MEL (Maya Embedded Language) is a powerful command and scripting language that gives you direct control over Maya’s features, processes, and workflow.

Maya’s user interface is built using MEL scripts and procedures. When you select menu items or otherwise use the interface, Maya performs the operations by internally running MEL commands. You can type the same MEL commands directly as a quick alternative to selecting menu items or doing other actions.

Although MEL is technically a scripting language, it contains features that are worthwhile and easy to learn even if you have no programming experience.

You can create custom buttons on the shelf so you can run the scripts you have created at the click of the mouse. In this way, you can organize MEL commands into reusable script files that let you:

- Automate tasks you might otherwise do more slowly or tediously with the user interface.
- Use additional Maya features.
- Create custom user interfaces to perform specialized workflows.
- Create new effects.

In this chapter you will be introduced to some of the basic concepts and techniques for working with MEL.
Preparing for the lesson

To ensure the lessons work as described, do these steps before beginning:

1. Create a new scene.
2. Use a computer that has an Internet connection and Web browser. Part of the lesson requires that you download material from our Web site at www.alias.com.
Lesson 1  MEL basics

In this lesson, you learn how to:

• Use the Script Editor to view and enter MEL commands.
• Enter MEL commands in place of selecting menu items.
• Create a Shelf button to run a series of commands.
• Run and create scripts that execute a series of commands.

Using MEL commands

In the first part of the lesson, you’ll learn how to type a MEL command as a substitute for selecting a menu item.

To use the script editor

1 Select Window > General Editors > Script Editor.

```mel
file -f -new;
// Result: /untitled //
sphere -p 0 0 0 -ax 0 1 0 -sw 0 -esw 360 -r 1 -c 3 -ut 0 -to 0.01 -a 8 -ep 4 -ch 1 -object;
select -cl ;
select + r -nurbsSphere1 ;
scale + r 5.808255 5.808255 5.808255 ;
move + r 6.459346 0 ;
displaySmoothness -divisionsU 3 -divisionsV 3 -pointsWire 16 -pointsShaded 4;
subDivDisplaySmoothness -smoothness 3;
```
The top part of the Script Editor shows the command history—a running list of commands that you or Maya have executed. The results of the commands are listed between pairs of “/\” characters. The meaning of the results will be obscure until you become familiar with MEL scripting.

The bottom (white) part of the window is the input section where you enter commands or scripts. If you make a typing mistake while entering commands, an error message appears in the history section.

2 In the input section, type the following command exactly as shown. Entries are case sensitive. The semicolon (;) signifies the end of the command.

```
sphere;
```

3 To execute the command, press Enter on the numeric part of your keyboard. The Enter (Windows, IRIX, and Linux) or Return (Mac OS X) key you normally use does not execute a command. It starts a new line so you can type several commands before executing them.

After you execute the command, Maya creates a NURBS sphere at the origin.

4 Enter the following command to delete the sphere.

```
delete;
```

This deletes the selected object, which, by default, is the last object you created.

Alternatively, you can enter individual MEL commands in the Command Line near the lower-left corner of the Maya window. This saves you the time it takes to display the Script Editor. In addition, the Command Line keeps a history of commands which can be readily accessed using the up and down arrow keys when working in it.

5 Create a bigger sphere:

```
sphere -radius 5 -name Planet;
```
The sphere, named Planet, has a radius of 5 grid units.

6 Move Planet to a new position:

move -relative 3 0 0;

This moves the selected object to a position 3 units in an X-axis direction (and 0 in Y and Z), relative to its current position. The option \textit{-relative} is known as a flag (or argument).

7 Select Help > MEL Command Reference to see a list of commands sorted by name. Find the move command and click its link to see the command description.

Maya includes command reference material that describes how to use each command and its flags. Alternatively, you can get quick help on a particular command directly in the Script Editor by typing the word \textit{help} followed by the name of the command you wish to obtain more information about. For example, typing \textit{help move} in the Script Editor will provide information about the various flags available to modify the move MEL command.

In the Flags section, you will see \textit{-r/-relative} and a description of the flag. The slash between \textit{-r} and \textit{-relative} means that you have a choice of typing \textit{-r} or \textit{-relative}. So you could have typed the following command instead of the prior one:

move -r 3 0 0;

8 Delete Planet.

Creating a Shelf button for a MEL command

In the next steps, you’ll create a Shelf button that executes three commands. The commands create a pair of lights that you can use for a default simple lighting arrangement in your scenes.
To create a Shelf button for MEL commands

1. In the Script Editor, select Edit > Clear History. This removes all text from the history section of the Script Editor to lessen clutter.

2. Select Create > Lights > Directional Light. A directional light is similar to the sun. Its parallel rays strike all objects in the scene from a single direction.

3. Use the Rotate Tool to rotate the light’s Rotate X value to -35 or so. To ensure that the Shelf button works correctly, use the Rotate Tool, not the Channel Box or Attribute Editor. This rotational change gives the light a downward sloping direction.

   Do not use the Channel Box or Attribute Editor when changing attributes for the purpose of creating custom Shelf buttons as these editors set attributes for uniquely named nodes vs. executing the commands for whatever is currently selected.

4. In the history section of the Script Editor, highlight the following command text:
   
   ```mel
   defaultDirectionalLight(1, 1, 1, 1, "0", 0, 0, 0);
   rotate -r -os -35.048621 0 0 ;
   ```
Lesson 1 > Creating a Shelf button for a MEL command

5 Left-mouse drag (Mac OS X) or Middle-mouse drag (Windows, IRIX, and Linux) the text to the input section of the Script Editor.

6 Select Create > Lights > Ambient Light. An ambient light emits light uniformly in all directions. It’s useful for simulating low-intensity indirect lighting, like lighting reflected off the walls of a room.

7 In the history section of the Script Editor, drag through the following text to highlight it:

```
defaultAmbientLight(1, 0.45, 1,1,1, "0", 0,0,0, "1");
```

8 Left-mouse drag (Mac OS X) or Middle-mouse drag (Windows, IRIX, and Linux) the text to the line below the existing two lines in the input section. It will appear as follows:

```
defaultDirectionallight(1, 1,1,1, "0", 0,0,0);
rotate -r -os -35 0 0 ;
defaultAmbientLight(1, 0.45, 1,1,1, "0", 0,0,0, "1");
```

If the dragged text is positioned to the right of the second line rather than on its own line, position the cursor to the left of `defaultAmbientLight` and press Enter on the alphanumeric part of the keyboard.

With the desired statements in the input section, you could execute them by pressing Enter (numeric pad). Instead, you’ll create a button on the shelf so you can execute the commands now or later, in any scene you create.

9 Select the General Tab on the Shelf so it is displayed. The shelf is the row of icons immediately above the scene view.

10 Highlight the three lines of text in the input section and select File > Save Selected to Shelf in the Script Editor.

A window appears prompting to enter the name of the new shelf item.
11 **Type** `LightB` **and press Enter.**

The button named `LightB` with a MEL icon appears on the shelf.

12 **Click the MEL button.** The command text executes and creates the lights.

If you later decide to remove the button from the shelf, you can middle-mouse drag it to the Trash icon on the shelf.

If you forget the purpose of a button, move the pointer over it. A pop-up box lists the first command or two that the button executes. For easier recognition, you can optionally customize the display of a button by adding an overlay label to it.

You can edit the contents of Shelf items by selecting **Window > Settings/Preferences > Shelves.** For further information and related techniques, refer to the Maya Help.

Now you’ll use the MEL button to create the lights and observe their effect on a sphere.
To use the MEL button

1. Delete the lights from the scene or create a new scene.
2. Create a polygonal sphere.
3. In the perspective view, press 5 on your keyboard (for Shading > Smooth Shade All).
4. Click the MEL button to create the directional and ambient lights.
   Maya creates the lights at the origin. The icons representing them are hidden by the surface of the sphere. By default, Maya lights the scene with a single invisible directional light that points in the same direction as the scene view.
5. From the perspective view menu, select Lighting > Use All Lights to light the scene with the lights you just created.
6. To return to the default lighting for comparison, select Lighting > Use Default Lighting.

Tip
To create a shelf button for a single menu item, for instance, Display > Grid, hold down Shift-Ctrl-Alt (Windows, IRIX, and Linux) or Shift-Control-Option (Mac OS X) while selecting the menu item.
**Downloading Alias MEL scripts**

A MEL script is a file containing a collection of MEL procedures and commands. As mentioned in the lesson’s introduction, you can extend Maya’s capabilities with scripts. In the next steps, you obtain a free script available on our customer community Web site.

1. Create a new scene before starting the next part of the lesson.
2. Select Help > Alias on the Web to point your Web browser to the Alias web site and choose the downloads link. Alternatively you can launch your Web browser. Go to www.alias.com/maya/downloads.

The Maya downloads page provides a search engine to access the many shaders, plug-ins and MEL scripts available for download. You will need to register in order to obtain access. (Be sure to read the Terms and Conditions of Use document available from this page.)

3. Use the Web page’s search engine to search for Maya MEL scripts, and choose spiral.mel to obtain a script that creates a spiral curve based on values you supply for height, radius, and number of revolutions (“rounds”).
4. In the file browser that appears, select the appropriate destination directory for the scripts file to download to. For example:
Lesson 1 > Downloading Alias MEL scripts

(Windows) drive: \Documents and Settings\username\My Documents\maya\scripts

(Mac OS X) Users/username/Library/Preferences/Alias/maya/6.0/scripts

(IRIX and Linux) ~username/maya/scripts

Substitute your specific hard drive letter and login name where username and drive appear in the above paths.

Many of the shaders and scripts in the downloads area of the Web site are compressed to reduce download time. Ensure that the MEL script is uncompressed (if necessary) before proceeding. To see how to execute the script, you need to examine the contents of the spiral.mel script for comments and clues.

**To examine the MEL script’s contents**

1. In the Script Editor, select File > Open Script. In the resulting browser, open spiral.mel. The script appears in the Input area of the Script Editor.

2. Read the comments in the file—the text following the //</br>characters. The comments describe the purpose of the script in addition to copyright and disclaimer information.

   Notice the following line in the script:

   ```mel```
   ```
   global proc string spiral( float $ht, float $radius, float $numRounds)
   ```
   ```
   The three variables within the parentheses, $ht, $radius, and $numRounds are arguments to the procedure spiral. They represent values that you must enter when you execute the spiral.mel script in a later step. The variables specify the height, radius, and number of rounds for the spiral curve that will be created when you execute the script.

   The entry float is short for floating point—the data type of an argument. A floating point number is a number with a decimal point rather than an integer.
To execute the MEL script

1. To execute the script, enter the following text in the input section of the Script Editor. Remember to press the numeric Enter key after the text.
   
   spiral 5 2 10;

2. Maya finds the spiral.mel script in the scripts directory and executes the script contents to create the following spiral curve of height 5, radius 2, and 10 rounds:

   The script file must be in the appropriate scripts directory or your script will not execute when you enter its name in the input section of the Script Editor.

Creating a user interface slider using a MEL script

In the next steps, you’ll create a MEL script, save it to disk, and execute it. The script creates a user interface slider for controlling an attribute value. The advantage of creating a script file rather than a MEL button is that a script is portable. You can copy a script to another user’s scripts directory easily.

1. In the Script Editor, select Edit > Clear All to erase all text from the history and input sections of the Script Editor.
In the input section of the Script Editor, type the following text. In the text after the dotted line, use the exact lowercase and uppercase letters shown:

// Creation Date: <Date>
// Author: <Your Name Here>
//
// Description: This script creates a custom slider that controls the Translate Y attribute of an object named pSphere1.
// -----------------

global proc MySliderMaker()
{
    window -title "Slider for pSphere1.ty"
    SliderTest;
    columnLayout;
    attrFieldSliderGrp
    -min 0
    -max 10
    -at "pSphere1.ty";
    showWindow SliderTest;
}

Statements that follow a pair of backslash characters (//) are comments for other people who might read the script later to extend its capability. Maya ignores these statements. The remaining text is a procedure—a collection of MEL commands and operations contained in braces and given a name.

The procedure’s name is MySliderMaker. In technical terms, the procedure is a global procedure, which means it can be executed from any other script in Maya.

Drag through the entire text to highlight it.
From the Script Editor menus, select File > Save Selected. In the file browser that appears, enter the name MySliderMaker.mel. By default, Maya automatically saves the text as a script file in the appropriate script directory.

Also, the base name of the script file must be the same as the name of a global procedure in the script. In this example, the script file has only one global procedure, MySliderMaker, so you must name the script file MySliderMaker.mel. (Script files always end with the extension .mel.)

If the system reports an error with your procedure it may mean you’ve made an error when typing. You will need to either correct the error or delete the global procedures name and start again. If you are new at writing MEL scripts, it’s best to position the global procedure (the one with the same name as the script file) as the last procedure in the script. Your script will be more likely to run without error.

The first line of the MySliderMaker.mel script is a procedure declaration. It informs the software that the lines between curly braces ({ }) are a related series of operations.

In general, a global procedure has this form:

```
global proc procedureName ( )
{
 commands
}
```

Maya executes the commands within the curly braces every time you enter the corresponding MEL script filename in the Script Editor. Be aware that using global procedures might cause errors if another script contains an identical procedure name. There are techniques involving the use of local procedures for avoiding such errors. See the Maya Help for details.

Select Create > Polygon Primitives > Sphere. By default, this creates a sphere named pSphere1. Press 5 to display the sphere with smooth shading.
6 Type: MySliderMaker in the input section of the Script Editor and press Enter on the numeric keypad. Do not type the .mel extension. Unlike the script you copied from www.alias.com earlier in the lesson, you do not type any arguments (for instance, -5) after the script name. Some scripts require you to type arguments. Others do not.

The MySliderMaker.mel script executes. It displays a custom slider that controls the translateY attribute of the object named pSphere1. (Drag the right edge of the slider window, if necessary, to see the entire slider.) Drag the slider back and forth to move the sphere up and down.

Next, you’ll learn about the four commands that make up the MySliderMaker procedure on a line-by-line basis. The commands are listed below:

```mel
window -title "Slider for pSphere1.ty" SliderTest;
columnLayout;
attrFieldSliderGrp -min 0 -max 10 -at "pSphere1.ty";
showWindow SliderTest;
```

The first command creates a window with a title bar (the -title flag) that contains the text string Slider for pSphere1.ty. The internal name of the window is SliderTest. The window is displayed only when the fourth command executes.

The second command creates the window layout in a column format.

The third command creates a slider group that consists of the label, text entry field, and slider. The slider value can range from a minimum of 0 to a maximum of 10. The entry -at pSphere1.ty specifies that the slider controls the translateY attribute (ty) of pSphere1.
Beyond the lesson

This lesson provided an introduction to using MEL for creating custom features in Maya. In this lesson you learned how to:

- Use the script editor for MEL commands.
- Use MEL to create customized interface components.

If you have never used a scripting or programming language, a few concepts in this lesson might seem foreign. With a little study and experimentation with MEL, you’ll quickly find ways to work more productively with Maya.

To become a successful MEL user, you need to learn the basics of the language. For details on the language and example scripts, see the Maya Help. The Maya Help MEL command reference includes examples of command usage at the bottom of most command descriptions.

If you have previously used a scripting or programming language before, you’ll have little difficulty working with MEL. If you’ve worked with the C language, you’ll have a head start; MEL uses many of the language elements of C, including data types and flow control methods.

Behind the lesson

Early in the lesson, you moved the following three lines of text from the Script Editor to a MEL button:

```
defaultDirectionalLight(1, 1,1,1, "0", 0,0,0);
rotate -r -os -35 0 0 ;
defaultAmbientLight(1, 0.45, 1,1,1, "0", 0,0,0, "1");
```

For brevity, we referred to the three lines as MEL commands. Only the second is a command; the other two lines are procedures.

A MEL procedure is a collection of MEL commands that are accessed by executing only the name of that procedure (and any associated arguments) instead of executing the individual MEL
commands one at a time. Procedures differ from MEL commands in that they require specific arguments in order to execute. These arguments are not optional such as they are with a MEL command flag option.

Although the procedures often have similar results to MEL commands, the syntax is different and you won’t find documentation of how to use them in the Maya Help MEL command reference.

If you want to know whether a statement you see in the Script Editor is a command or procedure, enter the command `whatIs` in the Script Editor as in the following example:

```
whatIs defaultDirectionalLight;
```

Depending on your computer platform and version of Maya you’ll see a response in the Script Editor similar to this:

```
// Result: Mel procedure found in: drive:/aw/Maya6.0/
scripts/others/defaultDirectionalLight.mel
```

This means that `defaultDirectionalLight` is a procedure in a script named `defaultDirectionalLight.mel`. You can tell that the script is part of the Maya software based on its location. The custom scripts you create instead must reside at `Users/username/Library/Preferences/Alias/maya/6.0/scripts`.

| Note | Be cautious if you decide to modify the supplied Maya software scripts in case you accidentally change one by mistake. You may wish to first duplicate and rename it before modification. Create and modify only your own scripts. |

If you want to create lights with a MEL button, you don’t need to use internal procedures such as `defaultDirectionalLight`. You can instead use MEL commands such as `directionalLight`. As usual with Maya, there are several ways to get the results you want.
You are not required to use the command history in the Script Editor when you create a MEL button. It’s there for convenience.

| Note | There is no guarantee that the history text you copy to a MEL button will have the same result as the user interface actions from which the text was generated. There are subtle issues in MEL command sequences that make them work in one workflow, but not in another. You need to check the results and make modifications as necessary. |

By default, not all Maya operations appear in the command history as you use the interface. Maya filters out many of the actions you usually don’t need to see. To see all operations, select Script > Echo All Commands (from the Script Editor window). This might be necessary if you are having difficulty getting the desired result from a text you copy to a MEL button.