The Basics of MEL Commands

CHAPTER CONTENTS

15.1 Can I Use MEL Without Scripting? .............................................................. 15-2
15.2 Command Line and Command Feedback Line .............................................. 15-2
15.3 Command Shell ........................................................................................ 15-3
15.4 Script Editor ............................................................................................. 15-3
15.5 Script Editor Versus Command Shell ........................................................... 15-5
15.6 Script Editor’s Messages as MEL Code ....................................................... 15-5
15.7 Making a Shelf Button for a MEL Script .................................................... 15-7
15.8 Saving a MEL Script .................................................................................. 15-7
15.9 Seductive Dangers of the Status Message Area ........................................... 15-8
15.10 The whatIs Command ................................................................................ 15-9
15.11 Basic Structure of MEL Commands ........................................................... 15-10
15.12 Where to Find Information About Maya and MEL on the Internet .......... 15-11
15.12.1 Web Sites ......................................................................................... 15-11
15.12.2 Newsgroups ...................................................................................... 15-12
15.13 How to Use MEL Scripts Found on the Internet ....................................... 15-12
15.14 What to Remember About How to Use MEL Without Writing Scripts ....... 15-13

In this chapter, you will learn

- How to run a MEL command or script and see its result
- That Maya provides useful tools that let you capture scripts by working in the interface and then using the resulting MEL commands to build shelf buttons
- How to read MEL commands that you capture from the interface
- Where to find Maya and MEL information on the Internet
- How to use MEL scripts that you’ve found on the Internet.
15.1 **CAN I USE MEL WITHOUT SCRIPTING?**

Before you sit down to write a MEL script, there are two alternatives that may greatly simplify automating your task, depending on what you are trying to do.

First, in many cases, another Maya user may have already developed a script or a plug-in that does what you need. Many Maya users make such tools available on the Internet, and electronic bulletin boards can serve as a place to ask other Maya users for suggestions on how to implement your own solutions.

Second, you can log the MEL commands that correspond to your actions as you work in the interface. Reading the logged commands from the interface can save hours of hunting through the MEL reference documentation to find the particular commands that will do what you already know how to do in Maya’s interface.

Resist the temptation to imagine that the commands you log can be used without alteration, though. Once you’ve learned MEL, you can use these logged commands as inspiration, but you may only cause yourself problems if you use the commands without learning what they do and how they work.

Some keyboards have two keys labeled Enter; others have one labeled Return on the main keyboard and one labeled Enter on the numeric keypad.

When you’re working in the Script Editor, pressing Return or Enter on the main keyboard takes you to the next line, while pressing Enter on the numeric keypad executes either all the MEL code that’s been typed into the Script Editor (if nothing is selected) or whatever MEL code is selected in the Script Editor (if you’ve dragged over some MEL code to highlight it).

Should you choose to execute the entire contents of the Script Editor by pressing Enter on the numeric keypad without selecting anything, if the contents of the Script Editor execute successfully it will be deleted! If you select one or more lines of text in the Script Editor by dragging the mouse over it, when you press Enter that text will remain in the Script Editor. Because of this behavior, many MEL programmers always execute MEL commands by selecting the commands in the Script Editor that they wish to run before pressing Enter so that they remain in the editor after they’ve been executed.

15.2 **COMMAND LINE AND COMMAND FEEDBACK LINE**

Maya’s command line provides a simple and immediate way to run MEL commands. Click in the command line, type a MEL command, and press Enter.

Maya’s command feedback line, next to the command line, gives you the result from the most recent MEL command or other operation.

Try the following:

1. Choose File > New Scene (and, if you like, save your changes).
2. Click in the command line.
3. Type `sphere`, and press Enter.
A sphere named nurbsSphere1 should appear, and in the command feedback line you should see

Result: nurbsSphere1 makeNurbSphere1

The sphere command you just executed gives as its result a list of the nodes that it created. (We'll talk more about results and how you'll use them in MEL scripts later.) In this command's result, nurbsSphere1 is the name of the sphere's transform node, and makeNurbSphere1 is the sphere's history node. Note that this operation also creates a shape node called nurbsSphereShape1, but, in keeping with Maya's practice of hiding the shape nodes from you, the result does not show you the shape node's name.

15.3 COMMAND SHELL

On Irix and Windows platforms, Maya offers a different version of the command line with which even many experienced Maya users are unfamiliar. By choosing Window > General Editors > Command Shell… you can open a window that offers a Maya command line, but retains a convenient history of the commands you type and their results.

1. Choose Window > General Editors > Command Shell…

   The command shell window appears.

2. Type sphere, and press Enter.

   A sphere appears, along with the text indicating the name of the sphere that was created.

3. Press the up-arrow key.

   Pressing the up-arrow key takes you back to earlier commands that you have typed. This can be very helpful when you have to apply the same command repeatedly or you have at some point typed a complex command that is close to what you need to type again.

4. Press Enter again.

   Another sphere appears.

15.4 SCRIPT EDITOR

The Script Editor is a much more extensive interface for entering commands and editing scripts than the command shell. Within the Script Editor, you can execute commands immediately or develop long scripts that can be saved or executed all at once.
The Script Editor is divided into two parts: the status message area on the top and the command input area on the bottom. You can type commands and scripts into the command input area, execute them, and see the result in the status message area.

While you work in the interface, the interface also generates MEL commands and their results for what you do. These commands and results appear in the status message area even when the Script Editor window is hidden. Usually, the last line of the status message area is the result that is displayed in the command feedback line when the Script Editor is not open.

Now, we will repeat the same example that we have used above for the command line and the command shell:

1. Choose File > New Scene, and click on No to skip saving the old scene.
2. Open the Script Editor by clicking on the Script Editor button to the right of the command feedback line, or by using Windows > General Editors > Script Editor.…
3. Click in the command input area with the white background at the bottom. If you've been working through the previous examples, you should be able to see some of your earlier commands represented in the gray status message area above.
4. Type `sphere`, and press Enter on the numeric keypad.

Maya now creates a sphere in your scene. Your command, along with its result, appears in the status message area, as follows:

```
sphere;  // Result: nurbsSphere1 makeNurbSphere1 //
```

Note that your command is now deleted from the command input area. Pressing Enter on the numeric keypad, or holding down Ctrl and pressing Enter or Return on the main keyboard, executes everything in the command input area and deletes it from that part of the window.

Often, if you're editing a script and wish to execute a command quickly, or if you have a number of commands that you would like to keep typed into the Script Editor, you may prefer that Maya not delete the commands as you execute them. If you select a command and then press Enter, it will remain in the Script Editor after it is executed. For example:

1. Type `sphere`.
2. Press Enter on the main keyboard.
3. Type `cone`.
4. Press Enter.
5. Type `cylinder`.
6. Press Enter.
7. Double-click on `cone` to select the whole word, and press Enter on the numeric keypad.

Now, a cone appears, but your commands remain. If you like, you could continue executing those commands by selecting them and pressing Enter on the numeric keypad.
keypad. You can also select and execute multiple lines of text within the command input area; you’re not limited to a single command.

If you find that you like to keep useful commands around in the Script Editor, be careful not to press Enter when you have nothing selected! If you do, Maya will execute everything in the window and empty out the command input area. Using the command line or the command shell for running quick commands can be much less error prone, at the expense of some convenience.

### 15.5 SCRIPT EDITOR VERSUS COMMAND SHELL

Many Maya users always enter MEL commands in the Script Editor rather than the command shell, whether they are writing a script or entering commands to be executed right away. Here are a few advantages to using the Script Editor over the command shell:

- You can keep a repository of useful commands in the Script Editor and execute them repeatedly, or in any order you like.
- The Script Editor allows you to copy and paste parts of commands or other text from the history in the window.
- If a command you’ve run from the Script Editor strikes you as useful, you can select it and then drag it to a shelf, where it will remain as a button that you can click to run the same command again.

There are also disadvantages to running commands from the Script Editor:

- As pointed out above, because the Script Editor treats the Enter key on the main keyboard differently from the Enter key on the numeric keypad, it can be easy to erase some of your work by mistake.
- If you’re editing a complex script in the Script Editor, it can be easier to run one or two commands in the command shell rather than typing those commands into the Script Editor somewhere among your script’s code, running them, and then deleting them again.

### 15.6 SCRIPT EDITOR’S MESSAGES AS MEL CODE

One convenient property of the messages that appear in the status message area of the Script Editor is that they are valid MEL commands. Furthermore, as you perform many tasks in Maya, MEL commands that are equivalent to what you do in the interface appear in the status message area. Because these are valid MEL commands, you can do something in the interface, copy a few commands from the status message area in the Script Editor, and then later on run those commands to repeat what you had done earlier.

Looking back at the first Script Editor example, where you typed `sphere` and pressed Enter on the numeric keypad, you received this result in the status message area:

```mel
sphere;
// Result: nurbsSphere1 makeNurbSphere1 //
```
Note that Maya has added a semicolon (;) to your command. In a MEL script or an expression, each command must end with a semicolon. If you type a single command into the command input area, you need not place a semicolon at the end. However, if you type more than one command at a time and want to execute them all at once, you must separate them with semicolons. (For example, to make two spheres, you could type `sphere; sphere` and press Enter on the numeric keypad.)

Also, around the result, Maya has placed double slashes (//). In a MEL script or an expression, any line that starts with double slashes is ignored. In a script, this is useful to provide notes to yourself about how your script works or how it is to be used. Programmers usually refer to these notes as “comments,” and new programmers are often told to “comment their code” by peppering it with notes to remind themselves how their programs work when they return to them later.

The reason Maya presents your commands and results this way in the status message area is so that you can copy the text there into a MEL script to perform those same commands again. The semicolons allow MEL to know where one command ends and another begins, and the slashes allow MEL to know that the result lines are not instructions to Maya to do something but instead represent information meant for you, the animator, to read.

Leaving the Script Editor open, try the following exercise in the interface.

2. Make a sphere by choosing Create > NURBS Primitives > Sphere.
3. Click on the Move tool.
4. Move the sphere along the Z-axis a few units.
5. Click on the Script Editor’s title bar to bring it to the front again. (If the Script Editor window is now covered up by another Maya window, you may have to choose Raise Application Windows from the Windows menu or you can click on the Script Editor button again to display it.)

Now, the most recent five lines or so in the Script Editor will be similar to the following lines. (Depending on exactly how far you dragged the sphere, your numbers may be slightly different, and the line that starts with `sphere` will probably wrap around in a different place.)

```meli
file -f -new;
// Result: ./untitled //
sphere -p 0 0 0 -ax 0 1 0 -ssw 0 -esw 360 -r 1 -d 3 -ut 0 -tol 0.01 -s 8 -nsp 4 -ch 1;
objectMoveCommand;
move -r 0 0 3.001753;
```

These lines make up a brief MEL script that does what you just did by hand in the interface. Now, try copying those lines of MEL to repeat what you’ve done, as follows:
1. Select the lines.
2. Choose Edit > Copy in the Script Editor window.
3. Click in the command input area (the white area at the bottom).
4. Choose Edit > Paste in the Script Editor window.
5. Press Enter on the numeric keypad.

Maya will now create a new file, make a sphere, and move it along the Z-axis, just as you had done in the interface.

15.7 MAKING A SHELF BUTTON FOR A MEL SCRIPT

You can easily create a button that you can click to repeat a series of MEL commands. Take the following steps:

1. Select the same lines you just executed in the status message area.
2. Hold down the middle mouse button (MMB) and drag the text you've selected up to the shelf.

When the mouse gets over the shelf, it should change to an arrow with a plus (+), indicating that if you let go of the MMB it will add a button for you.

3. Let go of the MMB.

Now, you have a new button labeled MEL. If you move the mouse over it, you can see some of the text you dragged to the shelf on the Help Line at the bottom of the screen as a reminder of what it does. (Of course, this text may not be very illuminating, depending on the script!) Clicking on this button would now execute the commands you just dragged onto the shelf.

4. Since this script isn’t very useful, use the MMB to drag the new MEL button to the trash can on the far right of the screen.

Away goes your MEL button.

15.8 SAVING A MEL SCRIPT

While placing a MEL script on the shelf is often convenient, most of the time you will want to save your MEL script as a file on disk. Three reasons for this follow:

- You can share the MEL script with other people. Since shelf buttons are hidden away in your Maya configuration, it’s difficult to pass them around to fellow animators.
- Saving your MEL scripts as files avoids cluttering up your shelf with lots of MEL buttons that you seldom use.
- Later on, you will find that it is often convenient for one MEL script to run another MEL script. It’s easy for one MEL script to run another that is in a file of its own and difficult for a MEL script to run a script associated with a shelf button.
With the Script Editor open, try this:

1. Select the lines of MEL that create a new file, make a sphere, and move it.
2. From the Script Editor window’s menus, choose File > Save Selected….
3. Type test.mel as the file name, and click Save.

By default, Maya will save your script in the scripts directory under the maya directory in your home directory (My Documents when running Windows). This is a special location where Maya knows to look for script files.

To run the script, in the Script Editor type

```
source test.mel
```

and press Enter on the numeric keypad. Maya will now run your script.

**Note:** When working on large MEL scripts (as we’ll get into later), it’s often best to use an external text editor, such as Notepad or Wordpad (Windows NT), jot (Irix), or TextEdit (Macintosh) to edit and save the file outside of Maya where you can’t accidentally delete your code.

### 15.9 SEDUCTIVE DANGERS OF THE STATUS MESSAGE AREA

It all seems so easy. Work as you normally do, open up the Script Editor, fish around for the MEL commands that Maya has cobbled together for you, and either drag them to a shelf or save them to a file. For some animators, this is all there is to MEL scripting.

Unfortunately, looking in the Status Message Area to see what MEL commands are used to perform some kinds of operations isn’t always as simple as the examples we’ve presented so far. For example, try using Create > Locator from the Maya menus, and watch your Status Message Area for output. You probably noticed that nothing was displayed there at all – then what is the MEL command to create a locator? Rest assured, there is such a MEL command. What is happening is that Maya, in its infinite wisdom, has decided that these MEL commands aren’t important to display by default. You can get Maya to display more of the MEL commands it is executing by using the menu item Edit > Show All Commands on the Script Editor window. Check this on, and try performing the Create > Locator command again. This time, you’ll see the following commands appear in the Status Message Area:

```
CreateLocator;
createPrimitive nullObject;
spaceLocator -p 0 0 0;
// Result: locator1 //
editMenuUpdate MayaWindow|mainEditMenu;
autoUpdateAttrEd;
```
If you execute these commands, you’ll get not one, not two, but three locators. The reason is that numerous MEL commands print other commands that they execute internally to do their jobs. `CreateLocator`, `createPrimitive nullObject`, and `spaceLocator -p 0 0 0` are each commands that make locators, each at a lower level than the last. Also, the `editMenuUpdate` and `autoUpdateAttrEd` commands are related to updating menus in Maya’s user interface and have nothing to do with creating the locator.

Unfortunately, working this way can have troubling side effects, unless you take the time to look at and think through the MEL scripts that Maya generates for you. Even though everything in the status message area is valid MEL code intended to show you what Maya is doing behind the scenes, it is not guaranteed to have the same result as what you did in the interface. In particular, when you enabled Show All Commands, Maya sometimes prints multiple commands in the Status Message Area that do the same thing.

These repeated or extraneous steps are usually related to Maya’s user interface updating to reflect what you have done. For example, creating an object usually requires Maya to update the Edit menu to reflect what you may undo, and Maya puts the MEL to perform this update in the status message area. However, whenever you execute a MEL command to create an object, Maya will perform this Edit menu update anyway, so your own script need not include a command to do so.

The solution to this problem is to go step by step through the commands that Maya has printed in the Status Message Area and to determine what they are and what they do. Once you have done this, problems will become obvious.

If you have learned the basics of MEL, you will find it easier to puzzle your way through lists of commands that the Script Editor gives you. Trying each command by itself to see what it does can be a useful approach.

### 15.10 THE whatIs COMMAND

The `whatIs` command can be handy, particularly when you have questions about the commands Maya has printed in the status message area. For example, if you’re curious how `editMenuUpdate` is defined, try typing

```mel
whatIs editMenuUpdate
```

in the Script Editor and pressing Enter. You will get something like the following result:

```mel
// Result: Mel procedure found in: C:/AW/Maya4.0/scripts/startup/
//buildEditMenu.mel //
```

With the path to the file that contains the MEL script `editMenuUpdate`, you can hunt down what it is and how it works. Now, try

```mel
whatIs spaceLocator
```
Your result will be

// Result: Command //

This tells you that spaceLocator is a built-in MEL command and can probably be found in the MEL Command Reference part of the Maya documentation.

### 15.11 BASIC STRUCTURE OF MEL COMMANDS

Some commands include arguments, which are generally information the command needs to do its job. Usually, when a Maya command does something to an object or attribute, the object or attribute name will be an argument. For example,

```mel
select nurbsSphere1;
```

indicates that Maya should select `nurbsSphere1` as though you’d clicked on it. In this example, `select` is the MEL command, and `nurbsSphere1` is its argument – the object you want it to select.

Most commands have flags as well, which are additional instructions to the command to perform its task in a certain way. For example, one of our ways to create a locator is

```mel
spaceLocator -p 0 0 0;
```

The flag `-p` indicates that you want to provide a point in space where the locator should be created, and the numbers 0 0 0 are the X, Y, and Z locations for that point; they are the arguments for the flag. Flags do not have to have arguments, though; to get a list of all the objects that are selected, you can type

```mel
ls -sl
```

which tells the `ls` command (short for `list`) to list the objects in the selection (for which `-sl` is a shortened form). Usually, flags have a short and long form that mean the same thing; in this instance, it would also be correct to type

```mel
ls -selection
```

Many commands in MEL that are intended to make nodes have three standard flags: `-c`, `-q`, and `-e`. `-c` stands for “create” and usually tells the command to create an object. `-q` stands for “query” and usually tells the command to print out or return to your script a characteristic of the object. `-e` stands for “edit” and usually tells the command to change an existing object.
WHERE TO FIND INFORMATION ABOUT MAYA AND MEL ON THE INTERNET

There are several forums on the Internet that offer Maya users the opportunity to exchange scripts and plug-ins, discuss how to approach problems, and give each other assistance. Here are a few places to start.

As you discover what these sites have to offer, keep in mind that many such sites rely on the contributions of their readers to be most useful. If you come up with something clever or something that may benefit someone else, consider sharing the product of your effort in one of these forums.

Also, many sites like these offer community bulletin boards or discussion forums that allow people to post questions or ideas for others' response. If this is your first time participating in such forums, it can be a good idea to read the discussion for a while before contributing or asking a question. Often, such forums have a clearly defined scope of what is an appropriate topic for discussion, and reading for a while before asking a question can give you a sense of what you can expect to learn from a given forum.

15.12.1 Web Sites

http://www.melscripting.com
At melscripting.com, you can find the scripts and scene files, as well as errata and other updates.

forums.melscripting.com
The melscripting.com forums offer a place on the web to share MEL scripts and expressions, seek help from experts around the world on Maya development, and assist others with their projects.

http://www.highend3d.com/maya/
Highend3D is far and away the most comprehensive Web site for Maya users. It also includes valuable information for users of Softimage, Softimage XSI, Alias Studio, and the RenderMan and Jig renderers. Also, at Highend2D, http://www.highend2d.com, you can find similar resources for Shake and Combustion users.

In the Maya section of Highend3D are hundreds of MEL scripts, plug-ins, Maya tutorials, discussion forums, sign-up information for mailing lists, mailing list archives, and other resources. Much of the serious Internet discussion of Maya has shifted to Highend3D.

http://www.cgtalk.com
This is a broad-based web forum that specializes in helping students improve their 2D and 3D computer graphics skills. Many applications have discussion forums on the cgtalk.com Web site, and Maya's is among the most popular.
This site is Alias\Wavefront's official Web site. In addition to information on Alias\Wavefront's sales and technical support, this site includes useful articles that can help you learn Maya, discussion forums, information on user groups, and other information.

**15.12.2 Newsgroups**

**comp.graphics.apps.alias**

You can read USENET newsgroups, such as comp.graphics.apps.alias, with Netscape, Outlook Express, or any of a number of other news readers. As of this writing, there are no Maya-specific newsgroups, but comp.graphics.apps.alias mostly features Maya discussions.

**15.13 HOW TO USE MEL SCRIPTS FOUND ON THE INTERNET**

Usually, once you’ve downloaded a MEL script, you will have a file whose name ends with a `.mel` extension. To start with, look in your home directory for a directory called `maya`. (In Windows, look for a folder called *My Documents* and a directory called `maya` underneath that.) Inside the `maya` directory is a directory called `scripts`. Place your script inside this folder.

Scripts can be installed other places, and the list of those places is known as Maya's script path.

If there’s no documentation with your new MEL script to tell you how to use it, try opening the MEL file itself in a text editor. Often, programmers will place a description of how to use a script at the top of the script itself. If neither of these approaches yields anything useful, try simply typing the name of the MEL file (without `.mel`) in the Script Editor and typing Enter. Maya will often find the script and execute it on its own when you do this.

Finally, if none of the above works, you may have to type (assuming your file is called `blah.mel`)

```
source blah.mel
```

This will execute whatever MEL commands are in the `.mel` file that you have downloaded.

As you should with all software you download from the Internet, consider the trustworthiness of the source before blindly executing MEL scripts. A MEL script could alter your operating environment, change your open scenes, alter scenes on disk in undesirable ways, and more, either intentionally or unintentionally.
If you want to see what commands you’ve executed or learn whether your commands are generating errors, look in the Status Message Area of the Script Editor window.

The Script Editor and the command line provide convenient places to try commands before you put them in your MEL scripts.

Using Show All Commands will tell you the MEL commands that correspond to what you are doing in the Maya user interface by hand, but you should read and understand the generated commands before using them in your own scripts.

MEL commands are made up of a command name, flags, and arguments, all of which together specify what the command is to do. The command name (generally) specifies what the command is to do; the arguments specify to what the command should do it, and the flags and their arguments specify how it should be done.

You can install script files downloaded from the Internet in your own Maya scripts directory or in other directories that may be listed in the Maya script path.