Bethesda Tutorial Papyrus Hello World

creationkit.com/index.php

Contents

Overview

This tutorial introduces the basics of Papyrus, the Creation Kit's scripting language.

You will learn:

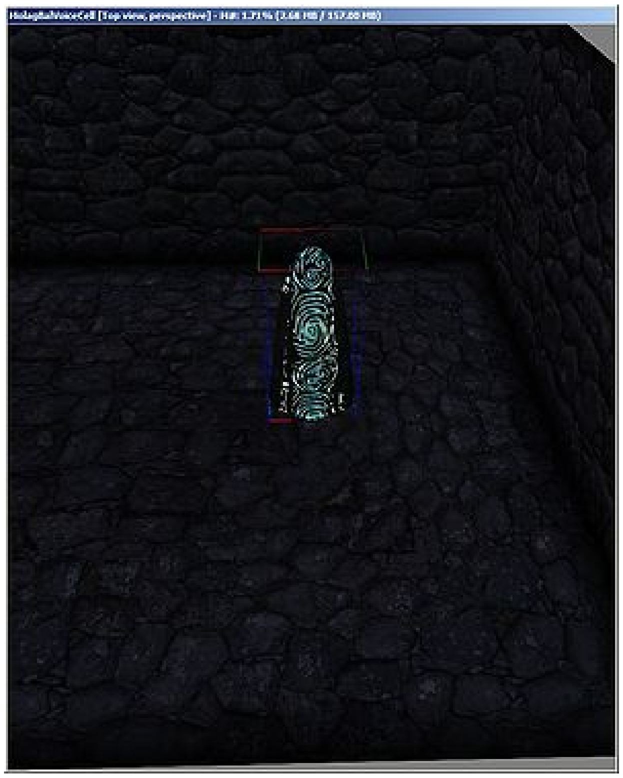
- How to create a new script and attach it to an object.
- How to get your script to respond to an event.

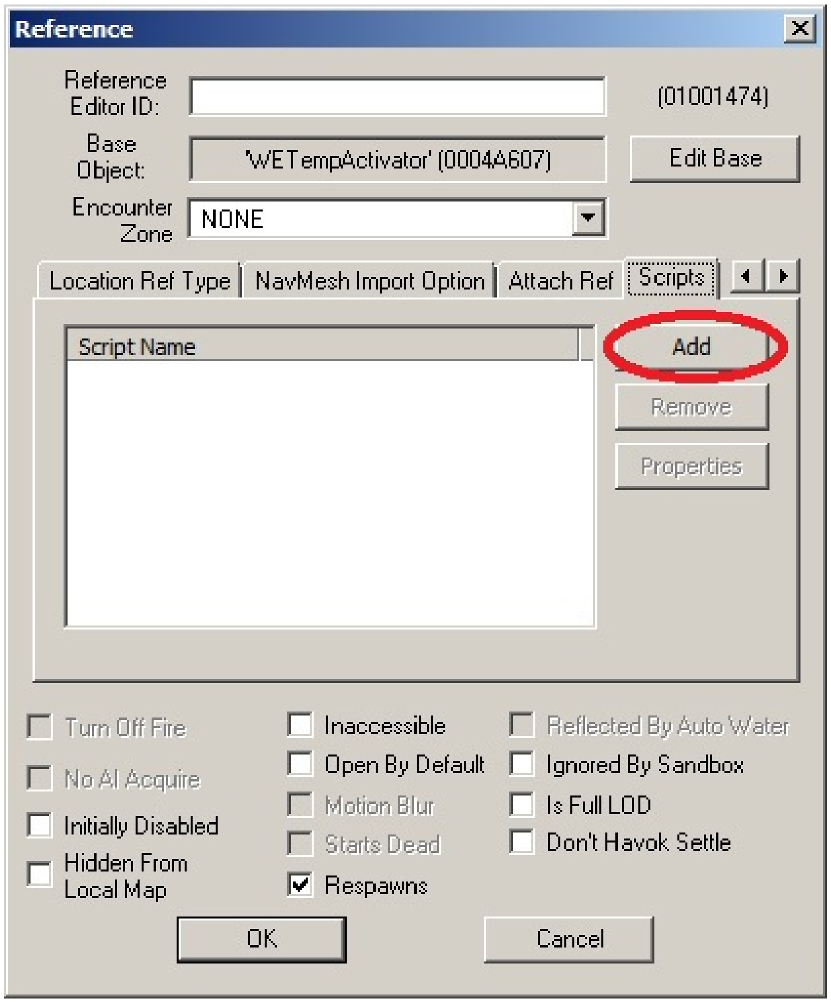
Creating a Script

The first thing we're going to do is to create a new script and attach it to an object in the world. Run the Creation Kit and load up a test cell of your choice. I'll be using MolagBalVoiceCell simply because it's a nice empty cell without any clutter to distract us.

Let's put an object in the cell - use **WETempActivator** from the Activator list. It's a simple glowing pillar that we used as a temporary object during development.

Double-click on the object to open the Reference window. Switch to the Scripts tab. This is where you can add scripts to any object in the game.


MolagBalVoiceCell [Top view, perspective] - Hit: 1.71% (2.68 MB / 157.00 MB)



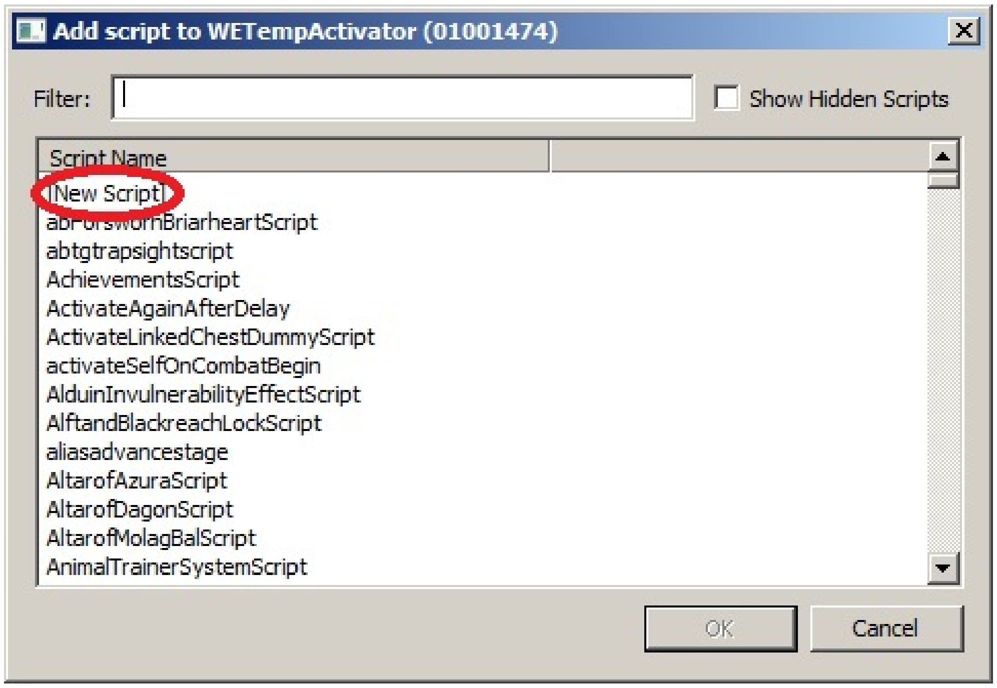
Reference Editor ID: (01001474)

Base Object: 'WETempActivator' (0004A607) Edit Base

Encounter Zone: NONE

☐ Turn Off Fire ☐ Inaccessible ☐ Reflected By Auto Water
☐ No AI Acquire ☐ Open By Default ☐ Ignored By Sandbox
☐ Initially Disabled ☐ Motion Blur ☐ Is Full LOD
☐ Hidden From Local Map ☐ Starts Dead ☐ Don't Havok Settle
☑ Respawns

Click the Add button to bring up the "Add script" window.

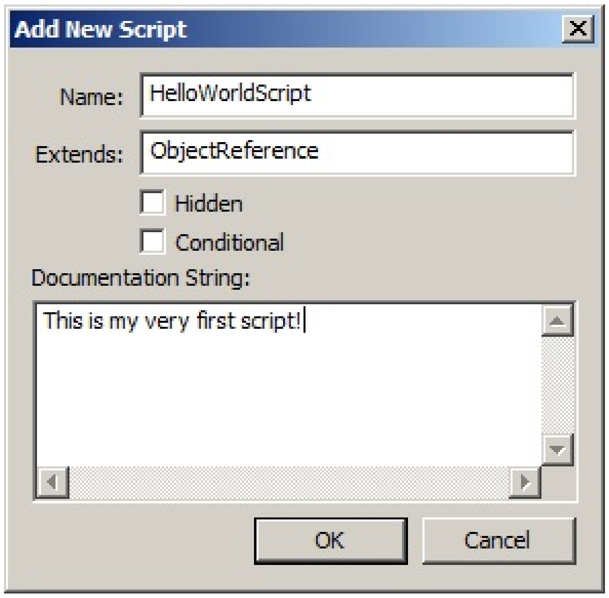


[New Script]
abForswornBriarheartScript
abtgtrapsightscript
AchievementsScript
ActivateAgainAfterDelay
ActivateLinkedChestDummyScript
activateSelfOnCombatBegin
AlduinInvulnerabilityEffectScript
AlftandBlackreachLockScript
aliasadvancestage
AltarofAzuraScript
AltarofDagonScript
AltarofMolagBalScript
AnimalTrainerSystemScript

Double-click "[New Script]" at the top of the list to create your new script. Change the Name field to "HelloWorldScript" (this will be the script's name), and hit "OK".

- If you get this error: "The extends script does not exist, please pick one that does" This seems to be due to the new update it has removed the loose papyrus source files and placed them in a .rar file ("scripts.rar") in the base Data folder of the Skyrim installation. You'll have to extract the rar back into their original folder again for CK to work correctly.

Please Note: the field labelled "Documentation String". Your script does not go here! The 'Documentation String' is a tool-tip primarily used to provide some quick information on the nature of the script. What it does, what objects it may apply to etc. You'll now see that your new script has been added to the pillar's script list:



Add New Script

Name: HelloWorldScript

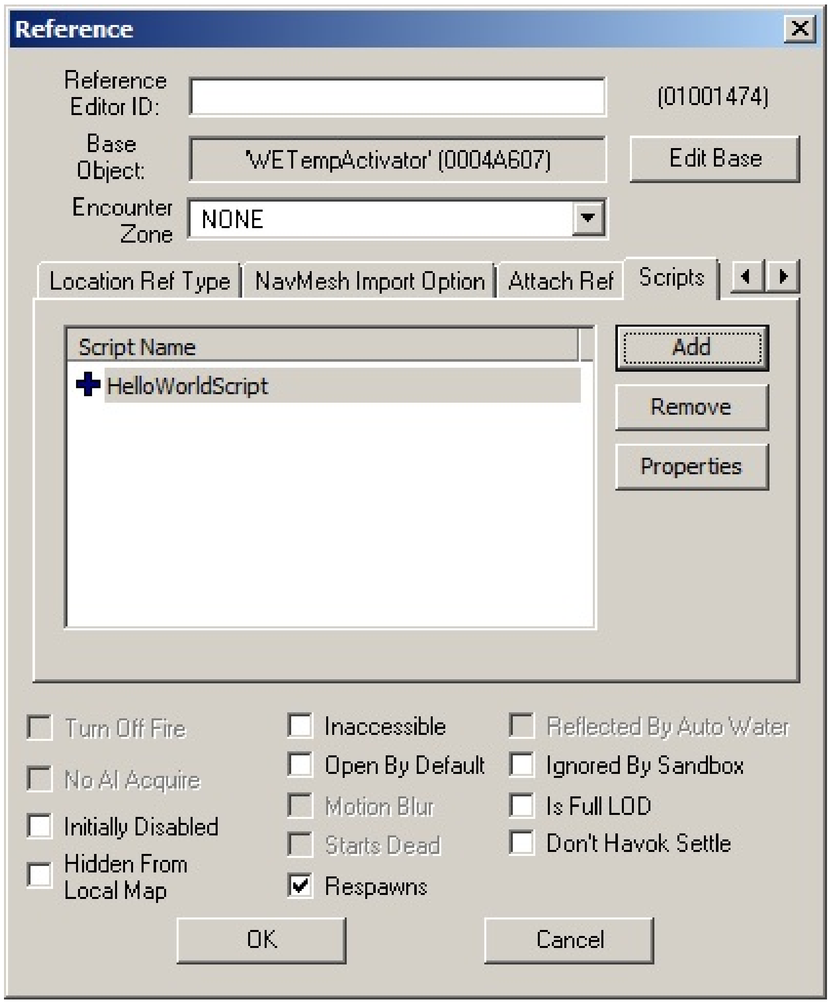
Extends: ObjectReference

☐ Hidden

☐ Conditional

Documentation String:

This is my very first script!



Reference

Reference Editor ID: (01001474)

Base Object: 'WETempActivator' (0004A607)

Encounter Zone: NONE

Script Name

HelloWorldScript

☐ Turn Off Fire
☐ No AI Acquire
☐ Initially Disabled
☐ Hidden From Local Map
☐ Inaccessible
☐ Open By Default
☐ Motion Blur
☐ Starts Dead
☑ Respawns
☐ Reflected By Auto Water
☐ Ignored By Sandbox
☐ Is Full LOD
☐ Don't Havok Settle

Hit "OK" to save the changes to the pillar reference. Congratulations! You've just created your first script and attached it to something in the world.

Adding an Event

Of course, your script doesn't do anything yet - it's just an empty shell waiting to be given something to do.

Because this object is an Activator, it can respond to being clicked on ("activated") by the player. So let's tell our script to show us a message when the player clicks on the pillar.

Reopen the pillar's Reference window, and right-click on the HelloWorldScript on the Script tab. Select "Edit Source", which will bring up the script editing window.

Now we need to tell the script to respond to being activated, which means we need to add the OnActivate event to our script. Add the following lines to your script:

```
Event OnActivate(ObjectReference akActionRef)

endEvent
```

For now, don't worry about how we knew what syntax to use here when we defined the OnActivate event - we'll go into more detail on that in a later tutorial. For now, let's just see if we can get our script to do something in the game.

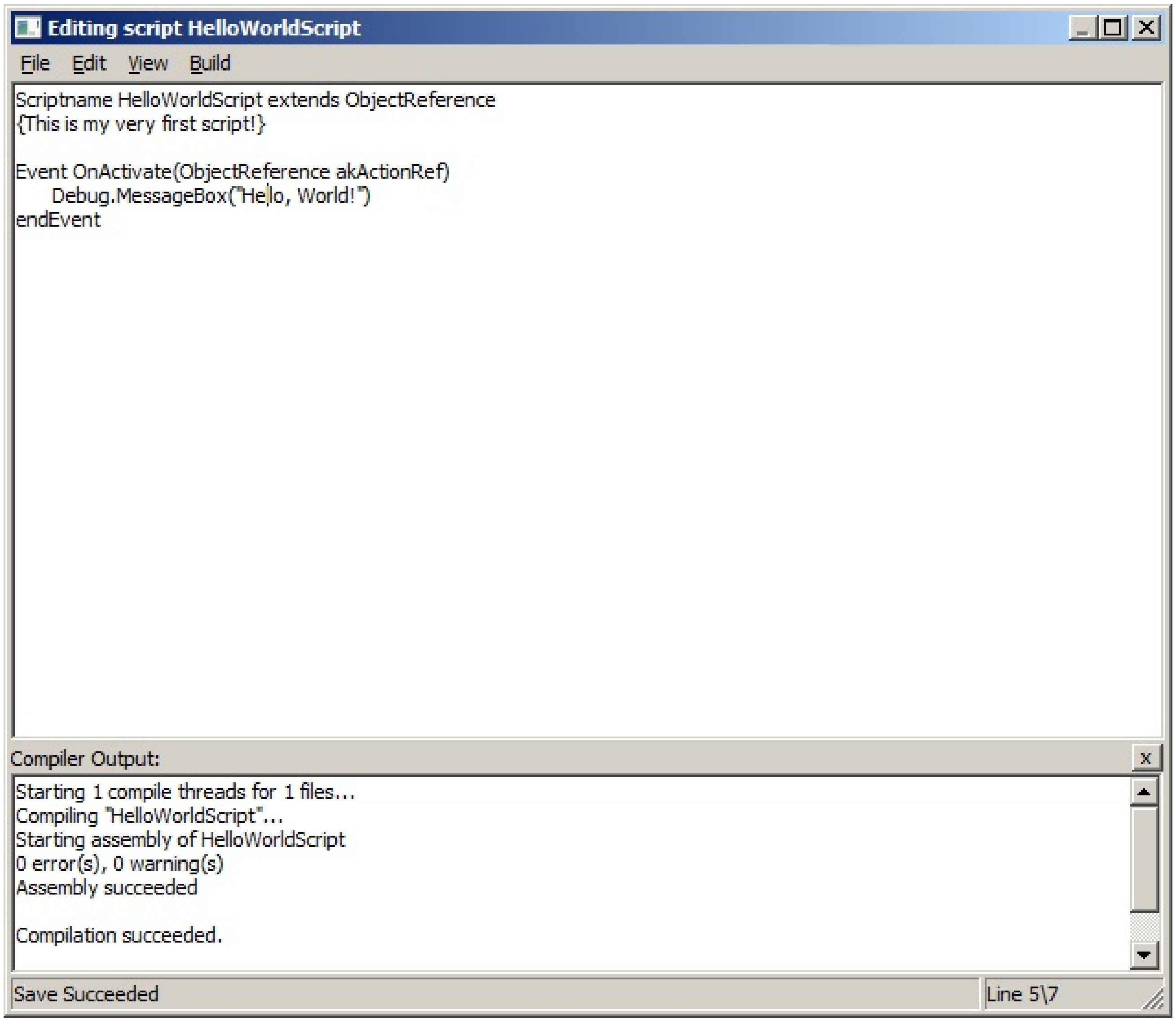
At this point, our script is ready to respond to the OnActivate event, so let's tell it to show a simple message box:

```
Event OnActivate(ObjectReference akActionRef)
   Debug.MessageBox("Hello, World!")
endEvent
```

To explain this last step a bit more, the syntax of the line breaks down as follows:

> **Debug.**MessageBox("Hello, World!"): This is telling the script that we're calling a function on a different script (in this case the special default Debug script object).
> Debug.**MessageBox**("Hello, World!"): MessageBox is a function that pops up a message box.
> Debug.MessageBox**("Hello, World!")**: The parentheses show that we're calling a function; whatever's inside the parentheses is the data that we're passing to the function (in this case, the text that we want to be displayed in the message box).

Save and compile your script by selecting "Save" from the File menu on the script editing window (or CTRL-S). If you typed everything correctly, you should see this:



```
Scriptname HelloWorldScript extends ObjectReference
{This is my very first script!}

Event OnActivate(ObjectReference akActionRef)
    Debug.MessageBox("Hello, World!")
endEvent
```

Compiler Output:

```
Starting 1 compile threads for 1 files...
Compiling "HelloWorldScript"...
Starting assembly of HelloWorldScript
0 error(s), 0 warning(s)
Assembly succeeded

Compilation succeeded.
```

Save Succeeded   Line 5\7

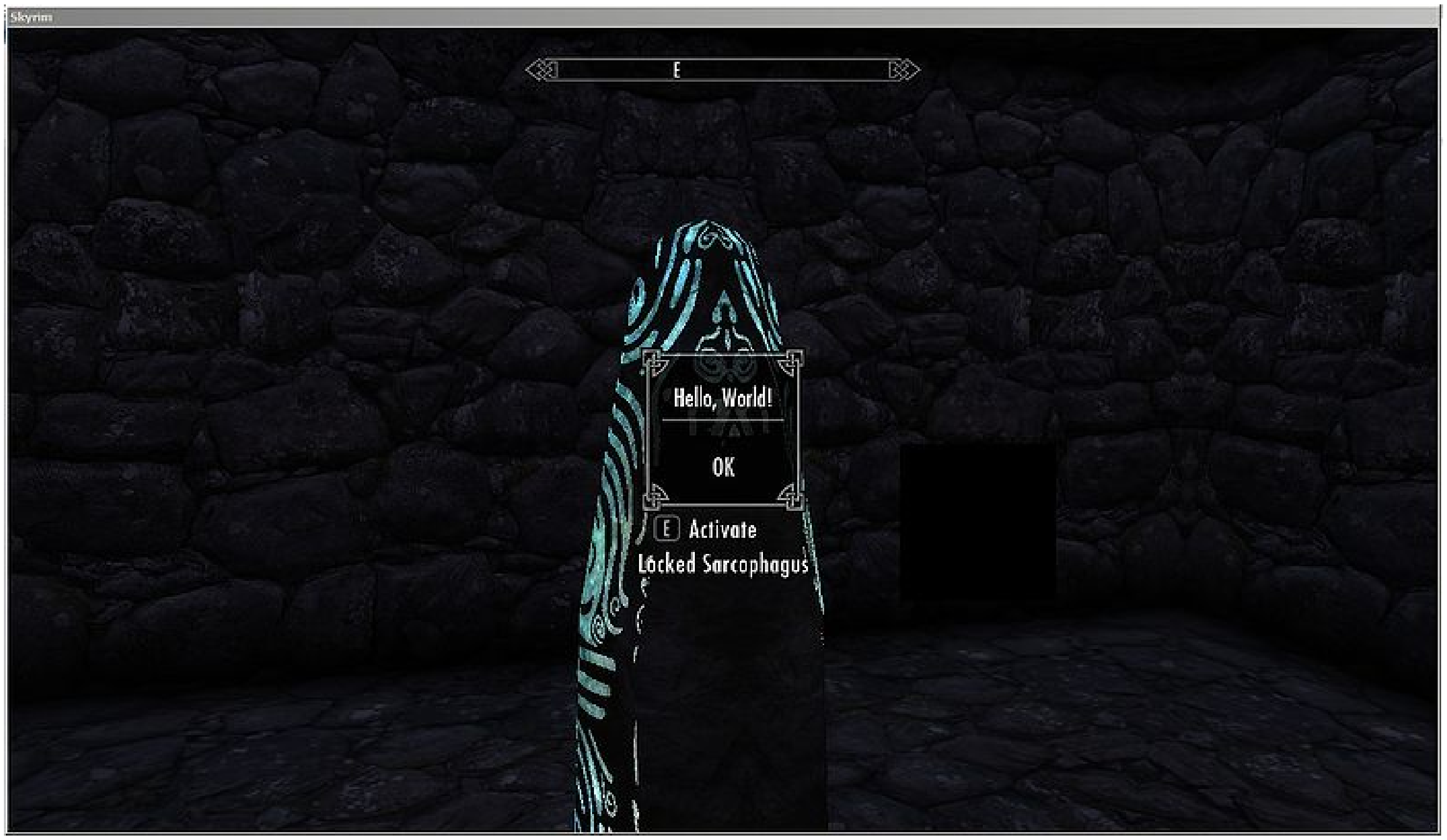
"Hello, World"

Now to go into the game and try it out. (Make sure to save your plugin, and set up the game to load it first.)

Once you're in the game, hit ~ to bring up the console. Type:

```
coc MolagBalVoiceCell
```

to move to the cell. Walk over to the pillar and activate it. You should see your new message box:



That's it. You've made an object that can respond to the player's actions!

Next, you can learn how to use variables and conditional statements to make your script a bit more sophisticated.