

The background of the slide features a dynamic, abstract design with swirling, liquid-like patterns in shades of deep red and black. The patterns create a sense of movement and depth, with highlights and shadows that give it a three-dimensional appearance.

Maya Commands

***Kristine Middlemiss**, Senior Developer Consultant
Autodesk Developer Network (ADN)*

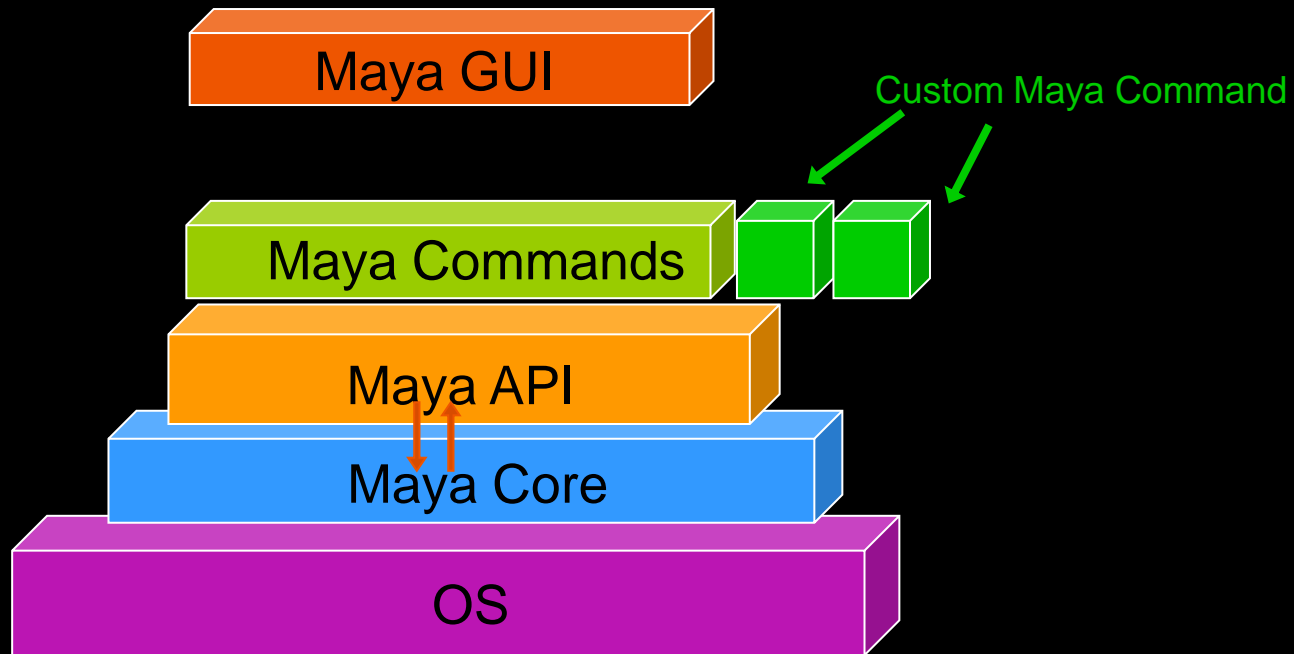
Autodesk®

Agenda

- Custom Commands



Extending the Script Functionality



Maya Command Architecture

- Maya Command Architecture is language neutral
 - Favours shell-style languages
- No Python-MEL translation

Custom Commands



MPxCommand

- All commands are derived from MPxCommand
- The only two necessary functions that need to be implemented for commands are:
 1. creator()
 2. doIt()

1. creator() Function

- This is what is creates an instance of the command.
- It's called without needing an instance of the class.
- Not necessary to call creator, however through uniformity we do so.

1. creator() Implementation

- The creator method is called to return a new instance of the command:

```
def cmdCreator():  
    return OpenMayaMPx.asMPxPtr(myFirstCmd() )
```


2. dolt() Function

- Called when command is executed.
- This does all the work for the command.
- Parse the “args” arguments and perform user-defined operation

2. dolt() Implementation

- Called when command is executed:

```
class myFirstCmd(OpenMayaMPx.MPxCommand):  
    def __init__(self):  
        OpenMayaMPx.MPxCommand.__init__(self)  
    def dolt(self,argList):  
        print "Hello World!"  
        # The body of the command code
```

MPxCommand Registration

- To register your custom command in Maya:

```
kPluginCmdName = "myFirstCmd"
def initializePlugin(mobject):
    mplugin = OpenMayaMPx.MFnPlugin(mobject)
    try:
        mplugin.registerCommand(kPluginCmdName, cmdCreator)
    except:
        sys.stderr.write( "Failed to register command: %s\n" % kPluginCmdName )
        raise
```

- To deregister your custom command:

```
def uninitializePlugin(mobject):
    mplugin = OpenMayaMPx.MFnPlugin(mobject)
    try:
        mplugin.deregisterCommand( kPluginCmdName )
    except:
        sys.stderr.write( "Failed to unregister command: %s\n" % kPluginCmdName )
        raise
```

Autodesk