

PowerPoint Commands

PowerPoint is an application that is supported with a variety of Natural Language commands. While these are helpful, power users of PowerPoint may benefit from creating custom commands for common functions not supported in those Natural Language commands.

It can be impressive (or simply necessary) to present a PowerPoint slide show hands-free, using verbal commands to open a presentation and then move slide-to-slide. Opening a presentation is easily done in an Advanced Scripting command, and moving slide-to-slide can be done using DVC commands that are only available in Sleep Mode. This is one of the rare cases where DVC commands can accomplish a task that (to my knowledge) can not be done with Advanced Scripting commands.

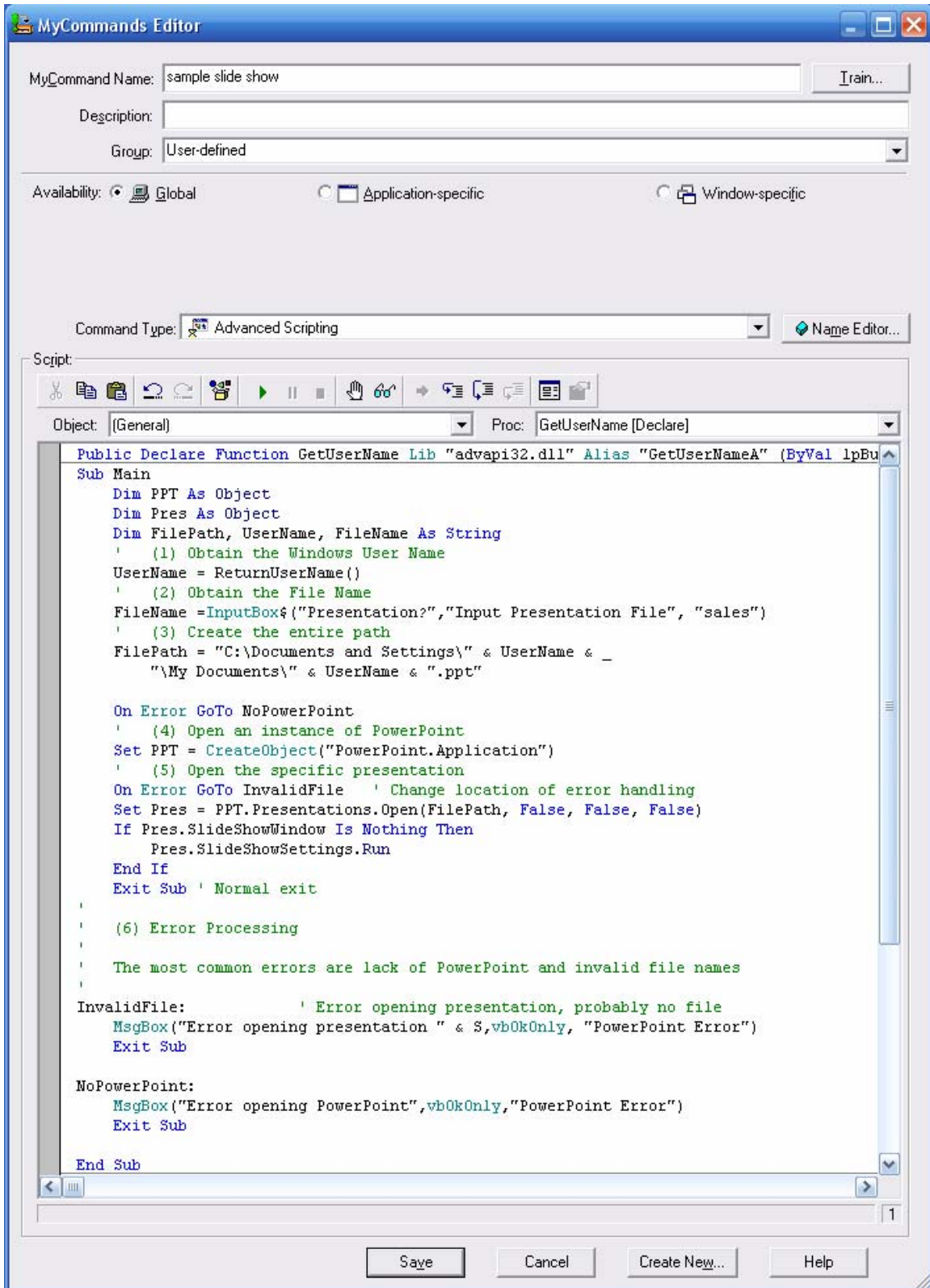
Opening a Presentation

Opening a presentation is shown as the global command “sample slide show”.

This command presumes that the presentation is directly in the My Documents folder. It can be modified to use other folders.

Step 1: A function ReturnUserName is used to obtain the Windows user name, which normally allows finding the entire directory string for a folder. The function is not shown below, but the code (copied from one of many web sites having code to provide the user name) is in the command that can be imported. Note that the first line of the “sample slide show” command is required for the ReturnUserName function.

Step 2: This command illustrates using an InputBox command to obtain a file name for the presentation. By voice, this works better if the file names correspond to words.



Step 3: The pieces of the complete path name are concatenated to produce the entire directory path "FilePath." If you wish, add a statement:

MsgBox FilePath

to see the entire path. Note that this example presumes a My Documents folder on a "C:" drive, which may not always be the case.

Step 4: Initiate PowerPoint using the PowerPoint.Application object. You will not see anything on the screen at this point, but using Task Manager after this is executed you will see the powerpnt.exe process is running.

Step 5: Open the specific presentation.

If you are going to use this command along with the commands shown below for operating a slide show using commands in sleep mode, then add a line:

GoToSleep

after the "End If" and before the "Exit Sub" to put DNS into sleep mode once the slide show is started.

Step 6: There are a pair of error handlers, one (NoPowerPoint) that would most commonly be executed when PowerPoint can not be started, likely because PowerPoint does not exist on the system; another (InvalidFile) that is most likely triggered by providing the wrong file name. Using error handlers allows isolating problems to allow easier resolution.

Step 7: (not shown above) The ReturnUserName is a segment of copied code obtaining the Windows user name.

Operating a Slide Show using Sleep Mode Commands

When presenting, one way to control slides by voice is to keep DNS in Sleep Mode most of the time, but have a few commands that allow moving from one slide to another while in sleep mode.

Following is the normal system.dvc file segment containing commands that are recognized in sleep mode, namely "wake up" and "listen to me"

```
MENU "Global Commands" {
```

```

STATE "Asleep" SLEEPING GLOBAL {
  COMMAND "<WakeUp>" {
    SCRIPT "WakeUp"
  }
  LIST "WakeUp" {
    "wake_up"
    "listen_to_me"
  }
}
}

```

Additional commands may be added into the “Asleep” state. Below is a modified file segment containing the commands “next slide” and “previous slide” as additional commands:

```

MENU "Global Commands" {
  STATE "Asleep" SLEEPING GLOBAL {
    COMMAND "<WakeUp>" {
      SCRIPT "WakeUp"
    }
    COMMAND "next slide" {
      SCRIPT {
        SendKeys "{PgDn}"
      }
    }
    COMMAND "previous slide" {
      SCRIPT {
        SendKeys "{PgUp}"
      }
    }
    LIST "WakeUp" {
      "wake_up"
      "listen_to_me"
    }
  }
}
}

```

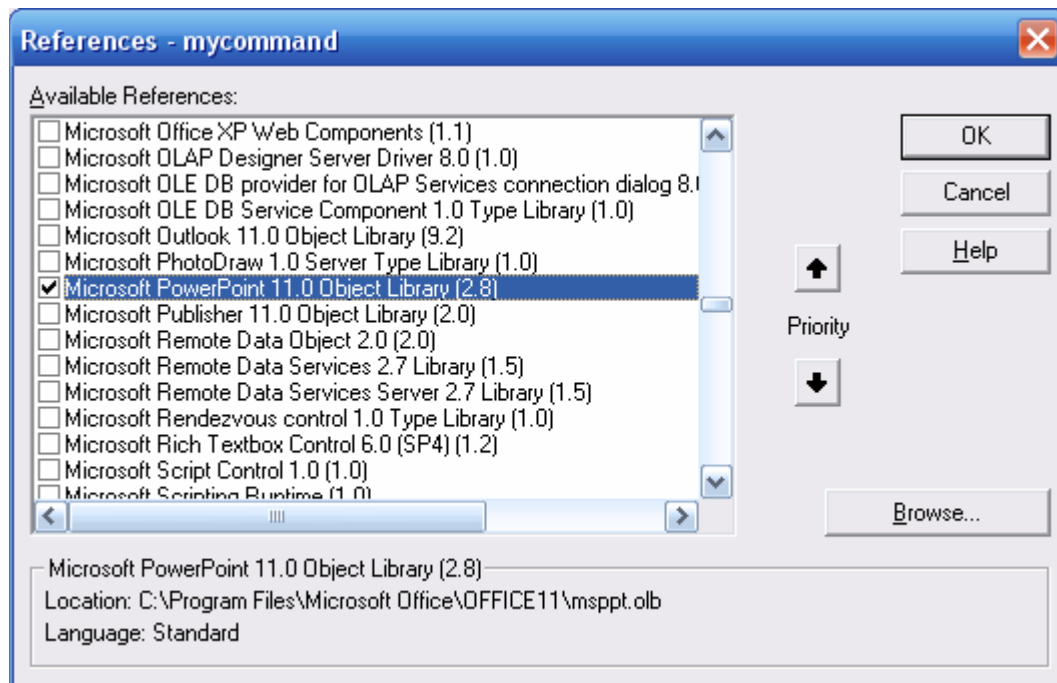
The “next slide” is a Page Down keystroke, “previous slide” is a Page Up keystroke. This same technique can be used to add other commands in the “Asleep” state. Note that pauses are needed, the above commands “next slide” and “previous slide” have been used for live presentations.

Commands for Building Presentations

Acknowledgements: Tom Perry has graciously contributed the basic macros included in this section. Modifications have been made, the author takes responsibility for any mistakes.

There are many functions of Powerpoint easily available using keystroke shortcuts. Techniques for creating commands for those functions are discussed elsewhere. But some functions are better done using the PowerPoint object model, supplied by Microsoft, more directly. Microsoft has reference material for the PowerPoint object model in the PowerPoint Help system.

To reference the object model (including referencing PowerPoint macros which you or someone else may have already created) one must start an Advanced Scripting macro with a reference to the PowerPoint object library. Put the cursor between the Sub Main and End Sub statements, right-click, and select “References.” Wait a few seconds while they load, then find the PowerPoint object library corresponding to the version of PowerPoint you are using. 11.0 corresponds to PowerPoint 2003 in the screenshot below.



Click OK, then begin your command.

The command will typically start with a declaration of an object “oApp” (you can pick your own name) and then using a “GetObject” statement to “connect” this macro to the running version of PowerPoint.:

```
Dim oApp As Object
Set oApp = GetObject(, "PowerPoint.Application")
```

The command should end by setting the oApp object to the value “Nothing” as in:

```
Set oApp = Nothing
```

So the command is basically:

```
Sub Main

Dim oApp As Object
Set oApp = GetObject(, "PowerPoint.Application")

` Do the work of the command

Set oApp = Nothing

End Sub
```

A complete example for changing the font size for selected text is:

```
Command Name:    font name <pcspeak_0_100>
```

where <pcspeak_0_100> is a list of numbers 0 to 100.

```
Sub Main
    Dim a As String
    Dim oApp As Object
    Set oApp = GetObject(, "PowerPoint.Application")
    a = ListVar1
    With oApp.ActiveWindow.Selection.TextRange.Font
        .Size = a
    End With
    Set oApp = Nothing
End Sub
```

This command, when executed, will change selected font to the stated size.

A command to run an existing macro “fixup” contained in template “corpstandard” would be:

```
Sub Main
    Dim oApp As Object
    Set oApp = GetObject("PowerPoint.Application")
    oApp.Run "corpstandard.ppt!fixup"
    Set oApp = Nothing
End Sub
```