Bethesda Tutorial Quest Objectives

creationkit.com/index.php

Contents

Overview

This tutorial will show the reader how to create objectives and quest targets so the player can keep track of their current goals more easily. We'll also tie up some loose ends that could confuse or frustrate the player.

The reader will learn:

- How to create aliases and objectives
- How to turn objectives on and off
- How to handle completing and shutting down a quest
- How to insert additional quest stages

Objectives

When a player goes through a quest in Skyrim, they're able to see the quest in their menu, see their current objective, and get an idea of where they need to go for the next part of the story.

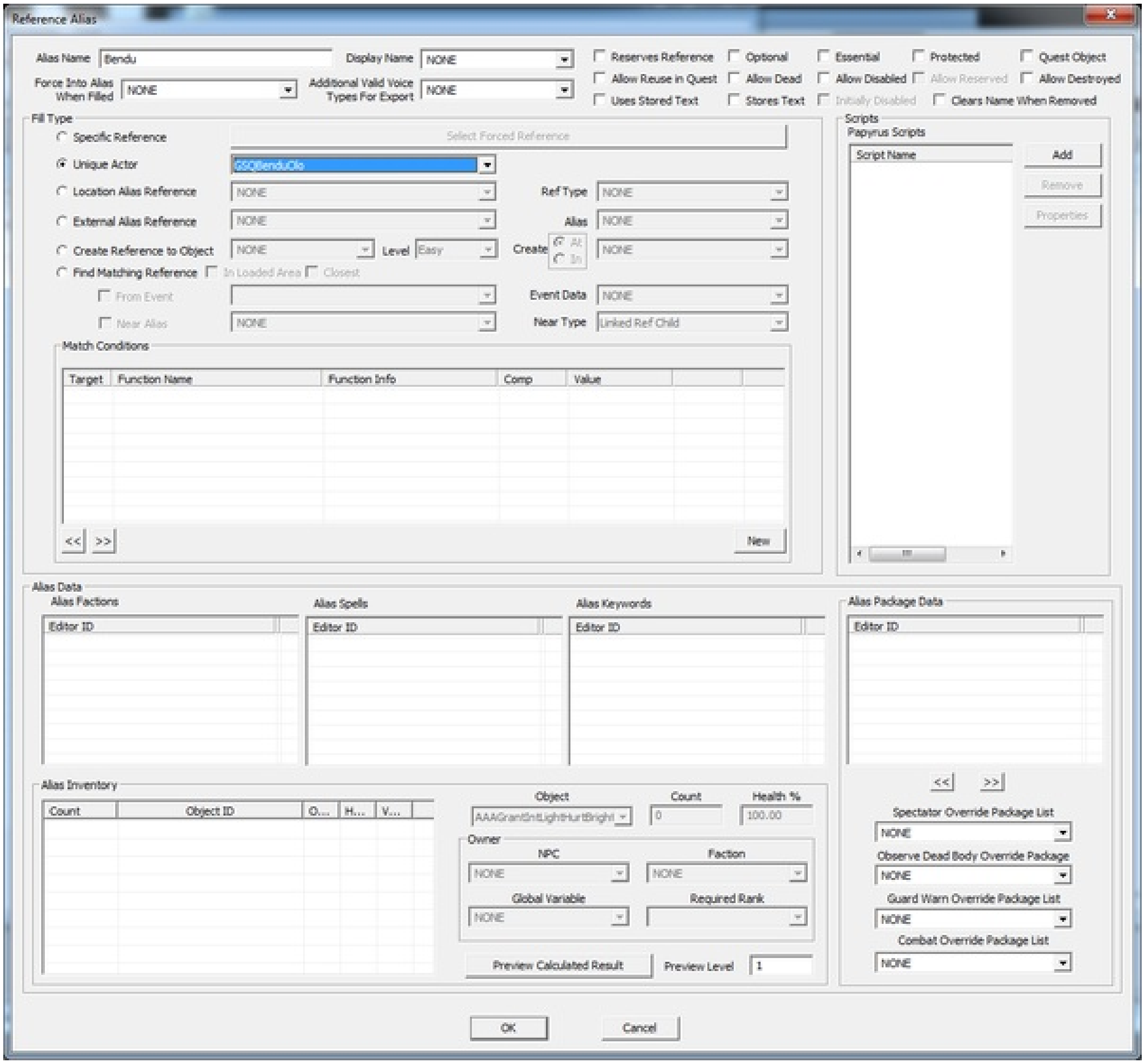
Our first step in making objectives is setting up their targets, which means our first dive into the world of Aliases.

Aliases

You can think of Aliases as the quest defining roles that various people and places will play in the course of its story. Each alias points to a specific reference or place in the game world. For our example quest, these roles include Bendu, the thief, and the amulet. Each of these has to get defined in a slightly different way.

Open up your quest again and this time go over to the Quest Aliases tab. It's pretty much a big empty table at this point. Right-click in the table and select "New Reference Alias" to open up a large intimidating window. Thankfully we'll be ignoring almost all of it for now (most of this functionality will be explained later in the tutorial devoted to aliases).

Fill in "Bendu" for the Alias Name field at the top, then check the "Unique Actor" radio button and select "GSQBenduOlo" from the pulldown menu it enables. When you're done, the window should look like this:



Reference Alias

Alias Name: Bendu
Display Name: NONE
Force Into Alias When Filled: NONE
Additional Valid Voice Types For Export: NONE

□ Reserves Reference □ Optional □ Essential □ Protected □ Quest Object
□ Allow Reuse in Quest □ Allow Dead □ Allow Disabled □ Allow Reserved □ Allow Destroyed
□ Uses Stored Text □ Stores Text □ Initially Disabled □ Clears Name When Removed

Fill Type
○ Specific Reference
● Unique Actor: GSQBenduOlo
○ Location Alias Reference: NONE Ref Type: NONE
○ External Alias Reference: NONE Alias: NONE
○ Create Reference to Object: NONE Level: Easy Create ○ At ○ In NONE
○ Find Matching Reference □ In Loaded Area □ Closest
□ From Event Event Data: NONE
□ Near Alias: NONE Near Type: Linked Ref Child

Alias Inventory
Object: AAAGrantSntLightHurtBright Count: 0 Health %: 100.00
Owner
NPC: NONE Faction: NONE
Global Variable: NONE Required Rank:
Preview Level: 1

Spectator Override Package List: NONE
Observe Dead Body Override Package: NONE
Guard Warn Override Package List: NONE
Combat Override Package List: NONE

- If you cannot find "GSQBenduOlo" on the list, then go back to his base object properties and check if he has the Unique flag on. The list of Unique Actors will show only actors with "Unique" flag.

- On some computers, the "Reference Alias" window is so tall that you won't be able to reach the bottom of it to click the "OK" button. To get around this, you can click back in the text box for "Alias Name" when you are all done editing and press the ENTER key on your keyboard. This will save the work done in the window.

Hit OK to close this window. Right-click and select "New Reference Alias" again to open up a fresh new alias window. Put "Thief" for the Alias Name. Because the thief is not a unique actor (since it's created from a template), we have to use the Specific Reference fill type this time. Click the "Specific Reference" radio button, and click the large "Select Forced Reference" button to its right. We now have two ways of choosing the reference:

1. Select "ReachwindEyrie01" from the Cell dropdown, and then "GSQThief" from the Ref dropdown.
2. If you have Reachwind Eyrie loaded and visible in your render window, you can click the "Select Reference in Render Window" to turn your mouse pointer into a target. Double-click on the big green M to select it.

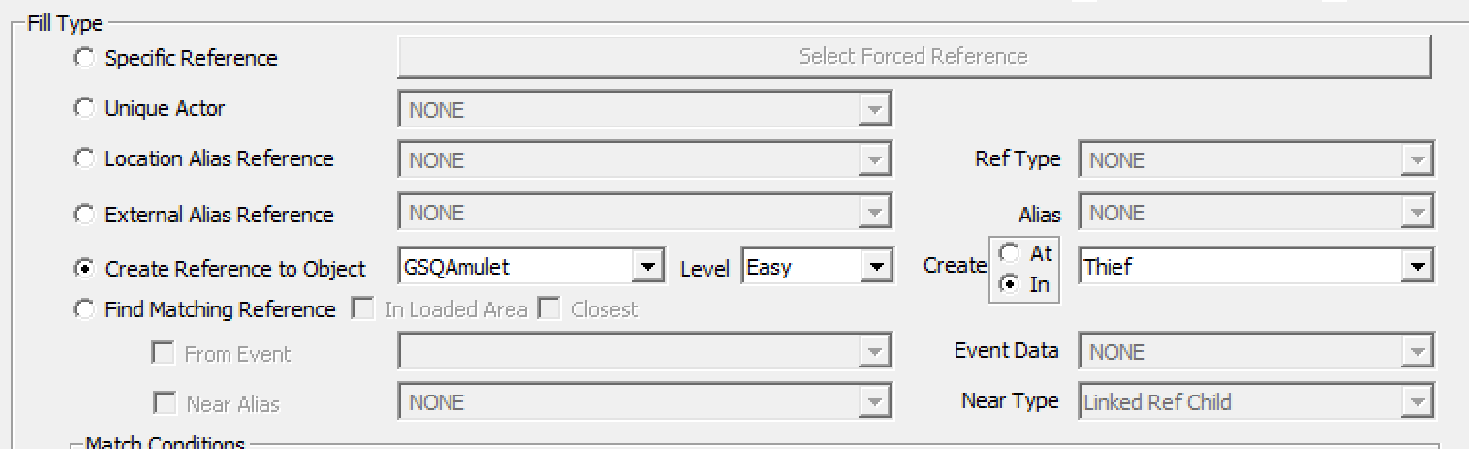
Either way, click OK in that window and then the alias window to finish making our second alias.

We could, of course, also use a specific reference to point to Bendu instead of choosing him from the Unique Actor list. For some rather complicated reasons, it's more memory efficient to use the Unique Actor type than the specific reference, when possible.

Now it comes time to make our third alias (for the amulet), but this one is tricky. Only actors can be unique, so the Unique Actor type won't work. And we can't point to it as a specific reference because it only exists in the inventory of our thief. So what we'll do is have the alias itself create its own reference.

Create a new reference alias one more time. Give it the name "Amulet" and click the radio button for "Create Reference to Object" under Fill Type. Moving from left to right, then we want to:

- Select "GSQAmulet" from the first pulldown menu.
- Ignore the Level/Difficulty menu (since we aren't making an actor, there's no notion of the object's difficulty).
- Click the radio button for "Create **In**"
- Select Thief from the final pulldown.



Fill Type

○ Specific Reference — Select Forced Reference

○ Unique Actor — NONE

○ Location Alias Reference — NONE — Ref Type — NONE

○ External Alias Reference — NONE — Alias — NONE

● Create Reference to Object — GSQAmulet — Level — Easy — Create ○ At ● In — Thief

○ Find Matching Reference □ In Loaded Area □ Closest

□ From Event — Event Data — NONE

□ Near Alias — NONE — Near Type — Linked Ref Child

Match Conditions

As you might imagine, this will create a new reference to the amulet in the thief's inventory when the quest begins (at the start of the game). Right now this would give the thief *two* copies of the amulet, so open up GSQThief and remove the amulet from its inventory.

Creating Objectives

Now move over to the Quest Objectives tab in your quest window. This is where we'll actually set up the objectives as the player sees them.

Each objective has an index (a number we'll use to refer to it), text that will be displayed to the player, and a list of targets that will get arrows placed on them in the player's map and in-game interface.

To make a new objective, right-click in the table at the top of the window and select "New" from the menu. In the "Objective Data" area below there, change the Index value to "10" and make the Display Text say "Kill the thief."

Generally, we try to match up the objective index number with stages where you would want to see that objective. For more complicated quests, things can get kind of wooly, but for straightforward ones, as we'll see, this makes things a lot easier to follow.

Finally, in the table below the Index and Display Text fields, right-click and select "New" to create a new quest target. This creates a big scary "NO TARGET" until you select something from the "Target Alias" pulldown below. In this list, you'll see all the aliases we defined earlier. Choose "Thief," and we can move on. (The "NO TARGET" will remain until we click away on something else.)

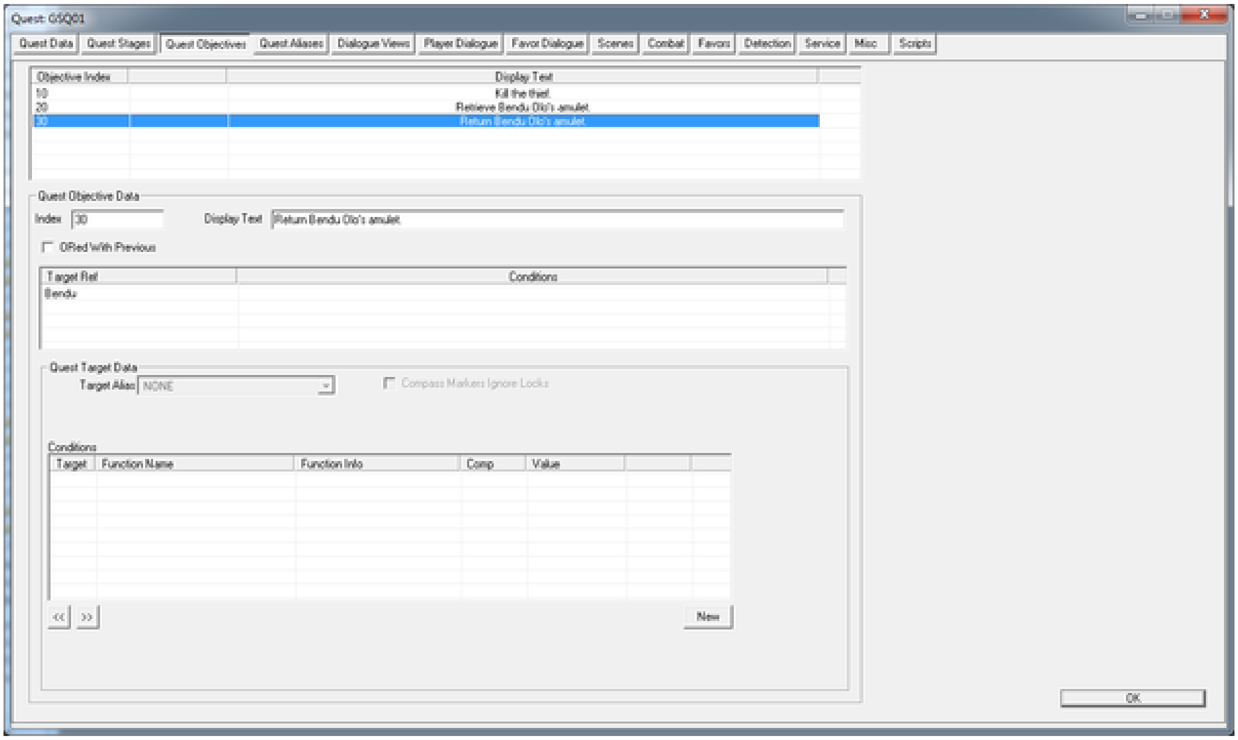
Now make another objective:

- **Index:** 20
- **Display Text:** Retrieve Bendu Olo's amulet.
- **Target Alias:** Amulet

And one more:

- **Index:** 30
- **Display Text:** Return Bendu Olo's amulet.
- **Target Alias:** Bendu

Your window should look something like this when we're done.



Objective Index		Display Text
10		Kill the thief.
20		Retrieve Bendu Olo's amulet.
30		Return Bendu Olo's amulet.

Index: 30
Display Text: Return Bendu Olo's amulet.
□ ORed With Previous

Target Ref	Conditions
Bendu	

Target Alias: NONE
□ Compass Markers Ignore Locks

Target	Function Name	Function Info	Comp	Value

Now we have our objectives made, we just have to tell the game when to show them.

Setting Objectives

Remember all of those `SetObjectiveDisplayed()`'s I told you to ignore? This is where they come into play. Should the following NOT work for some odd reason, those are a safety net. They will do what this is supposed to do. I suggest doing both to have all the bases covered.

Go back to the Quest Stages tab. We're going to add some logic in here as the quest advances.

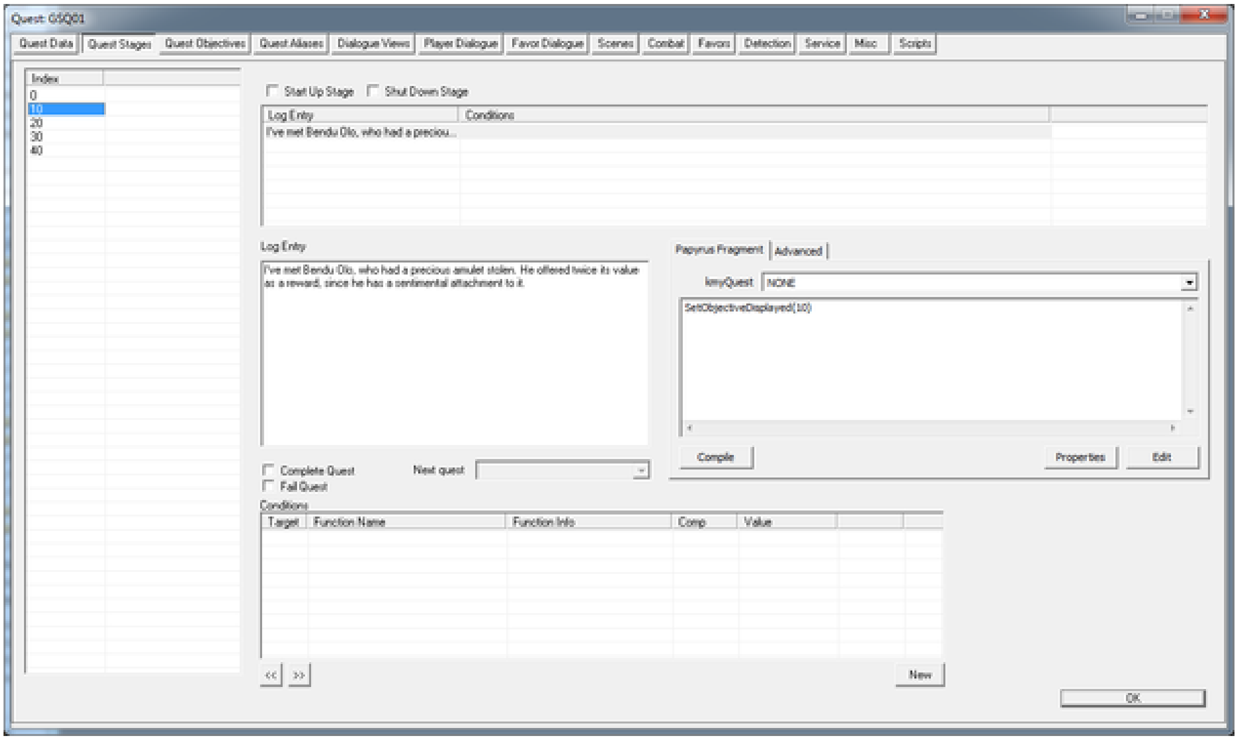
Click on stage 10 in the list on the left. In order to add script logic or a log entry to a quest stage, it needs to have at least one "quest stage item", so right-click in the table that says "Log Entry" at the top and select "New."

Log entries are the bits of text that will appear in the player's journal. Only the most recent log entry will be visible, so you can use this to update the story of the quest as it changes. Down in the "Log Entry" area, put the following:

> *I've met Bendu Olo, who had a precious amulet stolen. He offered twice its value as a reward, since he has a sentimental attachment to it.*

Click over to the text field on the right (in the Papyrus fragment tab). Doing so will pop up the spell-check window, complaining that "Bendu" and "Olo" are not proper words. Ignore both "mistakes" and we'll add the script logic.

In the Papyrus area, put: `SetObjectiveDisplayed(10)`. Press the "Compile" button to make sure you did it right. As you might guess, this set the quest's current objective to 10, and give the player the appropriate quest targets. Because it's the first objective shown for this quest, it will also cause the "STARTED" banner to be shown to the player.



Index
0
10
20
30
40

□ Start Up Stage □ Shut Down Stage

Log Entry	Conditions
I've met Bendu Olo, who had a preciou...	

Log Entry

I've met Bendu Olo, who had a precious amulet stolen. He offered twice its value as a reward, since he has a sentimental attachment to it.

Papyrus Fragment | Advanced

kmyQuest NONE

```
SetObjectiveDisplayed(10)
```

□ Complete Quest
□ Fail Quest

Next quest

Conditions

Target	Function Name	Function Info	Comp	Value

- You might wonder why the extra step of creating the "stage item" is necessary - in more complicated quests, you can have multiple stage items, some or all of which use conditions to control whether or not they happen (for example, to allow variant log entries depending on a choice the player had made earlier in the quest). In this case, our quest is simple enough to only need one stage item in each stage.

Now select stage 20 from the list on the left, and create a new stage item for it. Remember, stage 20 is after we've killed the thief but before we've gotten the amulet. We have two things to do here -- tell the player that the previous objective (to kill the thief) has been accomplished, and set their next goal. We do that with two lines of script:

```
SetObjectiveCompleted(10)
SetObjectiveDisplayed(20)
```

Note that "Completed" and "Displayed" are two different flags -- you can complete an objective that the player has never seen.

We can leave the journal entry here blank, because the overall story of the quest hasn't changed.

Do the same thing for stage 30, except this time your logic should be completing objective 20 and displaying objective 30.

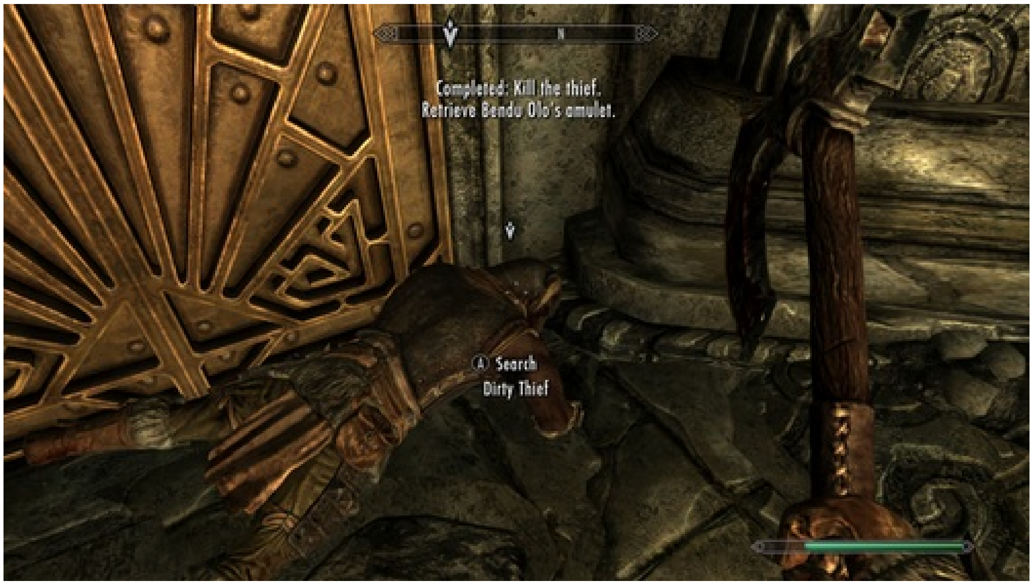
Finally, in stage 40, we just need to complete objective 30. You should also check the "Complete Quest" checkbox below the Log Entry text field, so that the "COMPLETED" banner will be shown when stage 40 is set. (This will also move the quest to the inactive portion of the player's journal, keeping it somewhat tidy.)

The final journal entry should sum up what happened in the quest, since it's now in the past. For example:

> *I met Bendu Olo and was rewarded handsomely for retrieving his stolen amulet.*

Trying it out

Now when you progress through the quest, your progress will be visible in your journal and you will have quest targets to the various plot points.



Completed: Kill the thief.
Retrieve Bendu Olo's amulet.