bar, click  
, and then select Find. The search begins at the current cursor position. Find does not identify entries in collapsed nodes.
Create Shortcuts to Rerun Commands

This example shows how to create, run, edit, and organize MATLAB shortcuts. A MATLAB shortcut is an easy way to run a group of MATLAB language statements that you use regularly. For example, use a shortcut to set up your environment when you start working, or to set the same properties for figures you create.

1 On the Home tab, click New, and then select Command Shortcut.

   If the Shortcuts tab is currently on the desktop, you can also click New Shortcut in the Manage section.

2 Complete the Shortcut Editor dialog box:

   a In the Label field, enter a name for the shortcut.

      For this example, enter my_Shortcut.

   b In the Callback field, type statements you want the shortcut to run.

      You also can drag and drop statements from the Command Window, Command History Window, or a file.

      For this example, enter these statements:

      format compact
      clear
      workspace
      filebrowser
      clc

      **Tip** If Command Window prompts (>>) appear, MATLAB automatically removes them from the Callback field when you save the shortcut.

   c In the Category field, type the name of a new category or select an existing category from the drop-down list. If you leave this field blank, the shortcut appears in the General section of the toolbar.
d In the Icon field, select an icon.

e Click Save.

The shortcut icon and label appear on the toolbar. If you have more shortcuts on the toolbar than the desktop can display concurrently, use the drop-down list to access them all.

To organize and edit shortcuts, on the Shortcuts tab, in the Manage section, click Organize Shortcuts to open the Shortcuts Organizer dialog box.

3 Run a shortcut by clicking its icon on the Shortcuts tab.

All the statements in the shortcut Callback field execute as if you ran those statements from the Command Window, although they do not appear in the Command History window.
### Set Command Window Preferences

You can customize the visual display of the Command Window and command output within it.

On the **Home** tab, in the **Environment** section, select **Preferences > Command Window**, and then adjust preference options as described in the table below.

<table>
<thead>
<tr>
<th>Preference</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text display</td>
<td>Select a <strong>Numeric format</strong> option to specify the output format of numeric values in the Command Window. For details, see “Format Floating-Point Numbers” on page 3-10.</td>
</tr>
<tr>
<td></td>
<td>Select a <strong>Numeric display</strong> option to specify whether blank lines appear in Command Window output. To suppress blank lines, select compact. To display blank lines, select loose.</td>
</tr>
<tr>
<td>Display</td>
<td>Select <strong>Wrap lines</strong> to make each line of input or output in the Command Window break into multiple lines to fit within the current width of the Command Window. For details, see “Wrap Lines of Code to Fit Window Width” on page 3-10.</td>
</tr>
<tr>
<td></td>
<td>Select <strong>Set matrix display width to eighty columns</strong> to limit the width of matrix output.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong>  If you also select Wrap lines, and the width of the Command Window is less than 80 characters, each row of 80 characters of matrix output wraps to fit within the width of the Command Window.</td>
</tr>
<tr>
<td></td>
<td>Select <strong>Show getting started message bar</strong> to display the Command Window message bar that provides links to introductory information.</td>
</tr>
<tr>
<td>Preference</td>
<td>Usage</td>
</tr>
<tr>
<td>------------------------------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Preference Usage</td>
<td>Select <strong>Show function browser button</strong> to display the Function Browser button ( \text{\footnotesize \textregistered} ) to the left of the prompt in the Command Window. You can use the Function Browser to search for MATLAB functions. Select <strong>Suggest corrections for mistyped functions and variables</strong> to display suggestions in the Command Window. If you enter an undefined function or variable name, MATLAB displays: <strong>Did you mean:</strong> followed by a suggested command at the command line. You can press <strong>Enter</strong> to execute that command, or <strong>Esc</strong> to delete the suggestion. Number of lines in command window scroll buffer specifies the maximum number of lines displayed in the Command Window. A larger scroll buffer provides a larger base for search features, but requires more memory. By default, the scroll buffer is set to 5,000 lines. The scroll buffer size does not impact the number of lines you can recall. By default, you can use the up arrow key ↑ to recall all lines shown in the Command History window, regardless of how many lines you can see in the Command Window.</td>
</tr>
<tr>
<td>Tab key</td>
<td><strong>Tab size</strong> specifies the number of spaces assigned to the tab key. <strong>Note</strong> This setting does not apply if you have enabled tab completion. To change tab completion settings, on the <strong>Home</strong> tab, select <strong>Preferences &gt; Keyboard</strong>.</td>
</tr>
</tbody>
</table>
Set Keyboard Preferences

Keyboard preferences enable you to set tab completion, function hints, and delimiter matching in the Command Window and Editor.

To set Keyboard Preferences, on the Home tab, in the Environment section, select Preferences > Keyboard, and then adjust preference options as described in this table.

<table>
<thead>
<tr>
<th>Preference</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tab completion</td>
<td>Select the tool or tools in which you want the Tab key to complete names known to MATLAB after you type the first few letters of the name. For details, see “Tab Completion” on page 3-25.</td>
</tr>
<tr>
<td></td>
<td>Select Tab key narrow completions to have MATLAB continue to reduce the list of possible names for completion as you type each additional character and press the Tab key.</td>
</tr>
<tr>
<td>Function hints</td>
<td>Specify the selected tool or tools that you want to display syntax function hints. When enabled, if you type a function name with an opening parenthesis, and then pause, a tooltip opens showing the basic syntax for the function. For example:</td>
</tr>
<tr>
<td></td>
<td><img src="image.png" alt="Example Image" /> For details, see “Function Syntax Hints” on page 3-27.</td>
</tr>
</tbody>
</table>
Entering Commands

Preference | Usage
--- | ---
Delimiter Matching | Specify when and if MATLAB alerts you to matched and mismatched delimiters. Delimiters include parentheses, brackets, braces, and, in the Editor only, paired keywords.

If you select **Match while typing**, MATLAB alerts you to matched and mismatched delimiters as you type.

If you select **Match on arrow key**, MATLAB alerts you to matched and mismatched delimiters when you move the cursor over a delimiter using an arrow key.

For details, see “Delimiter Matching” on page 3-24.

Select one of these **Show match with** options to specify how MATLAB indicates matching delimiters:

- **Balance** — The corresponding delimiter highlights briefly (default).
- **Underline** — Both delimiters in the pair display underlines briefly.
- **Highlight** — Both delimiters in the pair highlight briefly.

Select one of these **Show mismatch with** options to specify how MATLAB indicates mismatched delimiters:

- **Beep** — MATLAB beeps (default).
- **Strikethrough** — The delimiter you type appears crossed out briefly.
- **None** — There is no alert.
Check Syntax As You Type

<table>
<thead>
<tr>
<th>In this section...</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Syntax Highlighting” on page 3-23</td>
</tr>
<tr>
<td>“Delimiter Matching” on page 3-24</td>
</tr>
<tr>
<td>“Tab Completion” on page 3-25</td>
</tr>
<tr>
<td>“Function Syntax Hints” on page 3-27</td>
</tr>
</tbody>
</table>

Syntax Highlighting
To help you identify MATLAB elements, some entries appear in different colors in the Command Window. This is known as *syntax highlighting*. By default:

- Keywords are blue.
- Strings are purple.
- Unterminated strings are maroon.
- Comments are green.

```matlab
if A > B
    'greater'
elseif A < B
    'less'
end
```

Except for errors, output in the Command Window does *not* appear with syntax highlighting.

When you paste or drag a selection from the Editor to another application, such as Microsoft Word, the pasted text maintains the syntax highlighting colors and font characteristics from the Editor. MATLAB software pastes the selection to the Clipboard in RTF format, which many Microsoft Windows and Macintosh applications support.
You can change syntax highlighting preferences. On the **Home** tab, in the **Environment** section, select **Preferences > Editor/Debugger > Languages**.

### Delimiter Matching

MATLAB indicates matched and mismatched delimiters, such as parentheses, brackets, and braces, to help you avoid syntax errors. MATLAB also indicates paired language keywords, such as `for`, `if`, `while`, `else`, and `end` statements.

By default, MATLAB indicates matched and mismatched delimiters and paired language keywords as follows:

- Type a closing delimiter—MATLAB briefly highlights the corresponding opening delimiter.
- Type more closing delimiters than opening delimiters—MATLAB beeps.
- Use the arrow keys to move the cursor over one delimiter—MATLAB briefly underlines both delimiters in a pair. If no corresponding delimiter exists, MATLAB puts a strike line through the unmatched delimiter.

If a matching delimiter exists, but it is not visible on the screen, a pop-up window appears and shows the line containing the matching delimiter. Click in the pop-up window to go to that line.

```matlab
>> a = ['first', 'third', 'fourth', 'fifth', 'sixth']
```

You can change delimiter matching indicators, and when and if they appear. On the **Home** tab, in the **Environment** section, select **Preferences > Keyboard**.
Tab Completion
MATLAB can help you avoid typographical errors by completing the names of functions, models, MATLAB objects, files, folders, variables, structures, and Handle Graphics property names.

To complete names in the Command Window, type the first few characters of the name you want to complete, and then press the Tab key.

If MATLAB presents a list of possible matches, use the arrow keys to select the name you want, and then press the Tab key.

In addition, you can:

- Clear the list without selecting anything, by pressing the Esc (escape) key.
- Narrow a long list before making a selection, by adding additional characters to your original term.
- Complete parts of a name that uses dot notation by adding a dot, and then pressing the Tab key.
- Complete the names and values of Handle Graphics properties. Begin typing the first part of a property, and then press the Tab key. Type a comma after each property.

For MATLAB to complete a file or folder name, it must be on the search path or in the current folder. Variables and properties must be in the current workspace.

In the Editor, MATLAB completes:

- Nested functions only when they are available at the current location of the cursor.
• Names of variables defined in the active document. The variable must be valid at the current location of the cursor (that is, already defined).

In the Editor, MATLAB does not complete:

• Field names of structure arrays defined only within the active file.
• Method or property names for objects defined only within the active file.

**Note** To add spaces within statements using the Tab key in the Editor, first add a space, and then press Tab. Otherwise, when tab completion is enabled, MATLAB attempts to complete a name.

Tab completion is enabled by default. To change this setting, on the Home tab, in the Environment section, select Preferences > Keyboard.

**Example of Name Completion**
This example shows how to complete the name for the `containers.Map.keys` method.

1 In the Command Window, type `help cont`, and then press Tab.

MATLAB displays a list of selections.
2 Select `containers`, and then press Tab.

The Command Window displays `help containers`.

3 At the command prompt, add a dot after `containers`, and then press Tab.

The Command Window displays:

```
help containers.Map
```

4 At the command prompt, add a dot after `Map`, and then press Tab.

MATLAB displays a new list.

```
delete
findobj
findprop
isempty
isKey
isValid
keys
KeyType
```

5 Scroll down the list, select `keys`, and then press the Tab key.


**Function Syntax Hints**

As you enter a function in the Command Window or Editor, syntax hints open in a pop-up window to display allowable input arguments for a function.

Function hints appear for both MATLAB installed functions and functions you create. The syntax hints for MATLAB functions comes from the documentation. The syntax for functions you create comes from the function definition statement (first executable line) in the MATLAB program file. That file must be on the search path or in the current folder.
To use function syntax hints, type a function name with an opening parenthesis, and then pause. A tooltip opens showing the basic syntax for the function.

```
>> m = size( )
size(X)
size(X,dim)
```

You can type a variable for any argument that appears in blue. Enter your variable names, and not the argument names shown in the window.

The displayed syntax options change, based on the argument you just entered.

Some function names are overloaded. That is, there are methods with the same name as a function that support different types of inputs. Overloaded methods require that you pass an object as the first input. When you specify the object name, the syntax hints update to reflect the associated method, as shown.

```
>> m = matfile('topography.mat');
>> size( m, )
size(matlab.io.MatFile object,variable)
size(matlab.io.MatFile object,variable,dim)
```

Function syntax hints are suggestions only. Some allowable arguments might not appear, or could be in black text when they should be blue.

Function hints are enabled by default. To change this setting, on the Home tab, in the Environment section, select Preferences > Keyboard, and then set the options for Function hints.
**Command History**

<table>
<thead>
<tr>
<th>In this section...</th>
</tr>
</thead>
<tbody>
<tr>
<td>“What Is the Command History?” on page 3-29</td>
</tr>
<tr>
<td>“Using Command History Commands” on page 3-29</td>
</tr>
<tr>
<td>“Changing the Command History Date Format” on page 3-31</td>
</tr>
<tr>
<td>“Command History Preferences” on page 3-31</td>
</tr>
</tbody>
</table>

**What Is the Command History?**

The Command History window displays a log of statements you ran in the current and previous MATLAB sessions. The time and date for each session appear at the top of the statements listed for that session, in your operating system’s short date format. All entries remain until you delete them, or until the command history file exceeds its maximum size of 200,000 bytes. When the file exceeds its maximum size, MATLAB automatically deletes the oldest entries.

MATLAB saves statements that run in the Command Window to the to the history file, `history.m`. This includes statements you run using the **Evaluate Selection** item on context menus in tools such as the Editor, Command History, and Help browser. By default, MATLAB automatically saves the command history file after each command. The history file does not include every action taken in MATLAB. For example, modifications of values in the Variable Editor are not included in the Command History.

**Using Command History Commands**

You can select entries in the Command History window, and then perform the following actions for the selected entries.
<table>
<thead>
<tr>
<th>Action</th>
<th>How to Perform the Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create a script from a statement or statements.</td>
<td>Select an entry or entries, and then right-click and select <strong>Create Script</strong> from the context menu. The Editor opens a new file that contains the statements you selected from the Command History window.</td>
</tr>
<tr>
<td><strong>Run Command History commands in the Command Window.</strong></td>
<td>Do one of the following:</td>
</tr>
<tr>
<td></td>
<td>• Double-click an entry or entries in the Command History window.</td>
</tr>
<tr>
<td></td>
<td>• Right-click an entry and select <strong>Evaluate Selection</strong> from the context menu.</td>
</tr>
<tr>
<td></td>
<td>• Select an entry and press <strong>Enter or Return</strong>.</td>
</tr>
<tr>
<td>Copy statements to another window.</td>
<td>Do one of the following:</td>
</tr>
<tr>
<td></td>
<td>• Select an entry or entries, and then select <strong>Copy</strong> from the context menu. Paste the selection into an open file in the Editor or any application.</td>
</tr>
<tr>
<td></td>
<td>• Drag the selection from the Command History window to an open file or another application.</td>
</tr>
<tr>
<td>Create a shortcut from a statement or statements.</td>
<td>Do one of the following:</td>
</tr>
<tr>
<td></td>
<td>• Select an entry or entries, and then right-click and select <strong>Create Shortcut</strong> from the context menu.</td>
</tr>
<tr>
<td></td>
<td>• Drag the selection to the desktop Toolstrip. The <strong>Add Shortcut</strong> dialog box opens and the selected statements appear in the <strong>Callback</strong> field.</td>
</tr>
<tr>
<td>Delete Entries</td>
<td><strong>Note</strong> You cannot recall entries you delete from the Command History window.</td>
</tr>
<tr>
<td></td>
<td>Select the entries to delete, and then right-click and select <strong>Delete Selection</strong> from the context menu, or press the <strong>Delete</strong> key.</td>
</tr>
<tr>
<td>Action</td>
<td>How to Perform the Action</td>
</tr>
<tr>
<td>--------</td>
<td>---------------------------</td>
</tr>
<tr>
<td></td>
<td>To select all entries for a MATLAB session, select the timestamp for that session.</td>
</tr>
<tr>
<td></td>
<td>To delete all entries, right-click in the Command History window, and then select <strong>Clear Command History</strong> from the context menu.</td>
</tr>
</tbody>
</table>

### Changing the Command History Date Format
MATLAB uses your operating system’s short date format to display dates in the Command History window. To change the date format, for instance from MM/DD/YYYY to DD/MM/YYYY:

1. Change the short date format for your operating system as described in its documentation.

2. Right-click in the Command History window and select **Clear Command History**.

**Note** Clearing the command history deletes all entries from the Command History window. You can no longer recall those entries in the Command Window.

### Command History Preferences
You can exclude statements from the command history and specify how often to save the file in which the command history is stored, `history.m`. MATLAB uses the command history file for both the Command History window and statement recall in the Command Window.

**Note** When you exclude statements from the command history file, you cannot recall them in the Command Window, nor can you view them in the Command History window.
To set Command History preferences, on the **Home** tab, in the **Environment** section, select **Preferences > Command History**, and then adjust preference options as described in the table below.

<table>
<thead>
<tr>
<th>Preference</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Settings</strong></td>
<td><strong>Save exit/quit commands</strong> saves exit and quit commands in the command history.</td>
</tr>
<tr>
<td></td>
<td><strong>Save consecutive duplicate commands</strong> saves consecutive executions of the same statement in the command history.</td>
</tr>
<tr>
<td></td>
<td>• With this option selected, if you run <code>magic(5)</code> two times in a row, both entries for <code>magic(5)</code> remain in the command history.</td>
</tr>
<tr>
<td></td>
<td>• With this option cleared, the command history retains only one entry for <code>magic(5)</code>. If you then run <code>magic(10)</code>, the command history retains both entries.</td>
</tr>
<tr>
<td><strong>Saving</strong></td>
<td><strong>Save history file on quit</strong> saves the command history file only when you end a MATLAB session. If the session ends abnormally, such as due to a power failure, then MATLAB does not save the history file for that session.</td>
</tr>
<tr>
<td></td>
<td><strong>Save after n commands</strong> saves the command history file after MATLAB adds n statements to it. This option reduces the loss of entries to the saved history in case of an abnormal termination of the MATLAB session.</td>
</tr>
<tr>
<td></td>
<td><strong>Don’t save history file</strong> is useful when multiple users share the same machine. It prevents each user from viewing the statements others have run.</td>
</tr>
<tr>
<td></td>
<td>Any entries already in the command history remain unless you first delete entries from the Command History window.</td>
</tr>
</tbody>
</table>
Help and Product Information

- “Ways to Get Function Help” on page 4-2
- “Run Examples” on page 4-3
- “Search Syntax and Tips” on page 4-5
- “Bookmark and Share Page Locations” on page 4-7
- “Contact Technical Support” on page 4-9
- “Demos Are Now Called Examples” on page 4-11
- “Help Preferences” on page 4-13
- “Japanese Documentation” on page 4-15
- “Information About your Installation” on page 4-16
Ways to Get Function Help

Each MATLAB function has supporting documentation that includes examples and describes the function inputs, outputs, and calling syntax. This table describes ways to access that documentation.

<table>
<thead>
<tr>
<th>Type of Help</th>
<th>How to Access</th>
<th>Example or Icon</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference page in Help browser</td>
<td>Use the doc command. — or — Select a function name in the Editor, Command Window, or Help browser; right-click; and then select Help on Selection.</td>
<td>doc mean</td>
</tr>
<tr>
<td>Function syntax hints in Command Window</td>
<td>Pause after you type an open parentheses for function inputs.</td>
<td>mean(</td>
</tr>
<tr>
<td>Abbreviated help text in Command Window</td>
<td>Use the help command.</td>
<td>help mean</td>
</tr>
<tr>
<td>Function browser in Command Window</td>
<td>Click the function icon to the left of the command prompt.</td>
<td></td>
</tr>
<tr>
<td>Complete documentation in Help browser</td>
<td>Click the Help button on the quick access toolbar or on the Home tab. — or — Enter search terms in the Search Documentation box.</td>
<td></td>
</tr>
</tbody>
</table>

See Also  

- `doc`

Related Examples

- “Run Examples” on page 4-3
Run Examples

In this section...

“Code Excerpts” on page 4-3
“Example Scripts” on page 4-3

Code Excerpts

Run code from any page in the documentation by selecting the code, right-clicking, and then selecting Evaluate Selection.

Note On Macintosh systems, press Shift+F7, which copies code to the Command Window for evaluation.

Example Scripts

Featured example scripts include a button at the top of the Help browser for opening the script in the Editor.

Square Wave from Sine Waves

This example shows (graphically) how the Fourier series expansion for a square wave is made up of a sum of odd harmonics.
In the Editor, there are two ways to run the code:

- Run one section of the script at a time and view the incremental results. Select the first section, and then step through the script by clicking Run and Advance, 📚.

- Run the entire script by clicking Run, ⏯️.

See Also demo | echodemo

Related Examples • “Run Code Sections”
Search Syntax and Tips

Find keywords in the documentation by entering text in the Search box on the Desktop or in the Help browser.

The search engine ignores common, insignificant words such as *a*, *an*, *the*, and *of*, unless they are part of an exact phrase in quotation marks. It also ignores capitalization, punctuation, and special characters such as +. To find a symbol or special character:

- Search for the word instead of the symbol or character, such as *plus* instead of +.
- View the documentation on Operators and the Symbol Reference.
- Search the PDF documentation, available from the documentation home page.

Searches can include the following operators:

" "  Exact phrase

Example: "plot tools" finds pages that contain *plot tools*, in that sequence, with no words between them.

*  Wildcard

Requires at least two nonwildcard characters, and cannot appear at the start of a keyword or in an exact phrase.

Example: *plot* finds *plot*, *plot3*, and *plotting*.

OR  Boolean OR

Example: *plot* OR *graph* finds pages with either *plot* or *graph*.
NOT  Boolean NOT
Example: "plot tools" NOT "time series" finds pages with plot
tools but excludes pages with time series.

AND  Boolean AND
Implied when no operator is present between keywords.
Example: plot AND tools is equivalent to "plot" "tools".

The Help browser search evaluates NOT operators first, OR operators second,
and AND operators last. For example,

"plotting tool" OR "plot tools" NOT "time series" AND workspace
finds pages that contain either plotting tool or plot tools and contain
workspace, but do not contain time series.

The search engine searches the following text in the documentation:

- Documentation — Text and code shown in the Help browser
- GUI-based examples — Help comments in the program file
- Videos — Title
Bookmark and Share Page Locations

In this section...

“Bookmark Favorite Pages” on page 4-7
“View Page Locations” on page 4-8

Bookmark Favorite Pages
In MATLAB, bookmarks are called favorites. Add, find, and organize favorites by clicking the Favorites button in the Help browser, 🌟.

When you add a favorite, do not change the Callback. MATLAB requires special values to create a shortcut that opens the page in the Help browser. In addition, if you want the bookmark to appear in your list of favorites, keep the Category set to Help Browser Favorites, as shown.

Note You cannot migrate favorites that you save in one MATLAB release to a new release.
View Page Locations

To identify the location of a page in the Help browser to share with someone else, right-click within the page, and then select **Get Page Address**.

**Note**  This feature is not available on Macintosh systems.

The Help Page Location dialog box provides two ways to access the page:

- A web command to run from the command line that opens the page from the installed documentation. This command is subject to change between releases, so it is not always accurate for someone running a different version of MATLAB.

- A URL for the page corresponding to your product version at the MathWorks Web site. This documentation is available to anyone, even if they do not have MathWorks products. However, to access archived documentation from previous releases, you must log in with a MathWorks Account.

**Note**  If you are running a prerelease version, the URL is invalid because the documentation does not yet exist on the Web site.
Contact Technical Support

This example shows how to contact MathWorks Technical Support to report a bug or request help. This procedure requires Internet access.

1. Click Help > Submit Support Request.

2. When requested, log in using your MathWorks Account email address and password. If you do not have a MathWorks Account, create one.

3. Provide information to help technical support reproduce your issue, such as a description of the steps you followed or a code excerpt. Optionally, you can attach up to five files to your request, where each file is no larger than 3 MB. To submit files larger than 3 MB, upload them to the MathWorks FTP site.
4 Specify the product that is related to the issue.

5 Submit the request.

**External Web Sites**

- How do I access the MathWorks FTP site?
- MathWorks Support Page
**Demos Are Now Called Examples**

Starting in R2012b, executable scripts and videos that were previously called *demos* are now called *examples*. There are two ways to access these examples from the Help browser:

- At the top of the product landing page, click **Examples**.

- On any documentation page, click the Table of Contents button, and then select **Examples**.

You can filter documentation search results to display only examples.
For more information about changes to the Help browser, see the R2012b MATLAB Release Notes.
### Help Preferences

To set Help preferences:

1. On the **Home** tab, in the **Environment** section, click **Preferences > Help**.
2. Adjust the preference options as described in the table.

<table>
<thead>
<tr>
<th>Preference</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Documentation Location</strong></td>
<td>Specify whether to view the documentation provided with your installed products or the documentation on the Web at <code>http://www.mathworks.com/help</code>. Viewing the Web documentation requires an Internet connection and a MathWorks Account.</td>
</tr>
<tr>
<td></td>
<td>If your preference is set to view Web documentation, but your Internet connection becomes unavailable, MATLAB changes the preference to view the installed documentation. You can reset the preference after your connection is restored.</td>
</tr>
<tr>
<td><strong>Selected Products</strong></td>
<td>Select the products to include for viewing and searching documentation in the Help browser or Function browser.</td>
</tr>
<tr>
<td></td>
<td>If your <strong>Documentation Location</strong> is set to view documentation on the Web, then you can select <strong>Show products that are not installed</strong> to select and access documentation for all MathWorks products, even if you do not have those products installed.</td>
</tr>
<tr>
<td></td>
<td>When the Help browser is already open, changes to this preference apply only to new Help browser tabs.</td>
</tr>
<tr>
<td>Preference</td>
<td>Usage</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Quick Help Display</strong></td>
<td>Specify whether help links display content in the Help browser or in a small window. This preference applies to content that you access using:</td>
</tr>
<tr>
<td></td>
<td>• Help on Selection in context menus or <strong>F1</strong></td>
</tr>
<tr>
<td></td>
<td>• Function hints or the Function browser</td>
</tr>
<tr>
<td></td>
<td>• Links in error messages</td>
</tr>
<tr>
<td></td>
<td>Links to help from the Current Folder browser or the Help browser always open in the Help browser.</td>
</tr>
<tr>
<td><strong>Language</strong> (selected</td>
<td>Specify whether documentation in the Help browser and context-sensitive help should appear in English. Installed non-English documentation is not always current.</td>
</tr>
<tr>
<td>non-English systems only)</td>
<td></td>
</tr>
</tbody>
</table>

**Concepts**

- “Japanese Documentation” on page 4-15
Japanese Documentation

Many MathWorks products provide versions of the documentation translated from English to Japanese. However, the translated documentation usually is not available until about 2 months after the initial release of a new product version.

The new version of most products installs the translated documentation from the previous version and the English documentation for the current version. To view the English documentation, set the Help Language preference to English. To set Help preferences, access the Environment section on the Home tab, and click Preferences > Help.

The Language preference is available when the system locale is Japanese and the translated documentation is installed. The preference changes the language only in the Help browser and context-sensitive help. If the documentation for a product is not translated, the Help browser displays the English documentation.

When the translated documentation is available, you can view it by setting your Help Documentation Location preference to view documentation on the Web. Alternatively, download it from the MathWorks Web site at http://www.mathworks.co.jp/help.

For information about documentation in other languages, contact your MathWorks sales and service office.
# Information About your Installation

MATLAB software can tell you what products are installed, their versions, and other information about your license and platform. This information is important to have in the event you contact technical support.

<table>
<thead>
<tr>
<th>Type of Information You Want</th>
<th>To Get the Information</th>
</tr>
</thead>
</table>
| Version and license for Installed product | From the product, select **Help > About**. Or use functions:  
• *license* — for the license number  
• *ver* — for version numbers for MATLAB and libraries  
• *version* — for version numbers for MathWorks products |
| MATLAB platform | In MATLAB, select **Help > About MATLAB**. The About MATLAB dialog box shows 32-bit or 64-bit. |
| *arch* value used for the *mex* function | In MATLAB, select **Help > About MATLAB**. The About MATLAB dialog box shows the *arch* value, for example *win32*. Or use the *computer* function. |
| Passcodes and licenses | From any desktop tool, select **Help > Web Resources > MathWorks Account**. |
Workspace Browser and Variable Editor

- “What Is the MATLAB Workspace?” on page 5-2
- “View, Edit, and Copy Variables” on page 5-3
- “Keyboard Shortcuts for Navigating Variable Elements” on page 5-8
- “Save, Load, and Delete Workspace Variables” on page 5-9
- “Statistical Calculations in the Workspace Browser” on page 5-12
- “Set Workspace and Variable Preferences” on page 5-14
What Is the MATLAB Workspace?

The MATLAB workspace consists of the variables you create and store in memory during a MATLAB session. You add variables to the workspace by using functions, running MATLAB code, and loading saved workspaces. For example, if you run these statements:

\[
\begin{align*}
A &= \text{magic}(4); \\
R &= \text{randn}(3,4,5);
\end{align*}
\]

the workspace includes two variables, A and R.

The Workspace browser displays the variables in your workspace. From the Workspace browser, you can select variables to view, modify, or plot.

To open the Workspace browser if it is not currently visible, do either of the following:

- Type `workspace` at the Command Window prompt.

By default, the Workspace browser displays the base workspace. You also can view function workspaces if MATLAB is in debug mode. For more information, see “Debugging Process and Features” and the `dbstack` and `evalin` functions.

You can display additional columns, such as size (dimensions) and size in bytes in the Workspace browser. On the Workspace browser title bar, click ▽, and then click Choose Columns.
View, Edit, and Copy Variables

<table>
<thead>
<tr>
<th>In this section...</th>
</tr>
</thead>
<tbody>
<tr>
<td>“View and Edit Variables” on page 5-3</td>
</tr>
<tr>
<td>“Copy, Paste, and Rename Variables” on page 5-6</td>
</tr>
</tbody>
</table>

View and Edit Variables

This table shows how to view variables and their values.

<table>
<thead>
<tr>
<th>Action</th>
<th>Procedure</th>
</tr>
</thead>
<tbody>
<tr>
<td>List current workspace variables</td>
<td>Use the <code>who</code> function. To also list information about size and class, use the <code>whos</code> function.</td>
</tr>
<tr>
<td>Display variable contents in the Command Window</td>
<td>Type the variable name at the Command prompt.</td>
</tr>
<tr>
<td>Open a variable in the Variables editor</td>
<td>Do one of the following:</td>
</tr>
<tr>
<td></td>
<td>• Use the <code>openvar</code> function. For example, to open the variable <code>A</code>, type <code>openvar('A')</code></td>
</tr>
<tr>
<td></td>
<td>• In the Workspace browser, double-click a variable name. Some toolboxes allow you to double-click an object in the Workspace browser to open a viewer or other tool appropriate for that object. For details, see the toolbox documentation for that object type.</td>
</tr>
<tr>
<td></td>
<td>There are special attributes for <code>timeseries</code> objects; for more information, see “Viewing Time Series Objects”</td>
</tr>
</tbody>
</table>
After opening a variable, you can view and edit its values graphically. This example shows how to open a variable, and then navigate and edit its contents graphically. Note that you cannot edit elements or subsets of multidimensional arrays in the Variables editor.

1 Create a cell array, `C`.

```matlab
A = magic(4);
C = {A A A};
```

2 In the Workspace browser, open variable `C` to view its contents, by doing one of the following:

- Use the `openvar` function.

  ```matlab
  openvar('C')
  ```

- In the Workspace browser, double-click the variable name `C`.

The variable `C` opens in the Variables editor.

Lock icons, which can appear during debugging, denote protected and private properties of an object, indicating you do not have `get` access to those values outside class methods.
3 Double-click element C{1,1} to view the contents of that cell. The element opens in a new document within the Variables editor.

4 Edit the value of an array element by clicking the element, and then typing a new value. Press Enter, or click another element.
Increase the size of the array in C{1,1} by entering a value in element (5,5). Empty elements in numeric arrays fill with zeros. In a cell array, empty elements fill with empty arrays.

5 Remove a row by clicking in its row header. Right-click, and then select Delete Row.

6 Cut the elements of the first column by selecting column header. Right-click, and then select Cut. The value of each element you cut becomes 0 if numeric, or [ ] if a cell array. The cut values move to the clipboard.

7 Paste the cut elements into another column by selecting the element where you want the insertion to begin. Right-click, and then select Paste.

8 Change how numbers display when editing variables. On the View tab, in the Format section, select a number display format.

9 On the View tab, use the Go Up button to return to the cell array or structure.

Changes you make in the Variables editor are automatically saved in the workspace.

Changes you make to variables via the Command Window or other operations automatically update the information for those variables in the Variables editor.

### Copy, Paste, and Rename Variables

<table>
<thead>
<tr>
<th>Action</th>
<th>Procedure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Copy variables to clipboard</td>
<td>In the Workspace browser, select the variables, right-click and then select Copy. Then, you can paste the names, for example, into the Command Window or an external application. Multiple variables are comma separated.</td>
</tr>
<tr>
<td>Paste cells from Microsoft Excel spreadsheet</td>
<td>In a variable open in the Variables editor, right-click, and then select Paste from Excel.</td>
</tr>
</tbody>
</table>
If you cut and paste values from the Variables editor into text files or other applications, you can change the character that delimits decimals in the data that is exported. You might do this, for instance, if you provide data to a locale that uses a character other than the period (\textperiodcentered). To change the delimiter character, specify a \textbf{Decimal separator for exporting numeric data via system clipboard} in the “Variables Preferences” on page 5-15.
# Keyboard Shortcuts for Navigating Variable Elements

Use the following keyboard shortcuts to move among variable elements in the Variables editor. You cannot modify these keyboard shortcuts.

<table>
<thead>
<tr>
<th>Action</th>
<th>Keyboard Shortcut</th>
</tr>
</thead>
<tbody>
<tr>
<td>Commit changes to an element and move to next element.</td>
<td><strong>Enter</strong></td>
</tr>
<tr>
<td>“Variables Preferences” on page 5-15 enable you to specify what the next element is (the default is down).</td>
<td></td>
</tr>
<tr>
<td>Move right.</td>
<td><strong>Tab</strong></td>
</tr>
<tr>
<td>Within a selection, also moves from the last column to the first column in the next row.</td>
<td></td>
</tr>
<tr>
<td>Move in opposite direction of <strong>Enter</strong> or <strong>Tab</strong>.</td>
<td><strong>Shift+Enter</strong> or <strong>Shift+Tab</strong></td>
</tr>
<tr>
<td>Move up ( m ) rows, where ( m ) is the number of visible rows.</td>
<td><strong>Page Up</strong></td>
</tr>
<tr>
<td>Move down ( m ) rows, where ( m ) is the number of visible rows.</td>
<td><strong>Page Down</strong></td>
</tr>
<tr>
<td>Move to column 1.</td>
<td><strong>Home</strong></td>
</tr>
<tr>
<td>Move to row 1, column 1.</td>
<td><strong>Ctrl+Home</strong></td>
</tr>
<tr>
<td>Edit current element, positioning cursor at the end of the element.</td>
<td><strong>F2</strong> (<strong>Ctrl+U</strong> on Apple Macintosh platforms)</td>
</tr>
</tbody>
</table>
Save, Load, and Delete Workspace Variables

The workspace is not maintained across sessions of MATLAB. When you quit MATLAB, the workspace clears. However, you can save any or all of the variables in the current workspace to a MAT-file (.mat). You can load MAT-files at a later time during the current MATLAB session, or during another session, if you want to reuse the workspace variables.

The following table describes how to save, view, and load workspaces.

<table>
<thead>
<tr>
<th>Action</th>
<th>Desktop Workflow</th>
<th>Programmatic Workflow</th>
</tr>
</thead>
</table>
| Save all workspace variables| On the Home tab, in the Variable section, click Save Workspace.                   | Use the save function followed by the file name you want to save to. For example, to save all current workspace variables to the file june10.mat:  
 save('june10')                                                               |
| Save selected variables     | Select the variables in the Workspace browser, right-click, and then select Save As. | Use the save function. For example, to save variables A and R to the file june10.mat:  
 save('june10', 'A', 'R')                                                    |
| Load a MAT-file             | Select the MAT-file in the Current Folder browser, right-click, and then select Load. | Use the load function. For example, to load all workspace variables from the file june10.mat:  
 load('june10')                                                               |
<table>
<thead>
<tr>
<th>Action</th>
<th>Desktop Workflow</th>
<th>Programmatic Workflow</th>
</tr>
</thead>
<tbody>
<tr>
<td>Load selected</td>
<td>Do one of the following:</td>
<td>Use the <code>load</code> function. For example, the load variables A and R from the file june10.mat:</td>
</tr>
<tr>
<td>variables</td>
<td>• In the Current Folder browser, select the MAT-file containing the variables.</td>
<td><code>load('june10','A','R')</code></td>
</tr>
<tr>
<td></td>
<td>Drag specific variables from the Details panel at the bottom of the Current</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Folder browser to the Workspace browser.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• On the <strong>Home</strong> tab, in the <strong>Variable</strong> section, click <strong>Import Data</strong>.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Select the MAT-file you want to load and click <strong>Open</strong>.</td>
<td></td>
</tr>
</tbody>
</table>

**Note** If any variables being loaded have the same names as variables in the current workspace, MATLAB asks if you want to replace the values in the current workspace with the values in the MAT-file, or cancel.
<table>
<thead>
<tr>
<th>Action</th>
<th>Desktop Workflow</th>
<th>Programmatic Workflow</th>
</tr>
</thead>
<tbody>
<tr>
<td>View a MAT-file without loading it into the workspace</td>
<td>Select the MAT-file in the Current Folder browser. The contents display in the Details panel at the bottom of the Current Folder browser.</td>
<td>Use the <code>whos</code> function with the <code>-file</code> option.</td>
</tr>
</tbody>
</table>
| Delete all variables in the workspace      | Select the variables in the Workspace browser, and then press the Delete key on your keyboard. Or right-click, and then select **Delete**. | Use the `clear` function to clear variables from the workspace. For example, to clear the variables A and R from the workspace:  
```  
clear A R  ``` |
| Delete selected variables                  | On the **Home** tab, in the **Variable** section, click **Clear Workspace**.       | Use the `clearvars` function with the `-except` option to keep specified variables and clear all other variables. |
Statistical Calculations in the Workspace Browser

In this section...

| “Improve Workspace Browser Performance during Statistical Calculations” on page 5-12 |
| “Include or Exclude NaN Values in Statistical Calculations” on page 5-12 |

Improve Workspace Browser Performance during Statistical Calculations

For each variable or object, the Workspace browser displays statistics such as the **Min**, **Max**, and **Mean** calculations, when relevant. MATLAB performs these calculations using the `min`, `max`, and `mean` functions, and updates the results automatically.

If you show statistical columns in the Workspace browser, and you work with very large arrays, you might experience performance issues when the data changes as MATLAB updates the statistical results. To improve performance, consider one or both of the following:

- **Show only the statistics of interest to you.**
  
  On the Workspace browser title bar, click ` `, and then select **Choose Columns**. Clear the statistics you do not want MATLAB to calculate.

- **Exclude large arrays from statistical calculations.**
  
  On the **Home** tab, in the **Environment** section, click **Preferences > Workspace**, and then use the arrow buttons to change the value of the maximum array size for which you want the Workspace browser to perform statistical calculations. Any variable exceeding the maximum array size reports `<Too many elements>` in Workspace browser statistics columns instead of statistical results.

Include or Exclude NaN Values in Statistical Calculations

If your data includes NaNs, you can specify that the Workspace browser statistical calculations consider or ignore the NaNs. On the **Home** tab, in the
Environment section, click Preferences > Workspace, and then select one of the following:

- **Use NaNs when calculating statistics**
  If a variable includes a NaN, and you select this option, the values for Min, Max, Var and some other statistics will appear as NaN. However, Mode, for example, shows a numeric result.

- **Ignore NaNs when calculating statistics**
  If a variable includes a NaN, and you select this option, numeric results appear for most statistics including Min and Max. Var, however, is still appears as NaN.
Set Workspace and Variable Preferences

In this section...

“Workspace Browser Preferences” on page 5-14
“Variables Preferences” on page 5-15

Workspace Browser Preferences

Workspace browser preferences enable you to restrict the size of arrays on which you perform calculations and to specify if you want those calculations to include or ignore NaNs.

To open Workspace browser preferences, on the Home tab, in the Environment section, click Preferences > Workspace.

<table>
<thead>
<tr>
<th>Preference</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>n element and smaller arrays show statistics</em></td>
<td>Limit the size of arrays for which the Workspace browser displays statistics to improve performance when MATLAB updates the statistical results in the Workspace browser. For more information, see “Statistical Calculations in the Workspace Browser” on page 5-12.</td>
</tr>
<tr>
<td>Handling NaN values in calculations</td>
<td>Specify whether NaN values be included or excluded from calculations for the statistics displayed in the Workspace browser.</td>
</tr>
<tr>
<td>• Use NaNs when calculating statistics</td>
<td></td>
</tr>
<tr>
<td>• Ignore NaNs when calculating statistics</td>
<td></td>
</tr>
</tbody>
</table>
Variables Preferences

When working in the Variables editor, Variables preferences enable you to specify the array formatting, cursor movement, and the decimal separator for exporting data using the system clipboard.

To open Variables preferences, on the Home tab, in the Environment section, click Preferences > Variables.

<table>
<thead>
<tr>
<th>Preference</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Format</strong></td>
<td>Select an option from the Default array format to specify the default array output format of numeric values displayed in the Variables editor. This format preference affects only how numbers display, not how MATLAB computes or saves them. For information on formatting options, see the reference page for the format function.</td>
</tr>
<tr>
<td><strong>Editing</strong></td>
<td>Specify where the cursor moves to after you type an element, and then press Enter:</td>
</tr>
<tr>
<td></td>
<td>• To keep the cursor in the element where you typed, clear the Move selection after Enter check box.</td>
</tr>
<tr>
<td></td>
<td>• To move the cursor to another element, select the Move selection after Enter check box. In the Direction field, specify how you want the cursor to move.</td>
</tr>
<tr>
<td><strong>International number handling</strong></td>
<td>In the Decimal separator for exporting numeric data via system clipboard field, specify the decimal separator for numbers you cut or copy from the Variables editor when you paste them into text files or other applications. This preference has no effect on numeric data copied from and pasted into MATLAB. Within MATLAB, decimal separators are always periods.</td>
</tr>
</tbody>
</table>
Managing Files in MATLAB

- “Understanding File Locations in MATLAB” on page 6-2
- “Working with Files and Folders” on page 6-11
- “Finding Files and Folders” on page 6-24
- “Creating, Opening, Changing, and Deleting Files and Folders” on page 6-31
- “Comparing Files and Folders” on page 6-47
- “Making Files and Folders Accessible to MATLAB” on page 6-65
- “Using the MATLAB Search Path” on page 6-71
Understanding File Locations in MATLAB

<table>
<thead>
<tr>
<th>In this section...</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Important MATLAB Folders” on page 6-2</td>
</tr>
<tr>
<td>“Path Names in MATLAB” on page 6-5</td>
</tr>
</tbody>
</table>

**Important MATLAB Folders**
When you work with files and folders, be aware of key locations that MATLAB uses.

**The Current Folder**
The *current folder* is a reference location that MATLAB uses to find files. This folder is sometimes referred to as the *current directory*, *current working folder*, or *present working directory*. It is *not* the same location as the operating system current folder.

You can always load files and execute scripts and functions that are in the current folder, even if that folder is not currently on the MATLAB search path. Functions in the current folder take precedence over functions with the same file name that reside anywhere on the search path.

**Viewing and Changing the Current Folder.** You can view and change the current folder using various desktop tools and functions, as described in the following table. To specify the current folder programmatically when MATLAB starts, see “Startup Folder for the MATLAB Program” on page 1-12.
<table>
<thead>
<tr>
<th>To:</th>
<th>Do this:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identify the current folder</td>
<td>Use one of the following:</td>
</tr>
<tr>
<td></td>
<td>- The current folder toolbar — if the full path is not visible, click ▼.</td>
</tr>
<tr>
<td></td>
<td>- The pwd or cd function.</td>
</tr>
<tr>
<td>Change the current folder to one you specify</td>
<td>Do one of the following:</td>
</tr>
<tr>
<td></td>
<td>- In the current folder toolbar, type or browse to a different folder. You also can click an arrow that appears between portions of the path, and then choose a drive or subfolder from the drop-down list.</td>
</tr>
<tr>
<td></td>
<td>- Use the cd function.</td>
</tr>
<tr>
<td>Change the current folder to a recently used folder</td>
<td>From the current folder toolbar, click the down arrow, and then select a folder from the history.</td>
</tr>
<tr>
<td>Change the current folder to an active document’s folder</td>
<td>Right-click the document tab in the Editor, and then select <strong>Change Current Folder to folder</strong>.</td>
</tr>
<tr>
<td>Make a subfolder the current folder</td>
<td>In the Current Folder browser, right-click the subfolder, and then select <strong>Open</strong>.</td>
</tr>
<tr>
<td>Copy the current folder as a string</td>
<td>Click an empty area on the right edge of the Current Folder address field, right-click, and then select <strong>Copy</strong>.</td>
</tr>
<tr>
<td>Get help using the current folder toolbar</td>
<td>Right-click an empty area on the address bar, and then select <strong>Help Using Address Field</strong></td>
</tr>
</tbody>
</table>

**matlabroot**

`matlabroot` is the folder where you installed MATLAB. The location differs for each installation of MATLAB. Determine its location by running the `matlabroot` function. When you start MATLAB, your current folder can be `matlabroot`, but in practice it is usually a different folder.
The Startup Folder
Each time you start MATLAB, your current folder is always the same. This location is called the startup folder. The operating system commands that runs MATLAB specifies the location of the startup folder. You can configure MATLAB to make your initial current folder a different location. For more information, see “Startup Folder for the MATLAB Program” on page 1-12.

Locations of MathWorks Products
Files and folders for products provided by MathWorks are in `matlabroot/toolbox`. The files and folders under `matlabroot` are important to your installation. In particular:

- Do not store your personal files and folders in `matlabroot/toolbox`.
- Do not change files, folders, and subfolders in `matlabroot/toolbox`. The exception is the `pathdef.m` file, which you can update and save in its default location, `matlabroot/toolbox/local`.

To improve performance, at the beginning of each session, MATLAB loads and caches in memory the locations of files in `matlabroot/toolbox`. If you make changes to files and folders in `matlabroot/toolbox`, running functions can produce unexpected results or generate warnings, that are related to the toolbox cache. See “Toolbox Path Caching in the MATLAB Program” on page 1-23.

To see a list of all toolbox folder names supplied with MathWorks products, run:

```matlab
dir(fullfile(matlabroot, '/toolbox'))
```

Locations for Storing Your Files
For your convenience, MATLAB provides a folder called MATLAB to store your files. At startup, MATLAB adds the folder to the search path, allowing MATLAB to access the files stored there.

The location of the `userpath` MATLAB folder varies by platform and system configuration. To determine the location, run the `userpath` function.
On Microsoft Windows platforms, MATLAB sets the current folder to `userpath` at startup. On other platforms, you instruct MATLAB differently to set the current folder to `userpath` at startup. For more information, see “Startup Folder for the MATLAB Program” on page 1-12.

If you create subfolders within the MATLAB folder, make the new subfolders accessible to MATLAB.

If you store files in locations other than the MATLAB folder:

- Make the files accessible to MATLAB by adding their folders to the search path.
- Do not store the files in the folders provided for MathWorks products.

**Path Names in MATLAB**

A path name specifies file locations, for example, `C:\work\my_data` (on Microsoft Windows platforms) or `/usr/work/my_data` (on Linux or Apple Mac platforms). Path name specifications differ, depending on the platform on which you are running MATLAB. When you work with files and folders, be aware of how MATLAB uses path names and the restrictions it places on them.

**Specifying Path Names on Apple Mac Platforms**

When you specify path names on Mac platforms, do not use accent characters. If path names include such characters, for instance umlauts or circumflexes, the Current Folder browser and MATLAB cannot recognize the path. In addition, attempts to save a file to such a path results in unpredictable behavior.

**Specifying File Separator Characters, / and \**

The file separator character is the symbol that distinguishes one folder level from another in a path name.

A forward slash (`/`) is a valid separator on any platform. A backward slash (`\`) is valid only on Microsoft Windows platforms.

In the full path to a folder, the final slash is optional.
Type `filesep` in the Command Window to determine the correct file separator character to use when working with files programmatically.

**Specifying Absolute and Relative Path Names**

MATLAB always accepts *absolute* path names (also called *full* path names), such as `I:/Documents/My_Files`. An absolute path name can start with any of the following:

- UNC path `\\` string
- Drive letter, on Microsoft Windows platforms, such as `C:\`
- `/` character on Linux platforms

Some MATLAB functions also support relative path names. The reference page for a function specifies the valid types of path name. Unless otherwise noted, the path name is relative to the current folder. For example:

- `/myfolder` refers to the `myfolder` folder in the current folder and `myfile.m` refers to the `myfile.m` file in the current folder.
- `../myfolder/myfile.m` refers to the `myfile.m` file in the `myfolder` folder, where `myfolder` is at same level as the current folder. Each repetition of `../` at the beginning moves up an additional folder level.

**Tip** If multiple documents are open and docked in the Editor, you can copy the absolute path of any of these documents to the clipboard. This is useful if you need to specify the absolute path in another MATLAB tool or an external application. Right-click the document tab, and then select **Copy Full Path to Clipboard**

**Case Sensitivity of File Names**

How MATLAB handles file names with respect to case depends on a number of factors. If you are unsure of how MATLAB will handle your specific case, it is best to specify path and case precisely when specifying a file name. The sections that follow describe how case affects common MATLAB operations.
Case Sensitivity of File Names When Calling a Function. You call function files by specifying the file name without the file extension. MATLAB assumes you want a case-insensitive match if it cannot find a case-sensitive match on the MATLAB search path (regardless of the operating system on which MATLAB is running).

Furthermore, if multiple files with the same name, but different extensions exist in the same folder, then MATLAB searches among the files in the folder in this precedence order:

- MEX-files
- MDL (Simulink model) files
- P-code files
- MATLAB code files (those with a .m extension)

For details on precedence, see .

For example, suppose myfile.m is on the search path, but MYFILE.M is not. If you type MYFILE at the MATLAB command prompt, then MATLAB runs myfile.m, but warns you that there is a case mismatch and advises you that this warning will become an error in a future release.

If myfile (a MEX-file) and MYFILE.m are on the search path, and you type MYFILE at the command prompt, MATLAB runs MYFILE.M, even if myfile is higher on the search path.

To see which file MATLAB will use without running that file, use which with the all option. For example, which myfile.m all.

Case Sensitivity of File Names When You Load a MAT-File. When you call load and specify a file without an extension, MATLAB searches for a MAT-file. Case-sensitivity depends on the operating system where MATLAB is running, as follows:

- Linux

  If you attempt to load MYFILE, and MYFILE.MAT is anywhere on the MATLAB search path, then MATLAB loads MYFILE.MAT. This is true, even
if myfile.mat, and myfile.m are higher than MYFILE.MAT on the MATLAB search path.

If you attempt to load MYFILE, and MYFILE.MAT is not on the MATLAB search path, then MATLAB returns the message, Unable to read file MYFILE.MAT: No such file or directory. This is true even if myfile.mat is on the MATLAB search path.

- Windows

  If you attempt to load MYFILE and myfile.mat is higher on the search path than MYFILE.MAT, then MATLAB loads myfile.mat without warning you that there is a case mismatch.

**Case Sensitivity of File Names When You Save a MAT-File.** When you call save and specify a MAT-file without an extension, MATLAB saves the file to the current folder. Case-sensitivity depends on the operating system where MATLAB is running, as follows:

- Linux

  MATLAB saves the file using the case you specify. Two files with the same name, but different cases can exist in the same folder.

- Windows

  Because the Windows operating system considers two files with the same name to be the same file (regardless of case), you cannot have two files with the same name in the same folder. If you save MYFILE, and myfile.mat already exists in the current folder, then MYFILE.MAT replaces myfile.mat without warning. If you save myfile, and MYFILE.mat already exists in the current folder, the contents of myfile.mat replace the contents of MYFILE.mat, but the name remains MYFILE.mat.

**Maximum Length of Path Names in MATLAB**

The maximum length allowed for a path name depends on your platform.

For example, on Microsoft Windows platforms:

- The maximum length is known as MAX_PATH.
- You cannot use an absolute path name that exceeds 260 characters.
• For a relative path name, you might need to use fewer than 260 characters. When the Windows operating system processes a relative path name, it can produce a longer absolute path name, possibly exceeding the maximum length.

If you get unexpected results when working with long path names, use absolute instead of relative path names. Alternatively, use shorter names for folders and files.

**Constructing Path Names on Different Platforms**

Use `fullfile` to construct path names in statements that work on any platform. This function is particularly useful when you provide code to someone using it on a platform different from your own. The `ismac`, `ispc`, and `isunix` functions identify the platform you are currently using.

**Including Spaces in Path Names**

When a function argument is a file or path name, and the name includes spaces, use the function syntax. For example:

```matlab
delete('temp file.m') % Function syntax works for a file name containing a space
```

The command syntax does not work. For example:

```matlab
delete temp file.m % Command syntax does NOT work for a file name containing a space
```

**Partial Path Names in MATLAB**

A partial path name is the last portion of a full path name for a location on the MATLAB search path.

Some functions accept partial path names. The reference page for a function typically specifies the valid types of path names.

Examples of partial path names are: `matfun/trace`, `private/cancel`, and `demos/clown.mat`.

Use a partial path name to:

• Specify a location more conveniently than by using the full path name.
• Specify a location independent of where MATLAB is installed.

• Locate a function in a specific toolbox when multiple toolboxes contain functions with that name. For example, to get help for the `set` function in the Database Toolbox™ product, type:

```
help database/set
```

• Locate method files. For example, to get help for the time series object `plot` method type:

```
help timeseries/plot
```

Specifying the at sign character (@) in method folder names is optional.

• Locate private and method files, which sometimes are hidden.

Be sure to specify enough of the path name to make the partial path name unique.

**See Also**

• “Slash and Backslash — / \”

• `ismac`, `ispc`, and `isunix` functions, for MATLAB statements that require different path names for different platforms

• “Private Functions”.
Working with Files and Folders

You can view information about folders from the MATLAB Desktop like you can from operating system windows. The principal Desktop tool for working with files and folders is the Current Folder browser. Like other Desktop components, you can dock the Current Folder browser or open it as a separate window. The Current Folder browser displays details about files in your current folder and within the hierarchy of the folders it contains. You can modify the kinds of information it displays to suit your needs, for example by reordering or deleting specific columns of information.

The Current Folder browser:

- Always displays your current folder, as well as its subfolders.
- Lets you access operating system file management features from within MATLAB.
- Is similar to file browsers provided with operating systems, but also includes features unique to MATLAB. For example, you can add folders to the search path from the Current Folder browser.

The following sections explain what you can do with the Current Folder browser and how to use it.
Opening the Current Folder Browser

To open the Current Folder browser, on the Home tab, in the Environment section, click Layout. Then, under Show, select Current Folder.

The Current Folder browser shows the contents of the current folder.

Double-clicking a subfolder displays its contents, and makes that folder the current folder.

If there are an exceptionally large number of entries to display, this message might appear in the Current Folder browser:

There are too many entries to display.
Click the search button in the address bar to reduce the number of entries.

When you click the search button , the address bar becomes a search field.
Preferences for the Current Folder Browser
You can set preferences for aspects of the Current Folder browser. On the Home tab, in the Environment section, click Preferences > Current Folder. The preferences are:

- History — The number of recently used folders maintained in the Current Folder browser drop-down list.
- Refresh — How frequently the Current Folder browser updates to reflect changes to files made outside of MATLAB.
- Path indication — Controls the appearance of folders and files that are inaccessible to MATLAB, and whether to display tooltips describing their status.
- Toolbar — Provides a link to the Toolbars preferences. Those preferences enable you to adjust the toolbar layout and controls for Desktop tools, including the Current Folder browser.
- Hidden files — Controls whether the Current Folder browser displays hidden files and folders.

This preference not on available Microsoft Windows platforms.

Tip For information on changing the date format in the Current Folder browser, see “Customizing the Column Display” on page 6-17
Refreshing the List of Files
When files and folders are created, deleted, or changed outside of MATLAB, the Current Folder browser automatically reflects the changes. When you access files on a network, frequent refreshing of the Current Folder browser can slow performance in MATLAB. If this seems to be a problem, try improving the performance by changing how frequently refreshing occurs using the Current Folder Refresh preference:

1 On the Home tab, in the Environment section, click Preferences > Current Folder.

   By default, the Auto-refresh view from file system option is on, with an update time of 3 seconds. Every 3 seconds, the Current Folder browser checks for and reflects changes made from programs and tools other than MATLAB.

2 Try to improve responsiveness by either:
   • Increasing the Number of seconds between auto-refresh.
   • Clearing the Auto-refresh view from file system check box to turn off the feature.

3 Click OK.

To manually refresh the view at any time:

1 Right-click in the list area of the Current Folder browser.

2 Select Refresh from the context menu.

Viewing Hidden Files and Folders
The operating system, by default, hides certain files and folders from system file browsers and file-listing commands. The Current Folder browser can display hidden files and folders. You control this in different ways on different operating systems.

On Microsoft Windows platforms, the Current Folder browser follows the Windows preference for showing hidden files. To set or change the Windows preference, access the Folder Options, and then select an option for viewing Hidden files and folders.
On other platforms, specify the behavior using Current Folder preferences:

1 On the **Home** tab, in the **Environment** section, click **Preferences > Current Folder**.

2 Specify the setting for **Hidden files and folders**.

**Controlling the Appearance of Files Inaccessible to MATLAB**
MATLAB cannot access files if they are not on the search path or, in some cases, if they are in a private folder. By default, the Current Folder browser dims the display of files and folders inaccessible to MATLAB. Furthermore, if you hover over a dimmed file, a tooltip provides information on why that file is inaccessible. If you disable this feature, the Current Folder browser displays all files and folders as undimmed and provides no tooltips regarding their availability to MATLAB.

To customize this feature:
1 On the **Home** tab, in the **Environment** section, click **Preferences > Current Folder**.

2 Select the **Indicate inaccessible files** check box to enable this feature; deselect it and skip to step 5 to disable this feature.

3 Move the **Text and icon transparency** slider to adjust the level of dimming.

View the region below the slider to preview how your choice will affect the appearance of files in the Current Folder browser.

4 Select **Show tooltip explaining why files are inaccessible** to enable tooltips; deselect it to disable them.

5 Click **OK**.

For more information, see “Private Functions” and “What Is the Search Path?” on page 6-71.

### Using the Current Folder Browser

- “Customizing the Column Display” on page 6-17
- “Viewing File Descriptions” on page 6-18
- “Viewing File Details Without Opening Files” on page 6-18
- “Viewing Help for a MATLAB Program File” on page 6-21
- “Sorting and Grouping Files and Folders” on page 6-22

The Current Folder browser lists details about files and folders in columns, beneath file and folder names, and in the details panel. Any file that is modified in the Editor, but not yet saved has an asterisk (*) next to it in the Current Folder browser. The browser displays columns for **Size**, **Date Modified**, **Type**, and **Description**. You can modify the information it displays. You also use this tool to perform operations on files and folders, such as moving, compressing, renaming, creating, and deleting them.
Note  Do not use accented characters, such as à, é, ñ, or ü, in folder names. The Current Folder browser cannot locate folders containing such characters or save files to them.

Customizing the Column Display
You can show and hide columns, change their order, and adjust the date format in the Current Folder browser.

To Specify the Columns to Display.

1  Right-click on any column header, or in the Current Folder browser, click , and then select Show.

2  Select the columns to show. Clear the columns to hide.

In addition, consider:

- Hiding the Type column if the icon column provides enough information about the type.
- Sorting or grouping by a column without showing the column.

In the Current Folder browser, click , and then select Sort By or Group By. Then, choose the method by which you want to sort or group columns.

To Modify Columns.

- To change the order, drag a column header to a new position.
- To change the width, drag the edge of the column header.

To Change the Date Format. MATLAB uses your operating system’s short date format to display dates in the Current Folder browser and the Command History window. To change the date format, for instance from MM/DD/YYYY to DD/MM/YYYY, (where MM is the numerical value for the month, DD is the numerical value for the day, and YYYY is the numerical value for the year):

1  Change the short date format for your operating system. For instructions, see your operating system documentation.
2 Refresh the date display by either restarting MATLAB or doing the following:

- In the Current Folder browser, right-click, and then choose **Refresh** from the context menu.
  
  Dates refresh and use the new format.

- In the Command History window, right click, and then choose **Clear Command History** from the context menu.
  
  The window clears. MATLAB specifies new dates in the window using the new format.

**Viewing File Descriptions**

To show or hide descriptions in the Current Folder browser, click , and then select **Show > Description**.

Descriptions appear in gray text beneath the name of the file or folder. When the Current Folder browser window is wide enough, descriptions display on the same line as file names. The Current Folder browser shows descriptions only for files and folders that are relevant to products from MathWorks. How the Current Folder browser gets the description depends on the type of item:

- **MATLAB program files** — The description is the first line of the help comments, known as the H1 line.

- **Simulink Models** — The description is from the Description pane of the Model Properties dialog box. Use the Current Folder browser to view model descriptions without starting the Simulink software.

- **Folders** — The description is the first comment line of the Contents.m file for the folder.

To provide descriptions for your own files and folders, see “Add Help for Your Program”.

**Viewing File Details Without Opening Files**

Display file details without opening a file by selecting the file, and then clicking the up arrow button on the lower right corner of the Current Folder browser. The details panel expands.
File and Folder Details. When you select a file or folder, the details panel displays more information about that file or folder, if possible. For example, if you select a MATLAB code file, the details panel shows the main functions or local functions that the file contains.

Image File Details. When you select a JPEG, JPG, BMP, WBMP, PNG, or GIF image, the details panel displays a thumbnail of the image and lists its width and height in pixels. To open the Import Wizard, double-click the thumbnail.
**Viewing Unsaved File Changes.** When you select a file that is currently open in the Editor and that contains unsaved changes, an asterisk (*) appears after that file name. The Current Folder browser columns reflect the content of the unsaved file. For instance, if you open a file, change it from a script to a function, and modify the H1 line, then the icon, type name, and description update in the Current Folder browser.

The preview in the details panel also reflects the unsaved file content, not the content on disk. For instance, in the following example, `PropertyTwo` exists in the modified `MyClass.m` file, but not the `MyClass.m` file on disk.
Viewing and Going to Elements within a MATLAB Program File. The details panel lists these elements when you selected a file with a `.m` extension:

- Local functions
- Cells
- Properties
- Methods

To open the file in the Editor, scroll to the start of the element in the details panel and double-click the element.

Viewing and Loading MAT-File Variables. Use the details panel to view the name, class, and value of all variables in the selected MAT-file. To load a variable into the workspace, select it in the details panel and drag it to the Workspace browser. The folder containing the MAT-file does not need to be on the search path for you to load it in this way.

Viewing Help for a MATLAB Program File

From the Current Folder browser, you can view help for a file that has a `.m` extension and is in the current folder or in a folder on the search path:

1. Right-click the file.
2 Select **View Help** from the context menu.

The reference page, if it exists, opens in the Help browser. Otherwise, help comments from the beginning of the file, if any exist, display in the Help browser.

**Sorting and Grouping Files and Folders**

Organize, find, and manage the files and folders you use with MATLAB by sorting and grouping items.

By default, sorting is by **Name** and grouping is off.

You can sort and then group, or group and then sort.

Regardless of the sorting and grouping options selected, the Current Folder browser lists folders and files separately.

**Tip** To view only files of a certain type (for example, files having a `.`m extension) use a simple search. See “Simple Search for File and Folder Names in the Current Folder Browser” on page 6-24.

**Sorting Items.** To change the order of items listed, sort by column:

1 In the Current Folder browser, click ☐, and then select **Sort By**.
2 Select the name of the column to sort by.

Alternatively, click the column header by which you want to sort. Click it again to reverse the direction of sorting.

**Grouping Items.** To see related items listed together, group them:

1 In the Current Folder browser, click ☐, and then select **Group By**.
2 Select **Type**, **Size**, or **Date Modified**.

Each group has a label. To hide the items in a group, click the collapse button (−) next to the label.
To turn off grouping, click in the Current Folder browser, and then select Group By > Stop Grouping.
Finding Files and Folders

<table>
<thead>
<tr>
<th>In this section...</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Finding Files and Folders by Name in the Current Folder” on page 6-24</td>
</tr>
<tr>
<td>“Simple Search for File and Folder Names in the Current Folder Browser” on page 6-24</td>
</tr>
<tr>
<td>“Advanced Search for Files — Find Files Tool” on page 6-25</td>
</tr>
<tr>
<td>“Locating a File or Folder in the Operating System Browser” on page 6-29</td>
</tr>
<tr>
<td>“Finding Files and Folders Using Functions” on page 6-30</td>
</tr>
<tr>
<td>“Additional Ways to Find Files” on page 6-30</td>
</tr>
</tbody>
</table>

Finding Files and Folders by Name in the Current Folder

In the Current Folder browser, use the typeahead feature to find a file or folder by name in the current folder:

1. Position the pointer in the list of files and folders in the current folder.

2. Type the first characters of the name you want to find.

   As you type, the Current Folder browser searches downward from the top of the window, looking through all expanded folders. It selects the first entry in the current folder whose name begins with the characters you typed.

   *Typeahead* and *find as you type* are other names for this feature.

Simple Search for File and Folder Names in the Current Folder Browser

To search for names that contain a specified series of characters in the current folder and subfolders:

1. In the current folder toolbar, change the current folder to be the one you want to search.
2 Click the search button in the current folder toolbar.

3 Type the absolute path name or begin typing a file name. The asterisk character (*) is a wildcard.

   If you type a partial path name, such as `matlab\toolbox`, MATLAB regards it as a file name.

4 Press Enter:

   - If you typed a path name, that path becomes the current folder.
   - If you typed a file name, MATLAB displays all files within the current folder (including its subfolders) that match that file name.

5 Continue filtering the list by doing either of the following:

   - Typing additional characters
   - Removing characters you already typed

   For example, to shows only file names that begin with `coll` and have a `.m` extension, type `coll*.m`

6 Clear the results and show all items in the current folder by pressing the `Esc` key.

*Instant search and filtering* are other names for this feature.

**Advanced Search for Files — Find Files Tool**

To look for a specified string in file names and within files located in multiple folders, in the **File** section of the **Home** tab, click **Find Files**, which opens the Find Files tool. The following sections provide details on using the tool.

- “Steps for Using the Find Files Tool” on page 6-26
- “Opening Files from the Results List” on page 6-27
- “Accessing Previous Results” on page 6-28
- “Skipping File Types” on page 6-28
## Steps for Using the Find Files Tool

1. On the **Home** tab, in the **File** section, click **Find Files**.

2. Search for file names containing a specified string by typing the string in the **Find files named** field.

   Ignore irrelevant characters in the string by using an asterisk (*) as the wildcard character. For example, type `coll*` to search for file names that start with `coll`.

3. Search for a specified string in the content of files by typing the string in the **Find files containing text** field.

   For example, search for `plot`. Alternatively, select text in the Command Window or Editor and that text appears in the field.

   - For partial word searching in file contents, select **Contains text** under the **More options** Search type.
   - Find an exact full-string match by selecting **Matches whole word**.

4. Specify file types to search for by selecting one of the options listed in the table.

| One file type | For **Include only file type(s)**, select the file type you are looking for.  
<table>
<thead>
<tr>
<th></th>
<th>For example, select <code>*.m</code> to limit the search to MATLAB program files.</th>
</tr>
</thead>
</table>
| All file types| **a** For **Include only file type(s)**, select **All files (*)**.  
|               | **b** Clear the **Skip file type(s)** check box, under **More options**. |
| Other variations| **a** For **Include only file type(s)**, select **All files (*)**.  
|               | **b** Select the **Skip file type(s)** check box, under **More options**.  
|               | **c** Select **Edit** to specify the file types. |
5 Specify the folders to search, using one of the **Look in** options:

- Select an option listed.
- Enter the full path for one or more folders. Separate each path by a semicolon (;).
- Include subfolders by selecting the **Include subdirectories** check box.

6 Further restrict the search using **More options**. For example, use the **Skip files over** option. It ignores large files that could take a long time to look through.

7 Perform the search by clicking **Find**.

The Find Files tool presents the search results in the right pane of the dialog box, with a summary at the bottom. For text searches, results include the line number and line of code.

8 Customize the display of results:

- To see file locations, select **Show full pathnames**.
- To sort results by a column, click the column heading. For example, click **Line** to sort results by line number.

### Opening Files from the Results List

1 Select the file to open. To select multiple files:

- Click to the left of an icon and drag up or down to select contiguous items
- **Shift**+click to select contiguous items
- **Ctrl**+click to select non-contiguous items

2 Right-click and select one of the **Open** options from the context menu.

For details about the **Open** options, see “Opening and Running Files” on page 6-40.
Accessing Previous Results
View the results of a previous search by selecting its tab at the bottom of the results pane. The Find Files tool shows up to 10 tabs for previous search results while the tool is open. File Files does not maintain the results after you close the tool.

Skipping File Types
Use the Find Files tool to look in all file types except file types you specify:

1. For Include only file type(s), select All files (*).
2. Specify the file types you want the search to ignore:
   a. Select the Skip file type(s) check box.
   b. Click Edit.
3. In the resulting Edit Skipped File Extension dialog box, specify which file types to look in and which to ignore:
   a. Ignore a file type by selecting its State check box.
   b. Look for a file type by clearing its State check box.
4. Add any file types not listed that you want to skip or look for:
   a. Enter the file extension in the field at the top of the dialog box.
   b. Click Add.
      
      The file type appears in the list.
   c. Verify that the State check box has the setting you want.
      
      The example at the end of this procedure shows the `.scc` file type added.
5. Reduce the size of the list by removing any file extensions irrelevant to your search:
   a. Select the name of the extension.
   b. Click Remove.
6. Click OK to accept your changes.
The Edit Skipped File Extensions dialog box closes.

When you use the Find Files tool, search ignores the selected file types after making the changes.

**Locating a File or Folder in the Operating System Browser**

To go to a file or folder location in Windows Explorer or Apple Mac Finder, do one of the following:

- In the Current Folder browser, right-click the file or folder, and then select **Show in Explorer** or **Show in Finder**.
- In the Current Folder browser, right-click in white space, and then select **Open Current Folder in Explorer** or **Open Current Folder in Finder**.
• In the Editor, right-click a document tab, and then select **Show in** **Explorer** or **Show in Finder**.

Document tabs appear in the Editor only when multiple documents are open and docked in the Editor.

The Windows Explorer or Mac Finder opens to the folder containing the selected item.

### Finding Files and Folders Using Functions

<table>
<thead>
<tr>
<th>To...</th>
<th>Use This Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>List files and folders in the current folder or in subfolders on the search path</td>
<td><code>dir</code></td>
</tr>
<tr>
<td>Determine if a variable, function, or folder exists.</td>
<td><code>exist</code></td>
</tr>
<tr>
<td>Search for the specified string in the first line of help in a MATLAB program file</td>
<td><code>lookfor</code></td>
</tr>
<tr>
<td>See files and folders that are relevant to MATLAB</td>
<td><code>what</code></td>
</tr>
<tr>
<td>See the full path to a file</td>
<td><code>which</code></td>
</tr>
</tbody>
</table>

### Additional Ways to Find Files

• “Find Functions Using the Function Browser” on page 3-4
• “Search Syntax and Tips” on page 4-5
Creating, Opening, Changing, and Deleting Files and Folders

In this section...

- “Creating New Files and Folders” on page 6-31
- “Copying, Renaming, and Deleting Files and Folders” on page 6-36
- “Opening and Running Files” on page 6-40
- “Running External Commands, Scripts, and Programs” on page 6-43

Creating New Files and Folders

You can add files and subfolders to your current folder with the Current Folder browser or by typing commands.

- “Creating Files and Folders with the Current Folder Browser” on page 6-31
- “Creating and Updating MAT-Files with the Current Folder Browser” on page 6-32
- “Creating and Managing Zip File Archives” on page 6-33
- “Creating Files and Folders Using Functions” on page 6-36

Creating Files and Folders with the Current Folder Browser

1 In the Current Folder browser, navigate to the folder where you want to create a file or folder.

   For guidance on where to create files, see “Locations for Storing Your Files” on page 6-4.

2 Right-click in white space, and then select one of the following from the context menu:
   - New Folder.
     MATLAB creates and selects a folder named New Folder.
   - New File > file-type,
For *file-type* you can choose: *Script, Function, Class, Enumeration, Model* (if Simulink is installed), or *Zip File*. Function, class, and enumeration files that you create this way contain template information representing the fundamental elements for the file (such as function arguments).

MATLAB creates and selects a new file named *untitled* with the appropriate extension.

3 Replace the selected name by typing a new name.

File names must start with a letter, and can contain letters, digits, or underscores.

4 Press *Enter*.

**Creating and Updating MAT-Files with the Current Folder Browser**

To create or update a MAT-file using variables in the workspace:

1 In the Current Folder browser, change the current folder to the folder where you want to save the variables. See “Locations for Storing Your Files” on page 6-4.

2 In the Workspace browser, select a variable to save. Hold down the *Ctrl* key and click any other variable names you want to include in the MAT-file.

3 Drag the selected variables from the Workspace browser to the Current Folder browser.

4 Drop the variables in the Current Folder browser:

   • Create a MAT-file by dropping the variables onto any empty location in the Current Folder browser. Then name the file.

   • Update an existing MAT-file by dropping the variables onto the file name. MATLAB warns you when the MAT-file contains variables of the same name. To update the existing variables, click *Continue*. Otherwise, click *Cancel*. 
You can suppress the warning. On the **Home** tab, in the **Environment** section, click **Preferences > General > Confirmation Dialogs**. Then, clear the preference, **Confirm when overwriting variables in MAT-files**.

See also “Opening Files and Importing Data Using the Current Folder Browser” on page 6-40.

### Creating and Managing Zip File Archives

To back up files, conserve file storage space, or to forward collections of files to other people, create archives using zip files. You can create, view, and adjust the contents of a zip file from within the Current Folder browser, as described in the sections that follow.

#### Viewing the Contents of Zip Files.

To view the contents of a zip file without extracting any files it contains, click the associated + (expand) button in the Current Folder browser. This feature is helpful when you want to:

- Confirm the contents of a newly created zip file
- View the contents of a zip file before extracting files
- Selectively open certain items from a zip file

By default, files within a zip file appear dimmed to indicate that they are not on the MATLAB path.

**Note** Archives created outside of MATLAB can be encrypted or password-protected. You cannot add files to, or extract files from, protected archives from within MATLAB.

### Creating Zip Archives.

You can either create an empty archive, or select files, folders, or both to create an initial archive. In either case, you can add more files later.

1. Do one of the following in the Current Folder browser:
   - Create an empty zip file:
Right-click white space, and then select **New File > Zip File**.

- Create a populated zip file from selected files, folders, or both:
  
  Select the folders and files you want to archive, right-click, and then select **Create Zip File**.

In either case, MATLAB creates an archive with a default name of `Untitledn.zip`, where `n` is an integer.

2 Type over the default file name to specify a descriptive name, for example `listmaster_export.zip`, as shown here.

![Image of file browser showing a zip file]

See also “Adding Files to a Zip Archive” on page 6-35.

**Extracting Files from Zip Files.** To extract a single file from within a zip file in the Current Folder browser, do one of the following:

- Copy a file name and paste it into a folder in the Current Folder browser.
- Drag the file into a folder in the Current Folder browser.

MATLAB extracts the file and saves it to the folder where you dragged or pasted it.

To extract all the files from a zip file, do one of the following:

- Double-click the zip file in the Current Folder browser.
Right-click the zip file, and then select **Extract**.

MATLAB extracts the entire contents of the zip file into a folder having the same name as the zip file.

Because MATLAB creates a folder when extracting files, none of the extracted files can overwrite files that have the same name. If you attempt to overwrite a folder with the same name when extracting, MATLAB prompts you to determine what you want to do.

**Adding Files to a Zip Archive.** To add files and folders to a zip file archive in the Current Folder browser, do one of the following:

- Select, and then drag the file that you want to add onto the archive.
- Copy the file that you want to add to the archive. Then, select the archive to which you want to add the file and paste the file into the archive.

If the archive contains a file or folder with the same name as the one you are adding, a MATLAB dialog box opens. The dialog box asks if you want to replace the existing file in the archive.

**Comparing the Contents of a Zip Archive to Unzipped Files and Folders.** To determine differences between archived and unarchived files, use the Comparison Tool from within the Current Folder browser as you would for any other files and folders.

For instance:

- Right-click a zip archive, and then from the context menu select **Compare Against** and specify the folder to which you want to compare the contents of the zip archive.
- Expand a zip archive, right-click a file within it, and then from the context menu select **Compare Against**. Specify the file to which you want to compare the archived file.

For details, see “Comparing Files and Folders” on page 6-47.
Creating Files and Folders Using Functions
As an alternative to using the Current Folder browser to create files and folders, you can run functions in the Command Window or from a script.

<table>
<thead>
<tr>
<th>To...</th>
<th>Use This Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create a folder</td>
<td>mkdir</td>
</tr>
<tr>
<td>Create a text file, such as a MATLAB program file</td>
<td>edit</td>
</tr>
<tr>
<td>Create a MAT-file</td>
<td>save</td>
</tr>
<tr>
<td>Create archive of files</td>
<td>zip, gzip, tar</td>
</tr>
<tr>
<td>Extract files from archive</td>
<td>unzip, gunzip, untar</td>
</tr>
</tbody>
</table>

See also “Locations for Storing Your Files” on page 6-4.

Copying, Renaming, and Deleting Files and Folders

- “Renaming Files Using the Current Folder Browser” on page 6-36
- “Renaming Files and Folders Using Functions” on page 6-37
- “Deleting Files and Folders Using the Current Folder Browser” on page 6-37
- “Deleting Files and Folders Using Functions” on page 6-38
- “Copying and Moving Files and Folders” on page 6-39
- “Changing Properties of Files and Folders” on page 6-40

Renaming Files Using the Current Folder Browser

1 Select the item to rename.

2 Right-click and select Rename from the context menu.

3 Type over the existing name with the new name. Warnings appear when:
   - The new name is invalid. Change the name to make it valid. File names must start with a letter, and can contain letters, digits, or underscores.
The folder is on the search path. See “Handling Errors and Unexpected Behavior When Updating Folders” on page 6-82.

4 Press Enter.

**Renaming Files and Folders Using Functions**

Use the `movefile` function.

**Deleting Files and Folders Using the Current Folder Browser**

To remove items:

1 Select the item to delete. To select multiple items:
   - Click to the left of an icon and drag up or down to select contiguous items
   - **Shift**+click to select contiguous items
   - **Ctrl**+click to select non-contiguous items

2 Right-click and select **Delete** from the context menu.

---

**Note** You cannot delete a folder while it is on the search path. See “Handling Errors and Unexpected Behavior When Updating Folders” on page 6-82.

---

When you delete a file or folder using the Current Folder browser, MATLAB permanently removes it or moves it to another location, based on your platform.
Managing Files in MATLAB®

Platform Behavior Deleting Files and Folders Using the Current Folder Browser

<table>
<thead>
<tr>
<th>Platform</th>
<th>Behavior Deleting Files and Folders Using the Current Folder Browser</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft Windows platforms</td>
<td>Follows the Windows system preference for sending files to the Recycle Bin. Some systems only allow recycling of local files and not files accessed on a network. To delete a selection permanently when the system preference is set to recycle, press <strong>Shift+Delete</strong>.</td>
</tr>
<tr>
<td>Linux platforms</td>
<td>Specify the behavior:</td>
</tr>
<tr>
<td></td>
<td>1 On the <strong>Home</strong> tab, in the <strong>Environment</strong> section, click <strong>Preferences &gt; General</strong>.</td>
</tr>
<tr>
<td></td>
<td>2 Set the <strong>Deleting files</strong> option you want.</td>
</tr>
<tr>
<td></td>
<td>To move files to a temporary folder, determine the location by running <strong>tempdir</strong>.</td>
</tr>
<tr>
<td></td>
<td>To delete a selection permanently when the preference is set to recycle, press <strong>Shift+Delete</strong>.</td>
</tr>
<tr>
<td>Apple Macintosh platforms</td>
<td>Follows your Macintosh system preference for sending files to the Trash.</td>
</tr>
</tbody>
</table>

Deleting Files and Folders Using Functions

<table>
<thead>
<tr>
<th>To...</th>
<th>Use This Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delete a file</td>
<td>delete</td>
</tr>
<tr>
<td>Delete a folder</td>
<td>rmdir</td>
</tr>
</tbody>
</table>

You cannot recover folders deleted using **rmdir**.

By default, the **delete** function permanently deletes files. To move them to a different location instead, use the **Deleting files** preference:

1 On the **Home** tab, in the **Environment** section, click **Preferences > General**.
2 Set the **Deleting files** option you want.

Setting the preference to delete files permanently makes `delete` run faster.

To override the preference when using the `delete` function, use the `recycle` function.

The location for deleted files varies by platform, as the following table indicates.

<table>
<thead>
<tr>
<th>Platform</th>
<th>Location for Files Not Permanently Deleted Using the <code>delete</code> Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft Windows platforms</td>
<td>Recycle Bin. Some systems only allow recycling of local files and not files accessed on a network.</td>
</tr>
<tr>
<td>Linux platforms</td>
<td>MATLAB_Files_&lt;day&gt;-&lt;mo&gt;-&lt;yr&gt;<em>&lt;hr&gt;</em>&lt;min&gt;_&lt;sec&gt; folder in the location returned by the <code>tempdir</code> function. For example, when <code>tempdir</code> returns <code>/tmp</code>, files deleted at 2:09:28 in the afternoon of November 9, 2009 move to <code>/tmp/MATLAB_Files_09-Nov-2009_14_09_28</code>.</td>
</tr>
<tr>
<td>Apple Macintosh platforms</td>
<td>Trash</td>
</tr>
</tbody>
</table>

Deleted files remain in these locations until you remove them. To remove deleted files, use operating system features, such as **Empty Recycle Bin** on Windows platforms.

**Copying and Moving Files and Folders**

Copy and move files and folders using the Current Folder browser using standard GUI practices. For example, click and drag a file from one folder to another or to another application, such as Windows Explorer.
To copy and move files and folders using functions, use `copyfile` and `movefile`.

**Changing Properties of Files and Folders**

To change some properties of files and folders, such as read/write permissions, use the `fileattrib` function.

**Opening and Running Files**

- “Opening Files and Importing Data Using the Current Folder Browser” on page 6-40
- “Opening Files and Importing Data Using Functions” on page 6-41
- “Opening Files and Functions in the Command Window” on page 6-41
- “Running MATLAB Program Files from the Current Folder Browser” on page 6-42

**Opening Files and Importing Data Using the Current Folder Browser**

1. In the Current Folder browser, right-click the file you want to open or load.
2. From the context menu, select an option for opening or importing the file:
   - **Open** — Opens the file using the appropriate MATLAB tool for the file type. For example, this option loads a MAT-file into the Workspace browser.
   - **Open in GUIDE** — Opens a FIG-file in GUIDE instead of a figure window. For more information, see “Opening GUIDE”.
   - **Open as Text** — Opens the file in the Editor as a text file, even if the file type is associated with another application or tool.
This is useful, for example, if you have imported a tab-delimited data file (.dat) into the workspace and you find you want to add a data point. Open the file as text in the Editor, make your addition, and then save the file.

- **Open Outside MATLAB** — Opens the file using the application or tool that the operating system associates with the file type.

For example, .mat is the extension for MATLAB data files and Microsoft Access files. Whereas Open loads the file into the MATLAB workspace, **Open Outside MATLAB** opens the file into Microsoft Access. See “Managing File Associations for MATLAB on Windows Systems” on page 1-5.

For information on how to view information about a file without opening it, see “Viewing File Details Without Opening Files” on page 6-18.

### Opening Files and Importing Data Using Functions

<table>
<thead>
<tr>
<th>To...</th>
<th>Use This Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open a file or open a variable in the Variable Editor</td>
<td>open</td>
</tr>
<tr>
<td>Add variables from a MAT-file to the workspace</td>
<td>load</td>
</tr>
<tr>
<td>Import data files</td>
<td>importdata</td>
</tr>
<tr>
<td>Import data file using the Import Wizard</td>
<td>uimimport</td>
</tr>
<tr>
<td>Access the system clipboard</td>
<td>clipboard</td>
</tr>
</tbody>
</table>

**See Also.**

- “Supported File Formats”

### Opening Files and Functions in the Command Window

To open a function, file, variable, or Simulink model from the Command Window — select the name in the Command Window, right-click, and then select **Open Selection**.
MATLAB runs the `open` function on your behalf to open the selected item in the appropriate tool. Specifically:

- Text files open in the Editor.
- Figure files (.fig) open in a figure window.
- Variables open in the Variable Editor.
- Models open in Simulink software.

See the `open` reference page for details about what action occurs if there are name conflicts. If no action exists to work with the selected item, `Open selection` calls `edit`.

**Running MATLAB Program Files from the Current Folder Browser**

For convenience, you can run MATLAB scripts and functions from the Current Folder browser. Script files do not accept input arguments or return values and can be run directly. If the program is a function which requires input arguments or returns output arguments, you can define a `run configuration` for it that defines arguments. See “Using Run Configurations for Functions” on page 6-43. Run any program file in the following way:

1. In the Current Folder browser, change the current folder to the folder containing the file to run.

2. Right-click the file name to open the context menu.

3. (Optional) If you have defined a run configuration for the file you want to use, select it from the **Run Configurations** on the context menu. Select **Edit Configurations** to edit or create one.

4. From the context menu, select **Run**.

If you have customized the Current Folder Browser toolbar with a **Run** button, you can select the file and then click the **Run** button. That button has a dropdown list of function run configurations you have defined. For details, see “Access Frequently Used Features” on page 2-10.
Using Run Configurations for Functions. If you run a function that requires input arguments, executing it from the Run context menu or toolbar button item might not work properly. Specify default input arguments for functions that require them by defining run configurations for them. To create a run configuration:

1 Right-click the name of a function in the Current Folder browser and select Run Configurations > Edit Configurations.

2 Give your run configuration a name.

3 Type in the expressions for running the function in the MATLAB expression panel.

4 Click Run if you want to test the configuration.

5 Click Close to save the configuration and exit the dialog box.

Executing a function with a run configuration sets up function arguments as the configuration specifies. You can create multiple configurations for a function. Your configurations are saved with your preferences. To use a run configuration:

1 Right-click on the function name and select Run Configurations > configuration name.

2 The function executes according to the configuration you select and that configuration is the selected one the next time you use this context menu.

For more information about using run configurations, see “Run Functions in the Editor”.

Running External Commands, Scripts, and Programs

The exclamation point character (!) sometimes called bang, is a shell escape and indicates that the rest of the input line is a command to the operating system. Use it to invoke utilities or call other executable programs without quitting MATLAB. On UNIX platforms, for example, the following code invokes the vi editor for a file named yearlystats.m:

!vi yearlystats.m
After the external program completes or you quit the program, the operating system returns control to MATLAB. Add & to the end of the line, such as

!dir &

on Windows platforms to display the output in a separate window or to run the application in background mode. For example

!excel.exe &

opens Microsoft Excel software and returns control to the Command Window so you can continue running MATLAB language statements.

Restrictions maintained within the operating system determine the maximum length of the argument list you can provide as input to the bang (!) command. If you are running the Microsoft Windows XP operating system, for example, the length of the argument list input to the bang command cannot exceed 8189 characters.

See the reference pages for the unix, dos, and system functions for details about running external programs that return results and status.

**Note** To execute operating system commands with specific environment variables, include all commands to the operating system within the system call. Separate the commands using & (ampersand) for DOS, and ; (semicolon) for UNIX platforms. This applies to the MATLAB ! (bang), dos, unix, and system functions. Another approach is to set environment variables before starting MATLAB.

On Macintosh platforms, you cannot run AppleScript (from Apple) directly from MATLAB. However, you can run the Apple Mac OS X osascript function from the MATLAB unix or ! (bang) function to run AppleScript from MATLAB.
Running UNIX Programs That Are Off the System Path

You can run a UNIX program from MATLAB when the folder containing that file is not on the UNIX system path that is visible to MATLAB. To determine the system path that is visible to MATLAB, type the following in the Command Window:

`getenv('PATH')`

You can make modifications to the system path that persist for the current MATLAB session or across subsequent MATLAB sessions, as described in the sections that follow.

**Modify the System Path for the Current MATLAB Session.** Do one of the following:

- Change the current folder in MATLAB to the folder that contains the program you want to run.
- Issue these commands using the Command Window:

  ```
  path1 = getenv('PATH')
  path1 = [path1 ':/usr/local/bin']
  setenv('PATH', path1)
  !echo $PATH
  ```

If you restart MATLAB, the folder is no longer on the system path visible to MATLAB.

**Modify the System Path Across MATLAB Sessions Within the Current Shell Session.** To add a folder to the system path from the shell:

1. Stop MATLAB.
2. Depending on the shell you are using, type one of the following at the system command prompt, where `myfolder` is the folder that contains the program you want to run:

   - Type this if you are using `bash` or a related shell:
     ```
     export PATH="$PATH:$myfolder"
     ```
   - Type this if you are using `tcsh` or a related shell:
Managing Files in MATLAB®

3 Start MATLAB.

4 In the MATLAB Command Window, type:

```bash
!echo $PATH
```

If you restart MATLAB within the current shell session, the folder remains on the system path visible to MATLAB. However, if you restart the shell session, and then restart MATLAB, the folder is no longer on the system path visible to MATLAB.

**Modify the System Path Across All MATLAB Sessions.** To make adjustments that persist across shell and MATLAB sessions, add the following commands to the MATLAB startup file as described in “Specifying Startup Options in the MATLAB Startup File” on page 1-19:

```matlab
path1 = getenv('PATH')
path1 = [path1 ':/usr/local/bin']
setenv('PATH', path1)
!echo $PATH
```

Comparing Files and Folders

You can use the Comparison Tool to determine and display the differences between selected pairs of files or folders. The comparison process involves three steps:

1  “Select the Files or Folders to Compare” on page 6-47

2  “Choose a Comparison Type” on page 6-48

3  “Explore the Comparison Tool Report” on page 6-49

**Select the Files or Folders to Compare**
You can compare files and folders using any of these methods:

- From the Current Folder browser:
  - Select a file or folder, right-click and select **Compare Against**, and browse to select a second item to compare.
  - For two files or subfolders in the same folder, select the files or folders, right-click and select **Compare Selected Files/Folders**.

- If you have a file open in the Editor, on the **Editor** tab, in the **File** section,
- Click **Compare** to browse to a second file for comparison.
- Alternatively, under **Compare**, select **Compare with Version on Disk** or **Compare with Autosave**. See “Comparing Files with Autosave Version or Version on Disk” on page 6-58.

- From the MATLAB desktop, on the **Home** tab, in the **File** section, click **Compare**. Select the files or folders to compare.
- From the command line, use the `visdiff` function.

**Choose a Comparison Type**

If you specify two files or folders to compare using either the Current Folder browser or the `visdiff` function, then the Comparison Tool automatically performs the default comparison type.

If there are multiple comparison types available for your selections, you can change what type of comparison to run. For example, text, binary, file list, or XML comparison. To change the comparison type, create a new comparison using the Comparison Tool. You can change comparison type in the Select Files or Folders for Comparison dialog box.

For example, from the Current Folder browser, if you select two MAT-files to compare, you get the default comparison type showing information about the variables. To change the comparison type to binary, create a new comparison...
using the Comparison Tool. See “Selecting Files or Folders to Compare from the Comparison Tool” on page 6-62.

**Explore the Comparison Tool Report**

Comparison Tool report features depend on your comparison type. You can use the tool to:

- Compare lines in two text files (some other applications refer to this as a *file diff* operation). See “Comparing Text Files” on page 6-53.
- Compare and merge variables in two MAT-files. See “Comparing MAT-Files” on page 6-59.
- Determine whether the contents of two binary files match. See “Comparing Binary Files” on page 6-61.
- Compare any combination of folders, zip files, or Simulink manifests to determine:
  - Which file and folder names are unique to each list
  - If files and folders with the same name in each list have the same content
  See “Comparing Folders and Zip Files” on page 6-49.
- Compare XML files:
  - If you select XML files to compare and you have MATLAB Report Generator™ software, the Comparison Tool runs a hierarchical matching algorithm. You then see a report showing a hierarchical view of the portions of the two XML files that differ.
  - If you have Simulink Report Generator software, you can select a pair of Simulink models (.mdl files) to compare XML files generated from them.

**Comparing Folders and Zip Files**

- “Folder Comparison Report” on page 6-50
- “Highlighting of Differences” on page 6-51
- “Next Steps Using the Report” on page 6-52
Folder Comparison Report

To select items to compare, see “Select the Files or Folders to Compare” on page 6-47. You can perform file list comparisons for any combinations of folders, zip files, and Simulink Manifests.

When you use the Comparison Tool to compare two folders (sometimes referred to as directories) or any file list comparison (for example, folder versus zip file), a window opens and presents the contents side by side. The tool enables you to:

- Determine the files that the comparison lists have in common.
- Determine if files with identical names that are common to both comparison lists also have identical content.
- Open a new comparison of two files or folders that are common to both comparison lists, but have different content.
- Open a file for viewing in the Editor.
- Specify filters to ignore certain files or folders

For list comparisons, if you want to expand the list to see all files in subfolders in one report, select the Include subfolders check box when selecting items to compare. If you do not include subfolders, you can click compare links in the report to open a new comparison of two folders with changed content.

To define filters to exclude unimportant differences, click the Filters toolbar button, or select Comparison > Edit File and Folder Filters.

The File and Folder Filters dialog box opens. Specify filters to ignore certain files and folders, such as backup files or files created by a revision control system. Filters can save time when reviewing differences, especially when comparing many subfolders. Double-click to edit existing filters.

For example, to ignore all files and folders in a folder named CVS, open the File and Folder Filter dialog box and enter:

CVS/

To ignore all files in a folder named CVS, but not ignore subfolders, enter:
Comparing Files and Folders

CVS/*

**Highlighting of Differences**
The Comparison Tool displays the contents of the lists side by side and highlights files and subfolders that do not match. The following table describes how the tool highlights each type of change. The status message (such as *identical* or *contents changed*) appears in the *Change Summary* column.

<table>
<thead>
<tr>
<th>Change Summary</th>
<th>Highlighting for Folders</th>
<th>Highlighting for Files</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contents changed</td>
<td>Dark pink</td>
<td>Pink</td>
<td>The contents of the files or folders differ. Click the <em>compare</em> link to investigate.</td>
</tr>
<tr>
<td>Added</td>
<td>Dark green</td>
<td>Green</td>
<td>File or folder only exists in the right list.</td>
</tr>
<tr>
<td>Removed</td>
<td>Dark purple</td>
<td>Purple</td>
<td>File or folder only exists in the left list.</td>
</tr>
<tr>
<td>Identical</td>
<td>None</td>
<td>None</td>
<td></td>
</tr>
</tbody>
</table>

The following image shows an example of the Comparison Tool when two folders are compared. The results are sorted by *Type*. 
Next Steps Using the Report

To explore the report you can use the following tools:

- You can sort the results by name, type, size, or last modified timestamp by clicking the column headers. For example, click the *Type* column header to sort by folder and file type, as shown in the preceding figure.

- To open a new comparison of two files or folders with changed contents, click the **compare** link next to file or folder names highlighted in pink.

- To open a file in the Editor, click the **open** link next to a file name.

If the file is present in both folders, you can click links to open the **left** or **right** file.
Comparing Files and Folders

If subfolders are very large and contain many files, analysis continues in the background. The tool displays the number of items still to be compared at the top of the report, as shown in the next figure. You can click the links to **Skip Current** item or **Cancel All** to stop further analysis.

For details on other comparison tool features, see “Using Comparison Tool Features” on page 6-62.

**Comparing Text Files**

- “Selecting Text Files to Compare” on page 6-53
- “Highlighting of Differences” on page 6-53
- “Stepping Through Differences” on page 6-56
- “Viewing a Summary of Differences” on page 6-56
- “Ignoring Whitespace Differences in Text Comparisons” on page 6-57
- “Showing Differences Only” on page 6-57
- “Changing the Display Width of a Text Comparison” on page 6-57
- “Saving HTML Report” on page 6-57

**Selecting Text Files to Compare**
To select files to compare, see “Select the Files or Folders to Compare” on page 6-47.

**Highlighting of Differences**
When you use the Comparison Tool to compare two text files, a window opens and presents the two files side by side. Symbols indicate how you can adjust the files to make them match. This feature can be useful when you want to compare the latest version of a text file to an autosave version.

The Comparison Tool report displays the files side by side and highlights lines that do not match, as follows:
• Dark pink highlighting indicates changed characters within lines.
• Pink highlighting and an x between the two files indicate that the content of the lines differs between the two files.
• Green highlighting and a right (>) or left (<) angle bracket between the two files indicate a line that exists on one side only.
Comparing Files and Folders

The Comparison Tool attempts to match lines and detects local text that is added, deleted, or changed. It does not do a simple line-by-line comparison. In the previous image, for example, the tool determines that `lengthofline.m` has a line of code that does not exist in `lengthofline2.m` and highlights it (line 23) in green. Also, notice that the tool takes the additional line into
account and determines that the line containing the end statement in each file matches, even though the end statement does not occur on the same line number.

If the files you are comparing are extremely long, the tool could run out of memory while attempting to perform the file comparison. In which case, the message,

**Maximum file length exceeded.**
**Defaulting to line-by-line comparison.**

appears. In a line-by-line comparison, the tool highlights the lines containing the end statement because in performing this operation, it finds that the last line in one file does not match the last line in the other file.

**Stepping Through Differences**

Because text files can be lengthy, the Comparison Tool provides toolbar buttons to help you step through the results from one difference to the next.

To navigate through comparison results:

- Click the down arrow toolbar button to go to the next set of lines that differ. If no additional sets of lines differ, the down arrow takes you to the end of the file.
- Click the up arrow toolbar button to go to a previous set of lines that differ. If no previous set of lines differ, the up arrow takes you to the beginning of the file.

Alternatively, use the menu items **Comparison > Next** or **Previous**.

**Viewing a Summary of Differences**

To see a summary of differences between two text files, scroll to the bottom of the Comparison Tool and view the list, which contains information such as:
Comparing Files and Folders

- Number of matching lines: 52
- Number of unmatched lines in left-hand file: 12
- Number of unmatched lines in right-hand file: 15

**Ignoring Whitespace Differences in Text Comparisons**
You may want to hide whitespace differences to help you distinguish between functional changes and changes to indentation.

Use the toolbar button, $\text{\$}$, to toggle the display of differences only involving whitespace characters, or select **Comparison > Ignore Whitespace**.

**Showing Differences Only**
You can specify whether to show only differences or entire files. It can be useful to hide unmodified lines in large text comparison reports. When you are showing differences only and sections are hidden, the report displays messages like the following: 10 unmodified lines hidden.

Use the toolbar button, $\text{\$}$, to toggle the display of sections of the report that do not contain any differences, or select **Comparison > Show Differences Only**.

**Changing the Display Width of a Text Comparison**
To increase or decrease the line lengths of the text files in the comparison display, edit the number in the **Columns visible** field. Resize the window, if necessary.

For details on other comparison tool features, see “Using Comparison Tool Features” on page 6-62.

**Saving HTML Report**
Click Save As on the toolbar to save a copy of the comparison report as an HTML file. The tool creates a corresponding folder containing the style sheet and JavaScript files that control the report highlighting. To preserve the highlighting, keep the folder with the HTML file.
Comparing Files with Autosave Version or Version on Disk

From the Editor you can compare one open text file with another, or you can choose to compare the latest version of a file in the Editor to an autosave version or a saved version. For an example, follow these steps:

1. Open one of the text files you want to compare in the Editor.

   To open the example file provided, `lengthofline.m`, run the following command in the Command Window:

   ```matlab
   open(fullfile(matlabroot,'help','techdoc','matlab_env','examples','lengthofline.m'))
   ```

2. On the Editor tab, in the File section, click Compare. If your file is modified, the Editor saves the file before comparing. Alternatively, under Compare, select Save and Compare with.

   Navigate to the file you want to compare against, select the file, and click Open. To open the example file provided, select `lengthofline2.m` from the folder where you found `lengthofline.m`.

   Other options available are:

   - To compare the open file to the Editor’s automatic copy (`filename.asv`), under Compare, select Save and Compare with Autosave. If your file is modified, the Editor saves the file before comparing. For more information, see “Autosaving Files”.

   - To compare an open file that has been changed, but not saved, to the saved version, under Compare, select Compare with Version on Disk.
Comparing MAT-Files

**Note**  To select files to compare, see “Select the Files or Folders to Compare” on page 6-47.

You can use the Comparison Tool to compare two MAT-files. The tool presents the variables in the two files side by side, which enables you to:

- View and sort by the name, size, class, and change summary of all variables.
- View details of differences between variables, to see which fields of a structure are different, and view differences in individual elements of an array.
- Merge changes between files by copying modified variables from one file to the other (*Caution: No undo*).
- See which variables are common to each file and which are unique.
- Load the contents of the variables into the Variable Editor by clicking the name of that variable.
- Load the MAT-files into the workspace by clicking a **Load** link.
- Save a copy of the report as an HTML file. Click **Save As** on the toolbar.

The Comparison Tool report highlights changes in variables as follows.

<table>
<thead>
<tr>
<th>Change Summary</th>
<th>Highlighting</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Modified</strong></td>
<td>Pink</td>
<td>Values of the variable differ between the two files. Click the View differences button to investigate. A new variable comparison report opens to display differences in individual array elements or differing fields of a structure. Double-click pink rows or cells to investigate further layers of differences.</td>
</tr>
<tr>
<td><strong>Added</strong></td>
<td>Green</td>
<td>Variable only exists in right file.</td>
</tr>
</tbody>
</table>
### Change Summary

<table>
<thead>
<tr>
<th>Change Summary</th>
<th>Highlighting</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Removed</td>
<td>Purple</td>
<td>Variable only exists in right file.</td>
</tr>
<tr>
<td>Identical</td>
<td>None</td>
<td>Variable identical in both files.</td>
</tr>
<tr>
<td>Class changed</td>
<td>Pink (only in Class columns)</td>
<td>Variable data class changed. Click the View differences button to investigate.</td>
</tr>
</tbody>
</table>

Click the Merge button in the Merge column to copy modified variables from one file to the other.

The following image shows the results when you compare two files, `data1.mat` and `data2.mat`. 
Comparing Binary Files

**Note** To select files to compare, see “Select the Files or Folders to Compare” on page 6-47.
You can use the Comparison Tool to compare two binary files such as DLL files or MEX-files. Also, you can select the Binary comparison type for any pair of files with a choice of comparison types.

- If the files are the same, the tool displays the message: The files are identical.
- If the files differ, the tool displays the message: The files are different.

If the files differ, you can click the Show Details link to view the binary files and the byte offset of the first difference.

**Using Comparison Tool Features**

You can use the Comparison Tool for the following tasks:

- “Selecting Files or Folders to Compare from the Comparison Tool” on page 6-62
- “Exchanging the Left and Right Sides of the Report” on page 6-63
- “Refreshing the Report to Show Updated Files” on page 6-63
- “Finding Text” on page 6-63
- “Viewing New Comparisons” on page 6-63
- “Viewing Previous Comparisons” on page 6-64

**Selecting Files or Folders to Compare from the Comparison Tool**

To compare two files or folders from the Comparison Tool, follow these steps:

1. From the MATLAB desktop, on the Home tab, in the File section, click Compare. Select the files or folders to compare.

If the Comparison Tool is already open, compare files or folders by clicking the New comparison button or by selecting File > New Comparison.

The dialog box Select Files or Folders for Comparison appears.
Comparing Files and Folders

2 In the dialog box, select two files or folders to compare. Use the drop-down lists to select recent comparison items, or the **Browse** buttons to locate and select the items that you want to compare.

You also can drag and drop a file or folder from Windows Explorer to the left and right file and folder fields.

3 Optionally, choose the comparison type you want to use. Either use the default **Comparison type** value, or if multiple comparison types are available, select a different one from the list. For example, for text files you could select text or binary comparison types.

4 Click **Compare**.

**Exchanging the Left and Right Sides of the Report**
To move the file or folder on the left side to the right side and vice versa, select **File > Swap Sides**, or click the Swap sides button.

**Refreshing the Report to Show Updated Files**
After making changes to and saving the files in the Editor, update the results in the Comparison Tool by selecting **File > Refresh** or clicking the Refresh button.

**Finding Text**
To find a phrase in the current display, select **Edit > Find**, or click the Find text button. The resulting Find dialog box is the same as the one you use in the Command Window. For more information, see “Find Text in Command Window or History” on page 3-13.

**Viewing New Comparisons**
You can perform another file comparison by selecting **File > New Comparison**, or by clicking the toolbar button.

New comparisons open additional tabs in the Comparison Tool.
**Viewing Previous Comparisons**
You can see the results of previous comparisons in the current session by selecting that comparison's tab entry on the document bar at the bottom of the window. If you close the Comparison Tool, the current and previous comparisons are lost.

**Function Alternative for Comparing Files and Folders**
Use the `visdiff` function to open the Comparison Tool from the Command Window.

```matlab
visdiff(fileorfoldername1, fileorfoldername2)
```

For example, type:

```matlab
visdiff('lengthofline.m', 'lengthofline2.m')
```
Making Files and Folders Accessible to MATLAB

In this section...  
“Files and Folders That MATLAB Can Access” on page 6-65
“How to Make Files Accessible” on page 6-65
“Determining if MATLAB Can Access a File” on page 6-67
“Ensuring MATLAB Uses the File You Want” on page 6-68

Files and Folders That MATLAB Can Access
For performance reasons, MATLAB limits where it looks for files. To run or get help for a MATLAB program, or to load a MAT-file, the file must be in one of these locations:

- The current folder in MATLAB
- A folder that is on the search path. See “What Is on the Search Path?” on page 6-71

Make the following accessible to MATLAB:

- Folders containing files that you and others create.
- Folders containing files that are called by files you run.
- Subfolders containing files that you run. Making a folder accessible does not make its subfolders accessible.

For files in @ (class) and + (package) folders, make the parent folder accessible. For details, see “Organizing Classes in Folders”.

How to Make Files Accessible
For files that you and others create, see “Basic Options for Making Files Accessible” on page 6-66.

To understand the differences in the basic options, and for other approaches, see “All Options for Making Files Accessible” on page 6-66.
Basic Options for Making Files Accessible

- Store the files that you and others create in the MATLAB folder, which is on the search path. See “Locations for Storing Your Files” on page 6-4.
- Change the current folder to the folder that contains the files.
- Add the folders that contain the files to the search path.

All Options for Making Files Accessible

<table>
<thead>
<tr>
<th>Usage</th>
<th>Recommendation</th>
</tr>
</thead>
<tbody>
<tr>
<td>You <em>seldom</em> run the file.</td>
<td>Change the current folder to the folder that contains the file. See “Viewing and Changing the Current Folder” on page 6-2.</td>
</tr>
<tr>
<td>The file is a <em>script</em> (takes no input or output arguments).</td>
<td>Use the <code>run</code> function.</td>
</tr>
<tr>
<td>The files are in <em>one</em> folder.</td>
<td>Put the files in the <code>userpath</code> folder. See “Locations for Storing Your Files” on page 6-4.</td>
</tr>
<tr>
<td>The files are in <em>multiple</em> folders.</td>
<td>Add the folders to the search path. See “Adding Folders to the Search Path” on page 6-75.</td>
</tr>
<tr>
<td></td>
<td>If you regularly use the files, save the changes. See “Saving Changes to the Search Path” on page 6-78.</td>
</tr>
<tr>
<td>The files call other files that are in <em>multiple</em> folders.</td>
<td>1 Determine the location of all the called files. See “Dependencies Within a Folder”.</td>
</tr>
<tr>
<td></td>
<td>2 Add the folders to the search path. See “Adding Folders to the Search Path” on page 6-75.</td>
</tr>
</tbody>
</table>
### Usage Recommendation

<table>
<thead>
<tr>
<th>Usage</th>
<th>Recommendation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Some files in multiple folders have the same name.</td>
<td>See “Detecting and Addressing Name Conflicts” on page 6-69.</td>
</tr>
<tr>
<td>You use files in different versions of MATLAB or on different platforms.</td>
<td>Modify the search path in a startup.m file. See “Using the Search Path with Different MATLAB Installations” on page 6-79.</td>
</tr>
</tbody>
</table>

---

### Determining if MATLAB Can Access a File

The following table lists ways to determine if MATLAB has access to a file.

<table>
<thead>
<tr>
<th>Option</th>
<th>When MATLAB Can Access the File</th>
<th>When MATLAB Cannot Access the File</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use the file.</td>
<td>Works successfully.</td>
<td>Produces an error. Typical error notifications include:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Dialog box</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Message: Undefined function or method 'fileName'</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Message: Cannot find function 'fileName'</td>
</tr>
<tr>
<td>View the file in the Current Folder browser.</td>
<td>Either or both of the following are true:</td>
<td>Either or both of the following are true:</td>
</tr>
<tr>
<td></td>
<td>• File is in the current folder.</td>
<td>• File is in a subfolder of the current folder, and the subfolder is not on the search path.</td>
</tr>
<tr>
<td></td>
<td>• File does not appear dimmed in Current Folder browser, assuming the Indicate inaccessible files option is selected in “Preferences for the Current Folder Browser” on page 6-13.</td>
<td>• File appears dimmed in the Current Folder browser, assuming the Indicate inaccessible files option is selected in “Preferences for the Current Folder Browser” on page 6-13.</td>
</tr>
</tbody>
</table>
### Ensuring MATLAB Uses the File You Want

#### About Name Conflicts and Shadowed Files
When MATLAB has access to multiple files with the same name, these precedence rules determine the file MATLAB uses:

- MATLAB uses the file in the current folder instead of a file in a folder on the search path.
- MATLAB uses the file whose folder is closest to the top of the search path instead of a file further down.

The file that MATLAB does not use is called a `shadowed` file. In some cases, MATLAB warns you that a shadowed file exists.

Other name conflicts include the following:
• A file has the same name as a variable in the base workspace.
• A file has the same name as a built-in function for a MathWorks product.

When there are name conflicts, MATLAB follows these precedence rules:

• “Function Precedence Order”
• “Class Precedence and MATLAB Path”

**Detecting and Addressing Name Conflicts**
MATLAB might not be accessing the file that you want it to when:

• You use a file and get a warning about a potential name conflict.
• You get unexpected results.

To identify a name conflict, try using the `which` function.

To address a name conflict, try one of the following:

• Change the current folder.
• Move or remove folders on the search path.
• Rename or move files.
• Specify the full path or partial path to the file that you want.
• Maintain a single version of a file instead of multiple versions.

Name conflicts can arise from using files that you create. Conflicts also can arise from using:

• Files that others create, such as from File Exchange
• A different system that has additional MathWorks products installed
• A different version of MATLAB, which could include new functions that have the same names as your existing files
See Also

- rehash
- “Toolbox Path Caching in the MATLAB Program” on page 1-23
# Using the MATLAB Search Path

## In this section...

- “What Is the Search Path?” on page 6-71
- “Viewing Files and Folders on the Search Path” on page 6-73
- “Changing the Search Path” on page 6-74
- “Using the Search Path with Different MATLAB Installations” on page 6-79
- “Recovering from Problems with the Search Path” on page 6-80
- “Handling Errors and Unexpected Behavior When Updating Folders” on page 6-82

## What Is the Search Path?

The search path, or *path* is a subset of all the folders in the file system. MATLAB software uses the search path to locate files used with MathWorks products efficiently. MATLAB can access all files in the folders on the search path.

## What Is on the Search Path?

- By default, folders provided with MATLAB and other MathWorks products. These folders are under `matlabroot/toolbox`, where `matlabroot` is the folder displayed when you type `matlabroot` in the Command Window.
- By default, the MATLAB `userpath`. See “Locations for Storing Your Files” on page 6-4.
- Folders you explicitly add to the search path for the files you and others create.

Adding folders to the search path is like performing an include or import operation in other applications.

Class, package, and private folders are *not* on the search path. See “Files and Folders That MATLAB Can Access” on page 6-65.
Order of Folders on the Search Path

The order of folders on the search path is important when two files with the same name are in folders on the search path. MATLAB uses the file nearest to the top of the search path. To customize the order of files on the search path, see “Ensuring MATLAB Uses the File You Want” on page 6-68.

Relationship Between the Search Path and the System Path

The search path is not the same as the system path. Furthermore, there is no explicit relationship between the MATLAB search path and the system path. However, both paths help in locating files, as follows:

- MATLAB uses the search path to locate MATLAB files efficiently.
- The operating system uses a system path to locate operating system files efficiently.

Therefore, you can issue MATLAB commands that result in the use of both the MATLAB search path and the system path. For example, if you type `dos('tasklist &')` in the MATLAB Command Window, then:

1. MATLAB uses the search path to locate and run `dos.m`.
2. The `dos` function passes `tasklist &` to the Microsoft Windows operating system.
3. Microsoft Windows uses the system path to locate and run `tasklist.exe`.

Similar behavior results when you use the MATLAB `unix` and `system` functions or the shell escape (!). For details on using the shell escape with MATLAB, see “Running External Commands, Scripts, and Programs” on page 6-43.

How MATLAB Stores the Search Path

MATLAB saves the search path information in the `pathdef.m` file. The `pathdef.m` file is a series of full path names, one for each folder on the search path, separated by a semicolon (;).
By default, pathdef.m is in `matlabroot/toolbox/local`.

When you change the search path, MATLAB uses it in the current session. To use it in future sessions, save the changes as described in “Saving Changes to the Search Path” on page 6-78.

**Viewing Files and Folders on the Search Path**

MATLAB provides various ways for you to view the search path, as described in these topics:

- “Using the Current Folder Browser” on page 6-73
- “Using the Set Path Dialog Box” on page 6-73
- Using MATLAB `path` and `pathtool` functions

**Using the Current Folder Browser**

To determine if files or folders in the Current Folder browser are on the search path:

1. In the Current Folder browser, right-click any file or folder, and ensure there is a check mark next to **Indicate Files Not on Path**.

   If there is no check mark, select **Indicate Files Not on Path**. A check mark appears.

2. Hover the pointer over any dimmed file or folder in the Current Folder browser to find out why it is dimmed.

   A tooltip opens with an explanation. Frequently, the tooltip indicates that the file or folder is not on the MATLAB path.

**Using the Set Path Dialog Box**

You can view the entire MATLAB search path. On the **Home** tab, in the **Environment** section, click **Set Path**.

The Set Path dialog box opens, listing all folders on the search path.
Changing the Search Path
MATLAB provides various ways for you to change the search path, as described in the following sections:

- “Adding Folders to the Search Path” on page 6-75
- “Removing Folders from the Search Path” on page 6-76
- “Changing the Order of Folders on the Search Path” on page 6-77
- “Saving Changes to the Search Path” on page 6-78
- “Specifying Startup Options in the MATLAB Startup File” on page 1-19
- Using MATLAB path and pathtool functions
Adding Folders to the Search Path

You can add folders to the search path for just the current session, or for both the current and future sessions.

Current Session Only. To add folders to the top of the search path for the duration of the current session use the following method:

1. From the Current Folder browser, select, and then right-click the folder or folders to add.
2. From the context menu, select Add to Path, and then select an option:
   - Selected Folders
   - Selected Folders and Subfolders

To add the folder that contains an Editor document to the top of the search path:

1. In the Editor, right-click the document tab.
2. Select Add folder-name to Search Path.

To change the ordering in the search path, follow the instructions in “Changing the Order of Folders on the Search Path” on page 6-77.

Current and Future Sessions. To add folders to the search path for the current session and future sessions, use the addpath function, or follow these steps:

   The Set Path dialog box appear.
2. Click one of these buttons:
   - Add Folder
   - Add with Subfolders
3. In the Add Folder to Path dialog box, select the folder to add to the search path, and then click Select Folder.
MATLAB adds the specified folder to the top of the search path.

If you do not want the folder at the top of the search path, see “Changing the Order of Folders on the Search Path” on page 6-77.

4 Keep or cancel the search path changes:

- To use the newly modified search path only in the current session, click Close.
- To reuse the newly modified search path in the current session and future sessions, click Save, and then click Close.
  For details on where to save the file, see “Saving Changes to the Search Path” on page 6-78.
- To undo your changes, click Revert, and then click Close.
- To restore the default search path, click Default, and then click Close. See “Restoring the Default Search Path” on page 6-79.

Removing Folders from the Search Path
You can remove folders from the search path for just the current session, or for both the current and future sessions.

For the Current Session Only. To remove one or more folders from the search path:

1 Select and right-click the folder or folders to remove.

2 From the context menu, select Remove from Path, and then select an option:
  • Selected Folders
  • Selected Folders and Subfolders

To remove the folder that contains an Editor document from the search path:

1 In the Editor, right-click the document tab.

Document tabs appear in the Editor only when multiple documents are open and docked in the Editor.
2 Select **Remove folder-name from Search Path**.

**For Current and Future Sessions.** To remove folders from the search path for the current and future sessions, use the `rmpath` function, or follow these steps:

1 Open the Set Path dialog box. On the **Home** tab, in the **Environment** section, click **Set Path**.

2 Select the folders to remove from the search path.

3 Click **Remove**.

4 Apply the changes:

   - To use the newly modified search path only in the current session, click **Close**.
   - To reuse the newly modified search path in the current session and future sessions, click **Save**.

     For details on where to save the file, see “Saving Changes to the Search Path” on page 6-78.

   - To restore the default search path, click **Default**. See “Restoring the Default Search Path” on page 6-79.

5 Click **Close**.

**Changing the Order of Folders on the Search Path**
Change the order of folders in the search path when files with the same name appear in multiple folders on the search path. When you specify such a file, MATLAB uses the one found in the folder nearest to the top of the search path.

**Moving Folders to Various Positions on the Search Path.** To change the order of folders on the search path:

1 Open the Set Path dialog box. On the **Home** tab, in the **Environment** section, click **Set Path**.

2 Select the folders to move on the search path.
3 Click one of the Move buttons, such as Move to Top. The order of the folders changes.

4 To use the modified search path in future sessions, click Save.

If you do not save the changes, the newly modified search path remains in effect until you end the current MATLAB session.

5 Click Close.

---

**Note** The MATLAB (userpath) folder automatically moves to the top of the search path the next time you start MATLAB. See “Locations for Storing Your Files” on page 6-4.

---

**Moving a Folder to the Top or Bottom of the Search Path.** To move a folder to the top or bottom of the search path, use the path function.

---

**Saving Changes to the Search Path**

Changes you make to the search path always remain in effect during the current MATLAB session. For MATLAB to use the changed search path in future sessions, save the search path, which updates the pathdef.m file.

---

**Note** The MATLAB (userpath) folder automatically moves to the top of the search path the next time you start MATLAB. See “Locations for Storing Your Files” on page 6-4.

---

**Ways to Save Changes.** To save changes to the search path, do one of the following:

- Click Save in the Set Path dialog box. See “Using the Set Path Dialog Box” on page 6-73.

- Use the savepath function.

**Where to Save the Search Path File.** Save the search path to the default location, matlabroot/toolbox/local, so MATLAB can locate it.
If you do not have write access to the default location, MATLAB prompts you for a different location. Choose the MATLAB startup folder.

**Restoring the Default Search Path.** The default search path contains only folders provided by MathWorks.

To restore the default search path, do one of the following:

- Click **Default** in the Set Path dialog box. See “Using the Set Path Dialog Box” on page 6-73. This method also adds the *userpath* folder. See “Locations for Storing Your Files” on page 6-4.
- Use the `restoredefaultpath` function.

See also “Recovering from Problems with the Search Path” on page 6-80.

**Using the Search Path with Different MATLAB Installations**

**Using the Search Path with Different Versions**
The default search path changes for each MATLAB version because the default folders that come with the products change. Different MATLAB versions cannot use the same `pathdef.m` file.

To use your files with a new MATLAB version or with multiple versions, do one of the following:

- For each version, add the folders containing your files to the search path. Save the search path where that version of MATLAB can access it.
- Instead of changing the `pathdef.m` file, include `addpath` statements in the `startup.m` file. Use the same `startup.m` file with the multiple versions of MATLAB.

**Using the Search Path with Different Platforms**
To use your files with MATLAB on different platforms, include `addpath` statements in the `startup.m` file. For more information, see “Specifying Startup Options in the MATLAB Startup File” on page 1-19.
Recovering from Problems with the Search Path

When there is a problem with the search path, you cannot use MATLAB successfully.

You could experience search path problems when:

- You save the search path on a Windows platform, and then try to use the same `pathdef.m` file on a Linux platform.
- The `pathdef.m` file becomes corrupt, invalid, renamed, or deleted.
- MATLAB cannot locate the `pathdef.m` file.

For example, when you start MATLAB, if a message like the following appears, it indicates a problem with the search path:

**Warning:** MATLAB did not appear to successfully set the search path...

To recover from problems with the search path, try the following steps. Proceed from one step to the next only as necessary.

1. Ensure MATLAB is using the `pathdef.m` file you expect:
   a. Run
      ```matlab
      which pathdef
      ```
   b. If you want MATLAB to use the `pathdef.m` file at another location, make corrections. For example, delete the incorrect `pathdef.m` file and ensure the correct `pathdef.m` file is in a location that MATLAB can access. See “Where to Save the Search Path File” on page 6-78.

2. Look for and correct problems with the `pathdef.m` and `startup.m` files:
   a. Open `pathdef.m` and `startup.m` in a text editor. Depending on the problem, you might not be able to open the `pathdef.m` file.
   b. Look for obvious problems, such as invalid characters or path names.
   c. Make corrections and save the files.
   d. Restart MATLAB to ensure that the problem does not recur.
3 Try to correct the problem using the Set Path dialog box:
   a Restore the default search path and save it. See “Using the Set Path Dialog Box” on page 6-73. Depending on the problem, you might not be able to open the dialog box.
   b Restart MATLAB to ensure that the problem does not recur.

4 Restore the default search path using functions:
   a Run restoredefaultpath, which sets the search path to the default and stores it in matlabroot/toolbox/local.
   b If restoredefaultpath seems to correct the problem, run savepath.
   c Restart MATLAB to ensure that the problem does not recur.

Depending on the problem, a message such as the following could appear:

The path may be bad. Please save your work (if desired), and quit.

5 Correct the search path problems encountered during startup:
   a Run

       restoredefaultpath; matlabrc

       Wait a few minutes until it completes.
   b If there is a pathdef.m file in the startup folder, it caused the problem. Either remove the bad pathdef.m file or replace it with a good pathdef.m file. For example, run:

       savepath('path_to_your_startup_folder/pathdef.m')

       See “Startup Folder for the MATLAB Program” on page 1-12.
   c Restart MATLAB to ensure that the problem does not recur.

After correcting problems with the search path, make any changes to run your files. For example, add the userpath folder or other folders to the search path.
Handling Errors and Unexpected Behavior When Updating Folders

You can encounter errors or unexpected behavior when you try to delete, rename, or move folders that:

- Are on the search path
- Contain subfolders that are on the search path

The behavior varies by platform because it depends on the behavior of similar features in the operating system.

If your task fails and the error message indicates it is because the folder is on the search path, then:

1. Remove the folder from the search path.
2. Delete, rename, or move the folder.
3. Add the folder to the search path.
Editor Preferences

- “Editor/Debugger Preferences” on page 7-2
- “Code Analyzer Preferences” on page 7-11
Editor/Debugger Preferences

<table>
<thead>
<tr>
<th>In this section...</th>
</tr>
</thead>
<tbody>
<tr>
<td>“General Preferences for the Editor/Debugger” on page 7-2</td>
</tr>
<tr>
<td>“Editor/Debugger Display Preferences” on page 7-3</td>
</tr>
<tr>
<td>“Editor/Debugger Tab Preferences” on page 7-4</td>
</tr>
<tr>
<td>“Editor/Debugger Language Preferences” on page 7-5</td>
</tr>
<tr>
<td>“Editor/Debugger Code Folding Preferences” on page 7-8</td>
</tr>
<tr>
<td>“Editor/Debugger Autosave Preferences” on page 7-9</td>
</tr>
</tbody>
</table>

General Preferences for the Editor/Debugger

You can specify which editor MATLAB uses, as well as how the MATLAB Editor behaves under various circumstances.

On the Home tab, in the Environment section, click Preferences. Select Editor/Debugger, and then adjust preference options as described in the table below.

<table>
<thead>
<tr>
<th>Preference</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Editor</td>
<td>Select which editor you want the MATLAB desktop to use when you edit a file:</td>
</tr>
<tr>
<td></td>
<td>• MATLAB Editor</td>
</tr>
<tr>
<td></td>
<td>• Text editor</td>
</tr>
<tr>
<td></td>
<td>If you select Text editor, specify the full path for the editor application you want to use, such as Emacs or vi. For example, c:/Applications/Emacs.exe.</td>
</tr>
<tr>
<td>Most recently used file list</td>
<td>In the Number of entries field, type the number of files that you want to appear in the list of recently used files at the bottom of the File menu.</td>
</tr>
</tbody>
</table>
## Editor/Debugger Preferences

**Preference Usage**

**Opening files in editor**
- Select **On restart open files from previous MATLAB sessions** if you want the Editor and the files it contained during your last MATLAB session to reopen when you restart MATLAB.

**Automatic file changes**
- Select **Reload unedited files that have been externally modified** if you want the Editor to automatically reload the version of a file that you opened and edited outside of MATLAB when the file currently open in the Editor has no unsaved changes.

- Select **Add line termination at end of file** to have MATLAB add a new empty line (sometimes referred to as a `<CR>`) to the end of a file automatically if the last line in the file is not empty.

## Editor/Debugger Display Preferences

You can change the appearance of the Editor.

On the **Home** tab, in the **Environment** section, click **Preferences**. Select **Editor/Debugger > Display**, and then adjust preference options as described in the table below.

<table>
<thead>
<tr>
<th>Preference</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>General display option</strong></td>
<td>Select <strong>Highlight Current Line</strong> and select a color to highlight the row with the cursor (also called the caret).</td>
</tr>
<tr>
<td></td>
<td>Select <strong>Show line numbers</strong> to display line numbers along the left edge of the Editor window.</td>
</tr>
<tr>
<td></td>
<td>Select <strong>Enable data tips in edit mode</strong> to display data tips when you are editing a MATLAB code file. (Data tips are always enabled in debug mode.)</td>
</tr>
</tbody>
</table>
### Editor/Debugger Tab Preferences

You can specify the size of tabs and indents and details about how tabs behave in the Editor.

On the *Home* tab, in the *Environment* section, click 📝 Preferences. Select *Editor/Debugger > Tab*, and then adjust preference options as described in the table below.

<table>
<thead>
<tr>
<th>Option</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Tab size</strong></td>
<td>Specify the amount of space inserted when you press the <strong>Tab</strong> key.</td>
</tr>
<tr>
<td></td>
<td>When you change the <strong>Tab size</strong>, it changes the tab size for existing lines in that file, unless you also select <strong>Tab key inserts spaces</strong>.</td>
</tr>
<tr>
<td><strong>Indent size</strong></td>
<td>Specify the indent size for <em>smart indenting</em>. Smart indenting is one of the “Editor/Debugger Language Preferences” on page 7-5.</td>
</tr>
</tbody>
</table>
### Editor/Debugger Preferences

#### Option Usage

<table>
<thead>
<tr>
<th>Option</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tab key inserts spaces</td>
<td>Select to insert a series of spaces when you press the Tab key. Otherwise, a tab acts as one space whose length is equal to the Tab size.</td>
</tr>
<tr>
<td>Emacs-style Tab key smart indenting</td>
<td>Specifies an indenting style similar to the style that the Emacs editor uses. Lines indent according to smart indenting preferences when you position the cursor in a line or select a group of lines, and then press the Tab key. Smart indenting is one of the “Editor/Debugger Language Preferences” on page 7-5. If you select this preference, you cannot insert tabs within a line.</td>
</tr>
</tbody>
</table>

#### Editor/Debugger Language Preferences

You can specify how various languages appear in the Editor. MATLAB applies language preferences based on the file extension of the file open in the Editor.

On the Home tab, in the Environment section, click Preferences. Select Editor/Debugger > Language, and then adjust preference options as described in the table below.

**Note** Not all preferences are available for all languages.
<table>
<thead>
<tr>
<th>Preference</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Language</td>
<td>Select the language for which you want to set preferences.</td>
</tr>
<tr>
<td>Syntax highlighting</td>
<td>Select <strong>Enable syntax highlighting</strong> to have the Editor use different colors for different language constructs. Then, adjust the colors you want to use for each language element.</td>
</tr>
<tr>
<td></td>
<td>Access color options for the MATLAB language by clicking <strong>Set syntax colors</strong>.</td>
</tr>
<tr>
<td></td>
<td>For all other languages, color options appear under <strong>Enable syntax highlighting</strong>.</td>
</tr>
<tr>
<td></td>
<td>For details, see “Syntax Highlighting” on page 3-23.</td>
</tr>
<tr>
<td>Variable and function renaming</td>
<td>Select <strong>Enable automatic variable and function renaming</strong> to have MATLAB prompt you to rename all instances of a function or variable in a file when you rename a function or variable.</td>
</tr>
<tr>
<td>MATLAB Language only</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Comment formatting</td>
<td>In the <strong>Maximum column width</strong> field, enter the maximum number of characters you want to allow in a line of comments, and then select where you want counting to begin.</td>
</tr>
<tr>
<td>MATLAB Language only</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Start from beginning of line</strong> when the absolute width of the comments is important. For example, set 75 columns from the start of the line to match the width that fits on a printed page when you use the default font for the Editor.</td>
</tr>
<tr>
<td>Preference</td>
<td>Usage</td>
</tr>
<tr>
<td>------------</td>
<td>-------</td>
</tr>
<tr>
<td></td>
<td>• <strong>Start from beginning of comment</strong> when comments are indented, and you want each block of comments to have a consistent indent and width.</td>
</tr>
<tr>
<td></td>
<td>Select <strong>Wrap comments automatically while typing</strong> to automatically wrap comments at the <strong>Maximum column width</strong> value when you type comments in an Editor document. If you clear this option, you can still wrap comments manually, as described in “Add Comments to Programs”.</td>
</tr>
<tr>
<td><strong>Indenting</strong></td>
<td>Select <strong>Apply smart indenting while typing</strong> to automatically:</td>
</tr>
<tr>
<td></td>
<td>• Indent the body of loops within the start and end of the loop statement.</td>
</tr>
<tr>
<td></td>
<td>• Align subsequent lines with lines you indent using tabs or spaces.</td>
</tr>
<tr>
<td></td>
<td>• Indent functions as specified with the <strong>Function indenting format</strong> option.</td>
</tr>
<tr>
<td></td>
<td>This is called <strong>smart indenting</strong>. You also can manually apply smart indenting after you type the code. For more information, see “Indenting Code”.</td>
</tr>
<tr>
<td></td>
<td>Select an option from <strong>Function Indentning Format</strong> (MATLAB Language only) to specify how functions indent in the Editor, as follows:</td>
</tr>
<tr>
<td></td>
<td>• <strong>Classic</strong> — The Editor aligns the function code with the function declaration.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Indent nested functions</strong> — The Editor indents the function code within a nested function.</td>
</tr>
</tbody>
</table>
Editor Preferences

Preference Usage

- **Indent all functions** — The Editor indents the function code for both main and nested functions.

  For more information and examples of each indenting format, see “Indenting Code”.

| File extensions | Add one or more file extensions to associate with the Language. The preferences you set for that language apply to all files with the listed extensions. |

Editor/Debugger Code Folding Preferences

Code folding enables you to expand and collapse blocks of MATLAB code that you want to hide when you are not currently working on them.

On the Home tab, in the Environment section, click Preferences. Select Editor/Debugger > Code Folding, and then adjust preference options as described in the table below.

For examples and detailed information about code folding, see “Code Folding — Expand and Collapse Code Constructs”.

<table>
<thead>
<tr>
<th>Option</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable Code Folding</td>
<td>Specifies whether you want code folding enabled for the programming constructs that have their corresponding Enable check box selected.</td>
</tr>
<tr>
<td>Enable</td>
<td>Specifies whether you want code folding enabled for the corresponding Programming Construct.</td>
</tr>
</tbody>
</table>
Editor/Debugger Preferences

Option Usage

If you select this option for any construct, but clear the Enable Code Folding option, the construct will not have code folding enabled.

Fold Initially

Specifies whether the corresponding Programming Construct displays collapsed (folded) the first time that you open a MATLAB file.

Editor/Debugger Autosave Preferences

You can specify if, when, and how you want MATLAB to automatically save files that are open in the Editor.

On the Home tab, in the Environment section, click Preferences. Select Editor/Debugger > Autosave, and then adjust preference options as described in the table below.

<table>
<thead>
<tr>
<th>Preference</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable autosave in the MATLAB Editor</td>
<td>Select to have MATLAB automatically save a copy of the files you are currently editing.</td>
</tr>
<tr>
<td>Save options</td>
<td>Save every n minutes specifies how often you want MATLAB to save a copy of the file you are editing.</td>
</tr>
<tr>
<td></td>
<td>Save untitled files saves a copy of new, untitled, files to Untitled.asv.</td>
</tr>
<tr>
<td></td>
<td>When there is more than one untitled file, each additional file is saved to Untitledn.asv (where n is an integer value).</td>
</tr>
<tr>
<td></td>
<td>For details, see “Autosaving Files”.</td>
</tr>
<tr>
<td>Close options</td>
<td>Automatically delete autosave files directs MATLAB to delete the autosave file when you close the source file in the Editor.</td>
</tr>
</tbody>
</table>
### Editor Preferences

#### Preference

<table>
<thead>
<tr>
<th>Preference</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>File name</strong></td>
<td>Select the naming convention you want MATLAB to use for autosave files. For example:</td>
</tr>
<tr>
<td></td>
<td>• If you specify <strong>Replace with extension:</strong> asv, the autosave file for <code>filename.m</code> is <code>filename.asv</code></td>
</tr>
<tr>
<td></td>
<td>• If you specify <strong>Append file name with</strong> ~, the autosave file for <code>filename.m</code> is <code>filename.m~</code></td>
</tr>
<tr>
<td><strong>Location</strong></td>
<td><strong>Source file directories</strong> specifies that you want autosave files stored in the same folder as the files being edited.</td>
</tr>
<tr>
<td></td>
<td><strong>Single directory</strong> specifies that you want autosave files stored in a single folder. Specify the full path to that folder and be sure you have write permissions for it.</td>
</tr>
</tbody>
</table>
Code Analyzer Preferences

In this section...

“Code Analyzer Preferences” on page 7-11
“Searching Messages in the Code Analyzer Preferences Dialog Box” on page 7-12

Code Analyzer Preferences

You can change how Code Analyzer messages appear in the Editor. With a few exceptions, these preferences apply to messages in the Editor, the MATLAB Function Block Editor (if your products use that tool), and the Code Analyzer Report.

On the Home tab, in the Environment section, click Preferences. Select Code Analyzer, and then adjust preference options as described in the table below.

<table>
<thead>
<tr>
<th>Option</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enabled Integrated Warning and Error Messages</td>
<td>Specify whether you want to display Code Analyzer message indicators, such as the underlining of code and the message indicator bar, for documents open in the Editor. For more information, see “Automatically Check Code in the Editor — Code Analyzer”.</td>
</tr>
<tr>
<td>Underlining</td>
<td>Specify the type of coding issues that you want to have underlined. Regardless of the underlining menu option you choose, the Editor marks errors and warnings in the message indicator bar.</td>
</tr>
<tr>
<td>Autofix</td>
<td>Provides a link to a preference panel that enables you to adjust the color highlighting errors and warnings that MATLAB can autofix. You trigger autofix by clicking the Fix button in a Code Analyzer message.</td>
</tr>
</tbody>
</table>
Editor Preferences

<table>
<thead>
<tr>
<th>Option</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Active Settings</strong></td>
<td>Select the set of message settings to use. Click the down arrow to select or browse to a previously saved settings file.</td>
</tr>
<tr>
<td><strong>Actions button</strong></td>
<td>Click to open a menu that enables you to select:</td>
</tr>
<tr>
<td></td>
<td>• <strong>Save as</strong> — Saves the current Code Analyzer message settings to a file.</td>
</tr>
<tr>
<td></td>
<td>The default location for settings is the MATLAB preferences folder (the folder returned when you run <code>prefdir</code>).</td>
</tr>
<tr>
<td></td>
<td>• <strong>Restore Defaults</strong> — Restores default Code Analyzer message settings.</td>
</tr>
<tr>
<td><strong>Search field</strong></td>
<td>Searches the list of Code Analyzer messages that display below the search field. For details, see “Searching Messages in the Code Analyzer Preferences Dialog Box” on page 7-12.</td>
</tr>
<tr>
<td><strong>Code Analyzer message settings</strong></td>
<td>Select or clear messages to enable or suppress their appearance in your Editor documents.</td>
</tr>
<tr>
<td></td>
<td>To suppress a message on a line-by-line or file-by-file basis, see “Adjust Code Analyzer Message Indicators and Messages”.</td>
</tr>
</tbody>
</table>

**Searching Messages in the Code Analyzer Preferences Dialog Box**

You can search the list of Code Analyzer messages in the Preferences dialog box to display only those messages that are currently of interest to you. Use any combination of the methods that the following table presents.

**Note** If you do not have the MATLAB Compiler™ installed, the Code Analyzer preferences pane does not display the **MATLAB Compiler (deployment) messages** category.
<table>
<thead>
<tr>
<th>To See a List of Messages ...</th>
<th>Perform this action...</th>
<th>Example Scenario</th>
</tr>
</thead>
<tbody>
<tr>
<td>Containing a given string in the:</td>
<td>Type the string in the search field.</td>
<td>You recall seeing a message containing a certain string that you want to review, but you cannot remember the exact message text. For example, type <code>com</code> in the search field to display those messages that contain that string in the short message, extended message, or message ID.</td>
</tr>
</tbody>
</table>
| • Short message  
• Extended message  
• Message category  
• Message ID | | |
| Corresponding to a given message ID | Type `msgid:` followed by the message ID in the search field. | You are reviewing the code that someone else wrote and you want to see the message that corresponds to a suppressed one using the `%#ok<AGROW>` directive. Type `msgid:agrow` in the search field. Messages IDs containing AGROW display as links. Click each link for more information about the message. Not all Code Analyzer messages have additional information. These messages do not appear as links. |
| That you can set using Code Analyzer preferences | Click the down arrow to the right of the search field, and then click **Show All.** | You want to see the complete list of messages after you have searched the messages for a given string or search menu option. |
### To See a List of Messages ...

<table>
<thead>
<tr>
<th>Performed this action...</th>
<th>Example Scenario</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Different from the default setting (of enabled or disabled)</strong></td>
<td>A coworker gave you a settings file and you want to review each message that the coworker changed from its default setting.</td>
</tr>
</tbody>
</table>
| Click the down arrow to the right of the search field, and then click **Show Messages Modified from Default.**  
A gray dot precedes a message with a setting different from the default. For example:  
● □ △ DATENUM(NOW) | |
| **In a given category** | You want to review messages that describe coding practices that make it difficult for others to use your code. |
| Click the down arrow to the right of the search field, click **Show Messages in Category**, and then click the category you want. | |
| **That are warnings** | You recall previous warnings that your code generated, but you cannot remember enough details to use the search field to find it. You want to skim all the warning messages to find a particular one of interest. |
| Click the down arrow to the right of the search field, and then select **Show All Warnings**. An exclamation point in a yellow triangle △ indicates a warning message. | |
To See a List of Messages ...

**Perform this action...**

Click the down arrow to the right of the search field, and then select **Show All Errors**. By default, an X in a red dot indicates an error message,  

**Example Scenario**

You want to find a message elicited by a script you worked on previously. All you can recall is that it was an error and it involved *parfor*.

Click the down arrow to the right of the search field, and then select **Show All Errors**. Then, type a space and *parfor* in the search field.

The Code Analyzer preference pane displays only error messages that contain the word *parfor*.

**Are disabled**

Click the down arrow to the right of the search field, and then select **Show Disabled Messages**.

**Example Scenario**

You want to see the messages that are disabled by default or you have previously disabled.

---

**Example of Searching Messages**

To display Code Analyzer error messages that contain the string *variable* and are disabled:

1. Click the down arrow in the search field, and then select **Show All Errors**.

   The search field contains the string `severity:error`.

2. At the end of the string `severity:error`, press the **Space** key, and then type *variable*.

3. Click the down arrow in the search field and select **Show Disabled Messages**.

   The search field now contains the string, `severity:error variable enabled:false`. Only the messages that fulfill those requirements appear in the Preferences pane.
To restore the list of all messages, click the clear search button \( \times \).
Internationalization

- “How the MATLAB Process Uses Locale Settings” on page 8-2
- “Setting the Locale” on page 8-4
- “Troubleshooting I18n Messages and Settings” on page 8-9
How the MATLAB Process Uses Locale Settings

A locale is part of the user environment definition. It defines language, territory, and codeset, which is a coded character set. The MATLAB process uses the user-specified locale name on all platforms. MATLAB also reads the user-specified UI language name, and uses it to select localized resources in the specified language. By using this feature, you can select localized resources in US-English. The user-specified UI language setting also controls language and country settings of the Sun™ Java Virtual Machine (JVM) software.

To see what your current settings are, use the instructions in “Setting Locale on Windows Platforms” on page 8-4, “Setting Locale on Linux Platforms” on page 8-7, or “Setting Locale on Macintosh Platforms” on page 8-8. For more information, “Troubleshooting I18nMessages and Settings” on page 8-9.

Consider the following when choosing your locale settings.

- **Default Locale Setting** — If the user-specified locale is not supported, MATLAB uses the default locale en_US.US-ASCII.

- **UI Language Setting** — The UI language setting should be set to either the same language as the user-specified locale or to US-English. Otherwise, non-7-bit ASCII characters might not display properly.

- **Supported Character Set** — MATLAB supports the character set specified by the user locale setting. However, MATLAB might not properly handle character codes greater than 2 bytes.

- **Script Compatibility** — Non-7-bit ASCII characters in MATLAB scripts created with one locale setting might not be compatible with a different locale setting.

  For example, if you create a script with the ja_JP.UTF-8 locale setting, the script might not be compatible when executed on a platform with the ja_JP.eucJP locale setting.

- **Numeric Format Uses C Locale** — MATLAB reads the user locale for all categories except for the LC_NUMERIC category. This category controls numeric data formatting and parsing. MATLAB always sets LC_NUMERIC to the C locale. For more information, see “Numbers Display Period for Decimal Point” on page 8-10.
• **Platform-Specific Localized Formats** — MATLAB usually uses platform-neutral localized formats and rules. You can, however, use the operating system short date format to display files, as described in “Customizing the Column Display” on page 6-17.

**Windows Platform-Specific Behavior**

The user locale and system locale must be the same value on the Microsoft Windows platform. If these values are not the same, you might see garbled text or incorrect characters. For information on controlling these settings, see “Setting Locale on Windows Platforms” on page 8-4.

**Macintosh Platform-Specific Behavior**

On the Apple Macintosh OS X platform, MATLAB reads the user locale setting and the user UI language setting. For information on controlling these settings, see “Setting Locale on Macintosh Platforms” on page 8-8. MATLAB ignores the LANG environment variable and the Terminal application locale setting.

MATLAB automatically chooses a codeset for each combination of language and territory on the Mac OS X platform. If you customize the locale setting on OS X, MATLAB ignores the customized portion.
Setting the Locale

<table>
<thead>
<tr>
<th>In this section...</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Setting Locale on Windows Platforms” on page 8-4</td>
</tr>
<tr>
<td>“Setting Locale on Linux Platforms” on page 8-7</td>
</tr>
<tr>
<td>“Setting Locale on Macintosh Platforms” on page 8-8</td>
</tr>
</tbody>
</table>

Setting Locale on Windows Platforms
MATLAB software uses the system locale and user locale on Windows platforms:

- “Setting User Locale on Windows 7 Platforms” on page 8-4
- “Setting System Locale on Windows 7 Platforms” on page 8-5
- “Setting User Locale on Windows Vista Platforms” on page 8-5
- “Setting System Locale on Windows Vista Platforms” on page 8-5
- “Setting User Locale on Windows XP Platforms” on page 8-6
- “Setting System Locale on Windows XP Platforms” on page 8-6

Setting User Locale on Windows 7 Platforms


2. Open Formats tab.

3. Select a target locale from the Format: drop-down list.

Note The user locale and system locale must be the same value on the Microsoft Windows platform.
Setting the Locale

Setting System Locale on Windows 7 Platforms

1 Select Start -> Control Panel -> Clock, Language, and Region -> Regional and Language.

2 Open Administrative tab.

3 Look in the Language for non-Unicode programs section.

4 Click Change system locale... button.

5 Select a target locale from the Current system locale: drop-down list.

6 Reboot the system.

Note When you change the system locale, you must reboot your system; otherwise, you might see unexpected locale-setting behaviors.

Setting User Locale on Windows Vista Platforms

1 Select Start -> Control Panel -> Regional and Language Options.

2 Open Formats tab.

3 Select an item from the drop-down list.

Note The user locale and system locale must be the same value on the Microsoft Windows platform.

Setting System Locale on Windows Vista Platforms

1 Select Start -> Control Panel -> Regional and Language Options.

2 Open Administrative tab.

3 Click Change system locale... button.
4 Select an item from the drop-down list.

5 Reboot the system.

**Note** When you change the system locale, you must reboot your system; otherwise, you might see unexpected locale-setting behaviors.

**Setting User Locale on Windows XP Platforms**

1 Select Start -> Control Panel -> Regional and Language Options.

2 Open Regional Options tab.

3 Select an item from the drop-down list.

**Note** The user locale and system locale must be the same value on the Microsoft Windows platform.

**Setting System Locale on Windows XP Platforms**

1 Select Start -> Control Panel -> Regional and Language Options.

2 Open Advanced tab.

3 Select an item from the drop-down list.

4 Reboot the system.

**Note** When you change the system locale, you must reboot your system; otherwise, you might see unexpected locale-setting behaviors.
Setting Locale on Linux Platforms

Linux platforms manage locale settings with six locale categories. These are the same categories used by C standard library functions.

The following locale categories are available:

- **LC_CTYPE** controls character data manipulations.
- **LC_COLLATE** controls character collation/sorting operations.
- **LC_TIME** controls date/time data formatting or parsing.
- **LC_NUMERIC** controls numeric data formatting or parsing.
- **LC_MONETARY** controls monetary data formatting or parsing.
- **LC_MESSAGES** controls the user UI language.

Setting User Locale and User UI Language

Use the `LANG` environment variable to specify a single locale for all locale categories. The locale specified with this variable might be partially or entirely over-written by other environment variables.

Use the environment variables `LC_CTYPE`, `LC_COLLATE`, `LC_TIME`, `LC_NUMERIC`, and `LC_MONETARY` to specify a locale for a particular category.

Use the `LC_ALL` environment variable to over-write all locales specified with other environment variables. If a single locale has to be set to all locale categories, use `LANG` instead of `LC_ALL`.

Configuring Fonts to Display Asian Characters

On some Linux systems, to properly display Asian characters in the MATLAB Desktop, you must configure the font with the Java Runtime Environment (JRE™). If you previously configured fonts for your system, you must also make the configuration changes for the JRE distributed with MATLAB.

To configure, make a symbolic link between your font and the MATLAB font fallback directory. For example, to use the Kochi font, at the Linux system prompt type:

```
ln -s /usr/share/fonts/truetype/kochi
```
where `matlabroot` is the folder where you installed MATLAB.

Alternatively, edit the `fontconfig.properties` file. See your Java documentation for information about this file.

**Setting Locale on Macintosh Platforms**

The Macintosh OS X platform manages the user locale setting and the user UI language setting.

**Setting User Locale**

1. Select **System Preferences** -> **Language & Text**
2. Open **Formats** tab
3. Select an item from the **Region** pop-up menu

**Setting UI Language**

1. Select **System Preferences** -> **Language & Text**
2. Open **Language** tab
3. Drag an item to the top of the **Languages** list
Troubleshooting I18n Messages and Settings

The term I18n is an abbreviation for internationalization, where 18 stands for the number of letters between the i and the n.

<table>
<thead>
<tr>
<th>In this section...</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Asian Characters Incorrectly Displayed on Linux Systems” on page 8-9</td>
</tr>
<tr>
<td>“Characters Incorrectly Displayed on Windows Systems” on page 8-10</td>
</tr>
<tr>
<td>“datenum Might Not Return Correct Value” on page 8-10</td>
</tr>
<tr>
<td>“Numbers Display Period for Decimal Point” on page 8-10</td>
</tr>
<tr>
<td>“MATLAB Displays Messages in English” on page 8-11</td>
</tr>
<tr>
<td>“File or Folder Names Incorrectly Displayed” on page 8-11</td>
</tr>
</tbody>
</table>

Asian Characters Incorrectly Displayed on Linux Systems

On some Linux systems, to properly display Asian characters in the MATLAB Desktop, you must configure the font with the Java Runtime Environment (JRE). If you previously configured fonts for your system, you must also make the configuration changes for the JRE distributed with MATLAB.

To configure, make a symbolic link between your font and the MATLAB font fallback directory. For example, to use the Kochi font, at the Linux system prompt type:

```
ln -s /usr/share/fonts/truetype/kochi matlabroot/sys/java/jre/glnxa64/jre/lib/fonts/fallback
```

where `matlabroot` is the folder where you installed MATLAB.

Alternatively, edit the `fontconfig.properties` file. See your Java documentation for information about this file.
**Characters Incorrectly Displayed on Windows Systems**

The user locale and system locale must be the same value on the Microsoft Windows platform. If these values are not the same, you might see garbled text or incorrect characters. For information on controlling these settings, see “Setting Locale on Windows Platforms” on page 8-4.

**datenum Might Not Return Correct Value**

To ensure the correct calculation of functions using date values associated with files and folders, replace `datenum` function calls with the use of the `dir` function `datenum` field.

For example, look at the modification date of your MATLAB license.txt file:

```matlab
cd(matlabroot)
f=dir('license.txt')
```

MATLAB displays information similar to:

```matlab
f =
    name: 'license.txt'
    date: '10-May-2007 17:48:22'
    bytes: 5124
    isdir: 0
    datenum: 7.3317e+005
```

If your code uses the `date` field of the `dir` command, similar to:

```matlab
n=datenum(f.date);
```

replace it with the `datenum` field:

```matlab
n=f.datenum;
```

**Numbers Display Period for Decimal Point**

MATLAB uses a period for a decimal point, regardless of the format specified by the user locale. For example, the value of pi can be displayed as 3,1416 or 3.1416, depending on the format used by a locale. MATLAB always displays 3.1416.
The MATLAB language reserves the use of commas to the cases described in the “Comma —,” topic of the Programming Fundamentals Symbol Reference.

**MATLAB Displays Messages in English**
MATLAB displays messages in English, regardless of the UI language setting, except when running in a Japanese Microsoft Windows environment.

**File or Folder Names Incorrectly Displayed**
On Windows and Linux platforms, characters used in file or folder names must be in the supported character set. See **Supported Character Set** in “How the MATLAB Process Uses Locale Settings” on page 8-2.

On Macintosh platforms, for files and folders used by MATLAB, characters in the file or folder name must be in the 7-bit ASCII character set.
Symbols and Numerics

, after functions 3-10
; after functions 3-10
! function 6-43
    argument length restrictions 6-44

A
absolute path name 6-6
    copying 6-6
accelerators
    Command Window 2-15
accelerators, keyboard 2-15
activate license 2-47
antialiasing
    desktop fonts 2-63
AppleScript
    running from MATLAB 6-44
archive files
    adding files to 6-35
    creating 6-33
    extracting files from 6-34
automatic completion of statement
    Command Window 3-25

B
bang (!) function 6-43
base workspace 5-2
batch mode for starting MATLAB 1-21
beep
    preferences 3-21
binary files
    comparing 6-61
bookmarks
    in Help browser 4-7
Boolean searching in Help browser 4-5
browser
    for Web 2-42
bugs, reporting to MathWorks 4-9

C
caching
    search path 6-4
case sensitivity
    of file names 6-6
    of path names 6-6
cell arrays
    editing 5-3
changing
    search path 6-74
character set
    preference for MAT-files 2-56
closing
    MATLAB 1-7
Code Analyzer preferences 7-11
    setting 7-11
code folding preferences in files 7-8
collapsing
    code in files 7-8
colors
    general preferences 2-8
    indicators for syntax 3-23
columns
    customizing in Current Folder browser 6-17
command flags 1-18
Command History
    about 3-29
    changing date format in 6-17
    file 3-29
    find entry by letter 3-15
    preferences 3-31
    running functions from window 3-30
command name completion
    Command Window 3-25
command switches 1-18
Command Window
    getting started message bar 3-19
    paging of output in 3-11
    preferences 3-19
    scroll buffer 3-19
Index

width 3-19
commands
  to operating system 6-43
comments
  color indicators 2-8
comparing
  directories 6-47
  files 6-47
Comparison Tool
  features of 6-62
completing statements automatically
  Command Window 3-25
compression
  MAT-files and Fig-Files 2-56
configuration, desktop 2-11
confirmation dialog boxes
  preferences 2-57
copying
  files and folders 6-39
  using functions 6-38
crash 1-9
creating
  files and folders
    using functions 6-36
current folder
  at startup for MATLAB 1-12
  changing 6-2
  viewing 6-2
Current Folder browser 6-11
  asterisks and 6-16
  changing date format in 6-17
  columns 6-16
  details panel 6-18
  preferences 6-13
  refresh display 6-14
  running scripts from 6-42
  viewing image thumbnails in 6-18

D
date format
changing in Command History window 6-17
changing in Current Folder browser 6-17
deactivate license 2-47
defaults
  preferences for MATLAB 2-52
  setting in startup file for MATLAB 1-19
deleting
  files and folders
    using Current Folder browser 6-37
    using functions 6-38
delimiter
  matching in Editor 3-21
  preferences for matching 3-21
description for file
  viewing in Current Folder browser 6-18
desktop
  configuration 2-11
  description 2-2
  font preferences for 2-62
development environment for MATLAB 2-2
diagnostics
  startup
    Macintosh 1-7
difference reporting for files 6-47
directories
  comparing 6-47 6-49
displaying
  output 3-10
do not show again
  preferences 2-57

E
encoding
  preference when saving 2-56
ending MATLAB 1-7
environment settings at startup 1-19
environment variables 6-44
error logs 1-9
errors
color indicators 2-8

evaluating
  selection in Command History window 3-30
  selection in Command Window 6-41

exact phrase
  Help browser search 4-5

exe 6-43

executables
  running from MATLAB 6-43

exection
  stopping 3-12

exiting
  saving workspace 1-9

exit MATLAB 1-7

confirmation 1-8

expanding
  code in files 7-8

searching contents of 6-25

Find Files dialog box 6-25

finding
  files and folders
    by name 6-24
  files by name and content 6-25
  text in Command History window 3-16
  text in Command Window 3-13

finish.m file running when exiting 1-9

firewall
  settings to work through 2-44 2-67

flags
  for startup 1-18

folders
  comparing 6-49
  creating 6-31

font
  antialiasing in desktop 2-63
  preferences in MATLAB 2-62
  smoothing in desktop 2-63

format
  preferences 3-19

full path name 6-6

function name
  automatic completion
    Command Window 3-25

function workspace 5-2

functions
  color indicators 2-8
  naming
    avoiding conflicts 6-68

file name
  case sensitivity 6-6

files
  comparing 6-47
  creating
    from Command History window 3-30

finding by name 6-25

log 1-20

naming
  avoiding conflicts 6-68

opening as text files 6-40

opening outside MATLAB 6-40

search path 6-65

HDF

preference when saving 2-56

Help browser
  viewing page location 4-8

hidden files
Index

viewing 6-14

history
  automatic log file 1-20
history file 3-29
history of statements 3-29
history.m file 3-29
hot keys 2-15
desktop 2-15
  Variable Editor 5-8
HTML viewer in MATLAB 2-42

I

image files
  previewing in Current Folder Browser 6-18
import
  files for use with MATLAB 6-71
include
  files with MATLAB 6-71
indenting
  in Command Window 3-23
initiation (init) file for MATLAB 1-19
Internet
  proxy server settings 2-44 2-67

J

Java Heap
  preferences 2-59
Java VM
  starting without 1-21

K

keyboard shortcuts
  Command Window 2-15
  Variable Editor 5-8
keywords
  color indicators 2-8
  matching in Editor 3-21

L

license information 4-16
license management 2-47
line wrapping 3-19
log
  automatic 1-20
  file 1-20
  statements 3-29
logfile startup option 1-20
login
  remote on Macintosh 1-7

M

Macintosh
  startup
    remote login 1-7
MAT-files
  comparing 6-59
  compatibility 2-56
  compression options 2-56
  creating 5-9
  defined 5-9
  preferences 2-56
  updating using Current Folder Browser 6-32
viewing variables without loading 6-18
matched delimiters
  preferences 3-21
matching parentheses
  in Editor 3-21
MATLAB
  exiting 1-7
    confirmation 1-8
  search path 6-71
MATLAB code files
  file association (Windows) 1-3
running
  at startup 1-21
matlab folder 1-13
MATLAB installations
search path with 6-79
matlabrc.m, startup file 1-19
minimize
   Windows startup option 1-20
model files
   description in Current Folder browser 6-18
more 3-11
moving
   files and folders 6-39
multidimensional arrays
   editing 5-3
multithreading
   turning off 1-21

N
name clashes 6-68
naming
   functions and variables
      avoiding conflicts 6-68
nojvm startup option 1-21
numeric format
   preferences 3-19

O
objects
   editing 5-3
operating system commands 6-43
operators
   searching for 4-5
options
   shutdown 1-9
   startup 1-18
output
   display
      hidden 3-10
hiding 3-10
paging 3-11
spaces per tab 3-20
   spacing of 3-19
   suppressing 3-10

P
paging in the Command Window 3-11
parentheses
   matching 3-21
parentheses matching
   preferences 3-21
partial
   path name 6-9
partial word
   Help browser search 4-5
path
   See search path
PATH environment variable 6-45
path name 6-6
   absolute 6-6
   case sensitivity 6-6
   length 6-8
   partial 6-9
   relative 6-6
   See also absolute path name; full path name;
      relative path name
path names
   relative 6-6
pathdef.m
   location 6-72
PDF
   reader, preference for Help browser 4-13
Perl variables
   passing
      at startup 1-21
preferences
   Current Folder browser 6-13
problems, reporting to MathWorks 4-9
product filter in Help browser
   preference 4-13
program control blocks
   code folding and 7-8
program elements
  going to 6-21
program files
  help
    viewing in Current Folder browser 6-18
  naming
    avoiding conflicts 6-68
    viewing help for 6-18
programs
  running from MATLAB 6-43
  stopping execution of 3-12
proxy server settings 2-44 2-67

Q
  quitting MATLAB 1-7

R
  recovering deleted files 6-37
  recycle 6-37
  refresh
    Current Folder browser 6-14
  relative path 6-6
  release
    latest 2-48
  remote login
    Macintosh 1-7
  renaming
    files and folders
      using Current Folder browser 6-36
      using functions 6-37
  requirements
    MATLAB 1-1

S
  saving
    MAT-files
      preferences 2-56
    workspace upon exiting 1-9
  script for startup 1-19
  scroll buffer for Command Window 3-19
  scrolling in Command Window 3-11
search path
  adding folders to 6-75
  behavior when changing folders 6-82
  changing 6-74
  default 6-65
  description 6-65
  problems and recovering 6-80
  saving 6-78
  using 6-71
  using with different MATLAB installations 6-79
  viewing 6-73
searching
  for files by name and content 6-25
Help browser
  Boolean 4-5
  exact phrase (" ") 4-5
  wildcard (*) or partial word 4-5
  in Current Folder browser
    by name 6-24
    typeahead 6-24
  special characters 4-5
  text
    Command History window 3-16
    Command Window 3-13
  segmentation violation 1-9
  segv 1-9
  semicolon (;)
    after functions 3-10
  session
    automatic log file 1-20
  session log
    Command History 3-29
  shadowed functions 6-68
  shell escape 6-43
  shortcut
    for MATLAB in Windows 1-2
keys in MATLAB  2-15
shortcut keys
   Variable Editor  5-8
shortcuts
   creating
      from Command History window  3-30
      toolbar  3-17
shutdown
   MATLAB  1-7
   options  1-9
Simulink models
   viewing complete descriptions of  6-18
singleCompThread startup option  1-21
spacing
   output in Command Window  3-19
   tabs in Command Window  3-20
special characters
   searching for  4-5
splash screen
   startup option  1-21
stack
   viewing  5-2
starting MATLAB
   DOS  1-2
   Linux  1-6
   Windows  1-2
startup
   diagnostics
      Macintosh  1-7
   files for MATLAB  1-19
   folder for MATLAB  1-12
   Macintosh, remote login  1-7
   options for MATLAB  1-18
   script  1-19
startup.m
   location  1-20
   startup file  1-19
strings
   color indicators  2-8
   saving as Unicode  2-56
structures
   editing  5-3
style preferences for text  2-62
support
   technical  4-9
suppressing output  3-10
switches
   for startup  1-18
symbols
   searching for  4-5
syntax
   color indicators  2-8
   coloring and indenting  3-23
system browser
   UNIX  2-45
system environment variables  6-44
system path for UNIX  6-45
system requirements
   MATLAB  1-1
system Web browser  2-42

T

tab
   spacing in Command Window  3-20

tab completion
   Command Window  3-25
Technical Support
   contacting  4-9
temporary folder
   for deleted files  6-37
text
   preferences in MATLAB  2-62
text files
   comparing  6-53
threads
   turning off multithreading  1-21
tmp/MATLAB_Files folder  6-38
token matching
   preferences  3-21

Index-7
toolbars
  customizing 2-10
  shortcuts 3-17

toolbox path cache
  preferences 1-24

tools in desktop
  description 2-2

U
Unicode
  preference when saving 2-56
UNIX
  system path 6-45
updates
  to newer versions 2-48
utilities
  running from MATLAB 6-43

V
Variable Editor
  decimal separator 5-15
  keyboard shortcuts 5-8
  preferences 5-15
  size limitations 5-4
variables
  naming
    avoiding conflicts 6-68
    saving 5-9
    viewing 6-18
version 2-48
  information for MathWorks products 4-16

W
Web
  preferences 2-67
  proxy server settings 2-44 2-67
Web Browser
  in MATLAB 2-42
wildcard (*)
  Help browser search 4-5
windows in desktop
  about 2-2
  arrangement 2-11
workspace
  base 5-2
  functions 5-2
  saving 5-9
  tool 5-2
  viewing 5-3
Workspace browser
  description 5-2
wrapping
  lines in Command Window 3-19

Z
zip files
  adding files to 6-35
  creating 6-33
  extracting files from 6-34
  viewing contents of 6-33