

Bethesda Tutorial Dialogue

Contents

Overview

This chapter will walk you through creating specific dialogue for our tutorial quest, "Bendu Olo's Only Hope."

The reader will learn:

- How dialogue for Skyrim is authored in the Creation Kit.
- How to script events based on the player's dialogue choices.

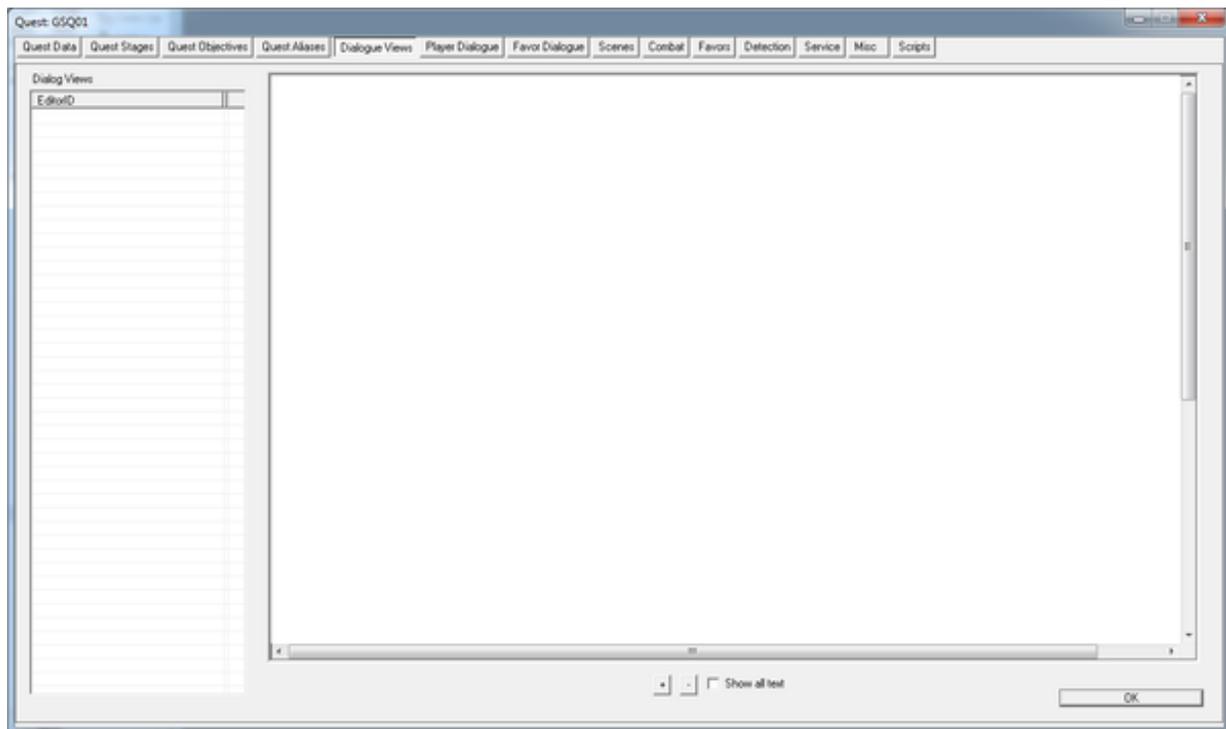
Dialogue Views

Skyrim provides a simple visual layout system so you can see the flow of your dialogue as you're creating it.



If you're used to the old-style dialogue creation system from Fallout 3 and earlier, it's still available to you under the [Player Dialogue tab](#) (and in fact, the views are just a snazzier front-end to that same system).

To get started with the view, navigate to your quest (GSQ01) and double click to open the quest window. Then navigate to the [Dialogue Views tab](#). Your window should now look like this:

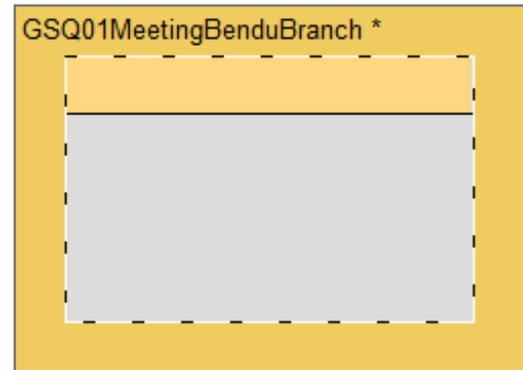


To make a new view, right-click in the table in the left part of the window and select "New." You'll be prompted to give an ID to this view, and it will automatically have the quest ID ready for you to use as a prefix. Call this view "GSQ01BenduView." Click on it in the table to select it.

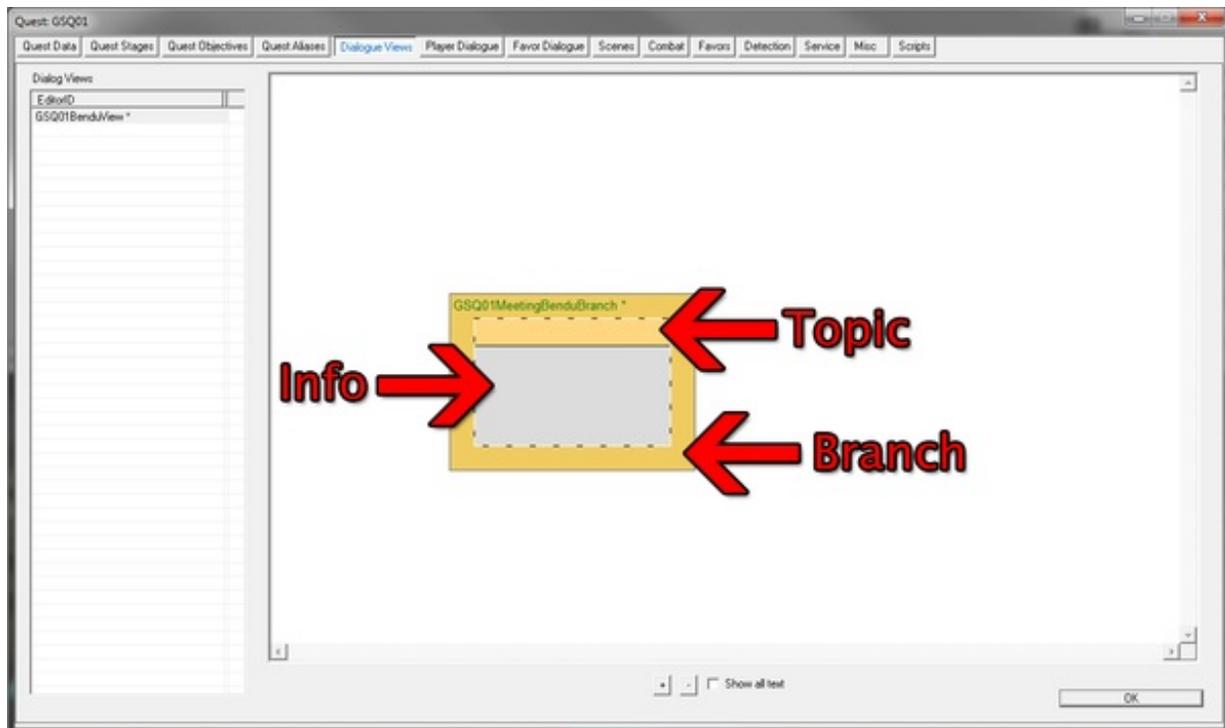
We haven't added anything to the view yet, so it's a boring white expanse. Right-click in that expanse, and select "Create Branch." That will prompt you to name this branch, which you should call "GSQ01MeetingBenduBranch". One more prompt for naming a topic (you can just accept the default of "GSQ01MeetingBenduBranchTopic"), and on hitting Enter you'll see a series of rectangles that probably doesn't mean much to you.

Before we go on, there's some terminology that can be confusing and we need to get straight:

- **Response:** A single line of dialogue said by an NPC. The text is limited to 150 characters. It can be assigned an emotion and animation for the NPC to play physically while saying it.
- **Info:** A collection of responses put together for a single circumstance. When a character says multiple lines one after the other, that's a single Info with multiple responses.
- **Topic:** A collection of infos that an NPC might say in a given situation, often in response to a specific dialogue choice.
- **Branch:** A collection of topics that comprise a particular piece of conversation flow from the player's perspective.



So in that set of clicks and typings above, we created a branch, which automatically created a topic for itself. The topic currently contains no infos.



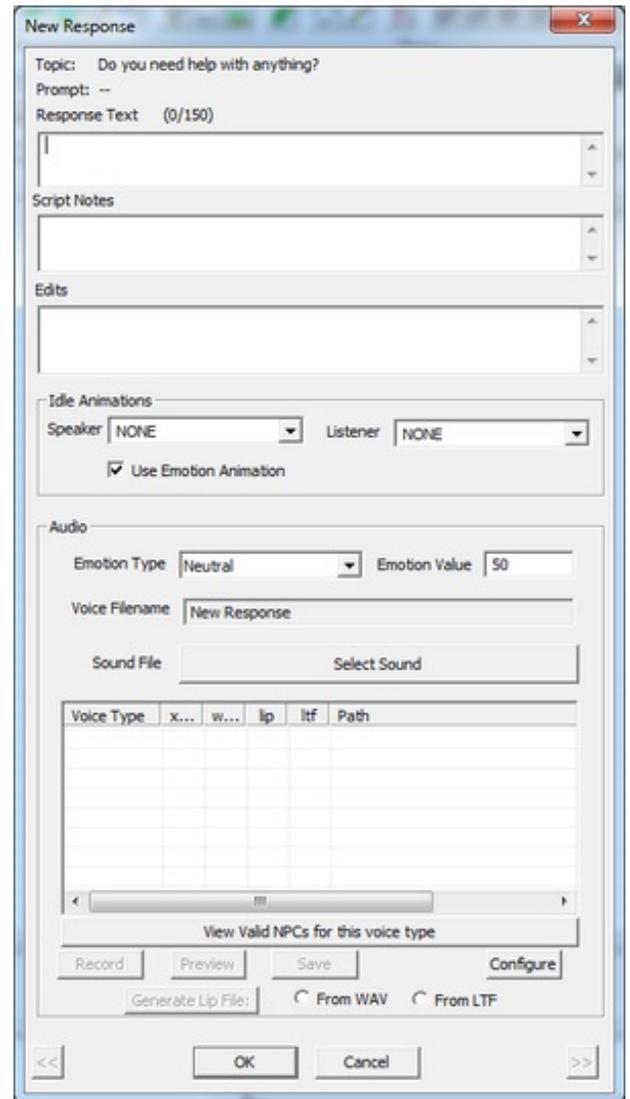
Topics

Double-click on the topic to open the Topic Window.

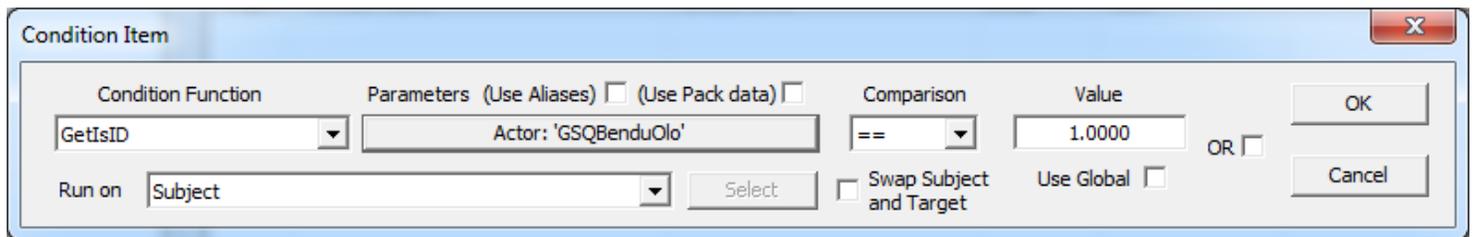
In the "Response Text" field, enter: "Why, yes, now that you mention it. Some dirty thief has made off with my amulet. Would you be so kind as to... ahem... fetch it back?"

Now hit the OK button at the bottom of the window **twice**. The window will disappear and you'll see the Topic Info window.

- ☞ The first time you hit the "OK" button in the response window, the editor runs a simple spell-check on what you just entered. If you copied that precisely, there are no spelling errors, so the first button press won't do anything.



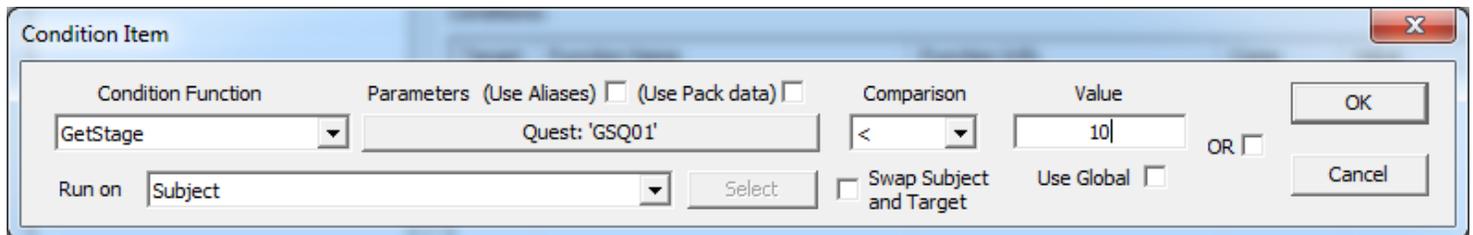
Topic Infos



Hit OK on the condition window to close it out, and you should then see it in the list. Now only Bendu can say it, but we don't want him saying it all the time; just before he's given the quest to the player. So we're going to add one more condition. Right-click in the condition list and select "New" again.

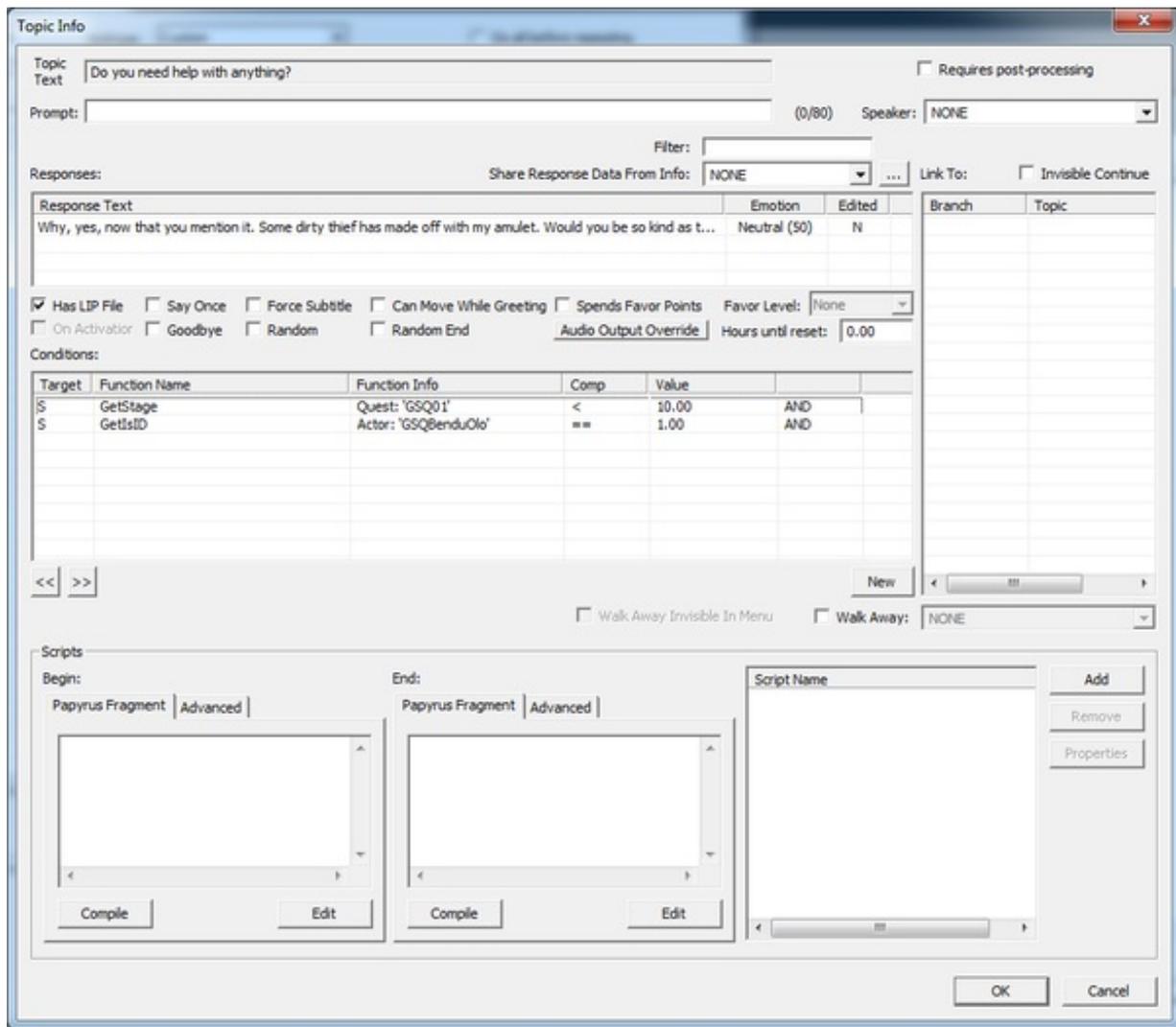
We're going to further limit this bit of dialogue based on the quests current stage. So click on the Condition Function pull-down and select "GetStage." (You can type the first few letters to skip around in the list -- much faster than scrolling.) Now if you click the "INVALID" button to select parameters, it will bring up a list of quests rather than base objects (since it knows that the GetStage condition only deals with quests). Select "GSQ01" from the menu and hit OK.

Now all that's left is to tell the condition which quest stage we care about. Since we know the quest is starting at stage 0, we could select "==" for our comparison and set the value to "0," but that might break this condition if we ended up adding some stage before this one. The safer way to set the condition is to check that our current stage is lower than 10, so choose "<" from the Comparison list and put "10" in the value field.



- Note that you can reorder conditions with the left and right arrow keys or the buttons at the bottom left of the condition window. In this example the order doesn't matter, but if you started setting up your conditions with ORs as well as ANDs, it would.

Our final topic info window should look like this:



Hit OK to close the Topic Info window, then OK again in the Topic window to close it.

Linking Dialogue

Finally, we're back at the Dialogue View! With our new bit of dialogue present. If you want to actually see all the text (and why wouldn't you?), click the "Show All Text" checkbox at the bottom of the view.

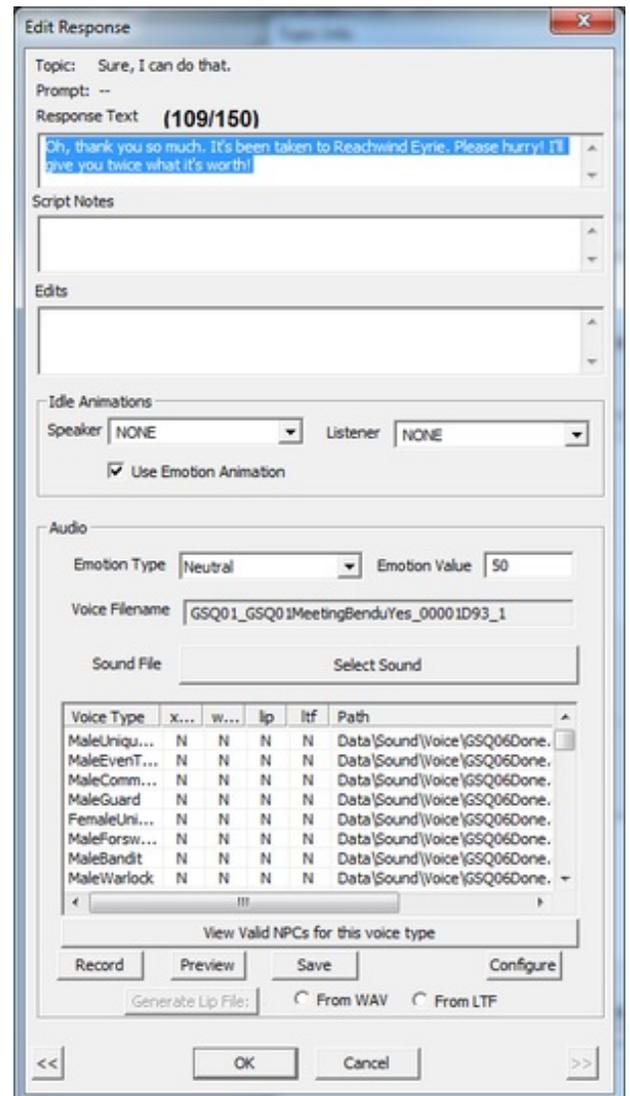
Since Bendu is posing a question with his line, we want to give the player a chance to answer. Right-click in the branch area and select "Add Topic." (If you don't see that in the contextual menu, you probably did not click in the branch.)

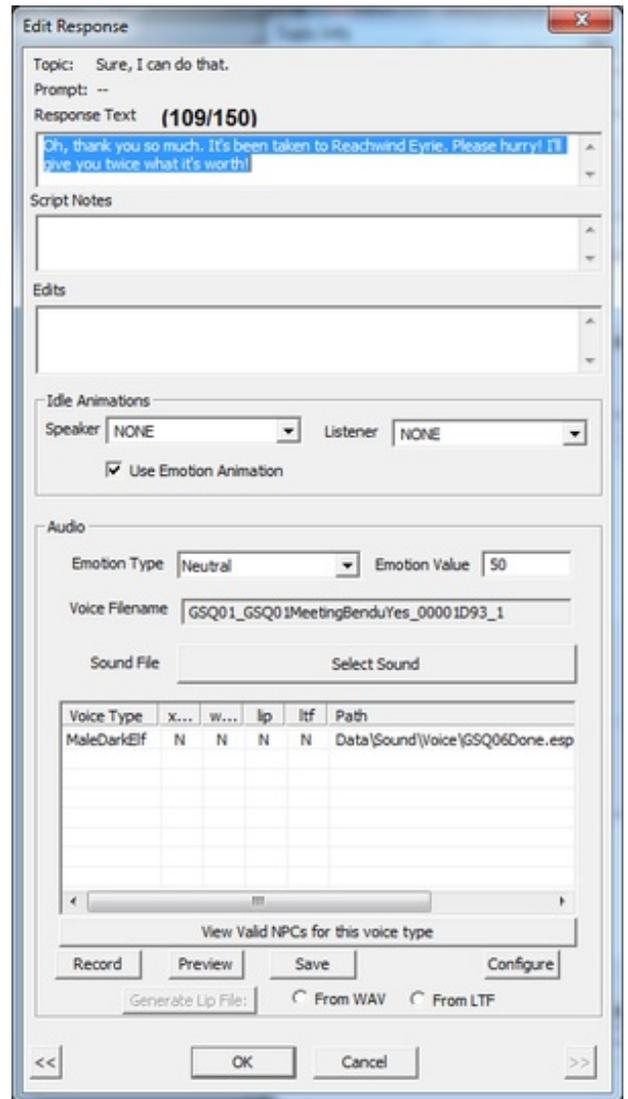
This will open up another topic window. You'll have to assign an ID to this one; "GSQ01MeetingBenduYes" works, since this will be our choice where the player agrees to help. Put "Sure, I can do that." as the topic text, and double click in the info table to make a response. For the text, enter "Oh, thank you so much. It's been taken to Reachwind Eyrie. Please hurry! I'll give you twice what it's worth!" This time when you hit OK the spell-check will complain about "Reachwind." Just hit the Ignore button to dismiss it, hit OK one more time, and you're back to the Topic Info window.

Now that you're here, double click on the response text again to bring it back up. Notice the list of sound files at the bottom of this window.

Since we haven't put any conditions on this dialogue, the game still thinks everyone in the world can say it, and will try to make appropriate sound files for every voice type. This is a big waste of disk space and (if you're planning to record voice) a waste of time recording unnecessary performances. So close this window and add a GetIsID condition looking for GSQBenduOlo (like we did in the previous topic info). Now if you look at the response again, you'll see that it knows that this line is only said by a MaleDarkElf voicetype, and so only lists the one file.

Now hit OK on the Topic Info window and the Topic window until you're back looking at the dialogue view again. You'll have a new topic floating in your branch, which has expanded to accommodate it.

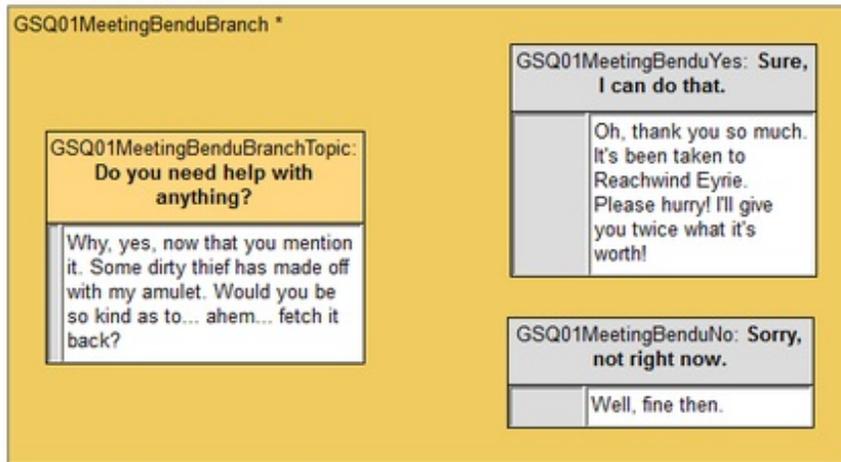




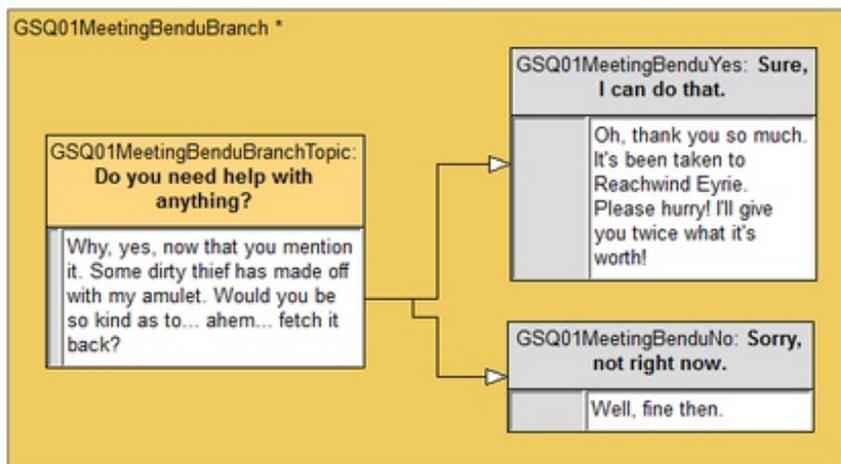
The beauty of Dialogue Views is that we can arrange these topics however makes sense for us. At runtime, the game doesn't care about this positioning; it's purely for the designer's convenience. So grab the new topic by its top bar and drag it to a convenient spot.

Make one more topic, this time with a Topic Text of "Sorry, not right now." Set Bendu's response to be "Well, fine then." Don't forget to put a GetIsID condition on the Topic Info. Once you've created those two, the Dialogue View

should look something like this.



All these topics exist within the branch, but the game doesn't know anything about the flow between them. We have to link them together. Click on GSQ01MeetingBenduBranchTopic to make it active. Then click once more on the text of the line (you'll see the cursor turn into a little hand when you roll over it) and drag a line to the GSQ01MeetingBenduYes. Do the same thing, dragging a line from the initial topic to the GSQ01MeetingBenduNo topic.



Now, when the player selects the prompt "Do you need help with anything?", Bendu will say his line and then present the player with another list of choices based on the links from that topic. The choices will be listed in the order they were linked from that topic.

Scripting Dialogue

We're going to make our first foray into the scripting system here, but it will be very straightforward, so don't worry if that's not your cup of tea.

Right now we have a dialogue flow set up, but the game has no notion of what the various choices mean. When the player says, "Yes," we have to tell it that the player has accepted the quest and we're now at the next stage.

Double-click on the text "Oh, thank you..." under the GSQ01MeetingBenduYes topic. (Note that this brings us

directly to the Info window without first bringing up the Topic window.)

At the bottom of this window, you see fields for Scripts marked "Begin" and "End." The Begin script runs when the NPC starts saying the line, and the End script will run when he finishes. (In scripting terminology, these little pieces of script are called "script fragments" - see [Topic Info Fragments](#) for more details.)

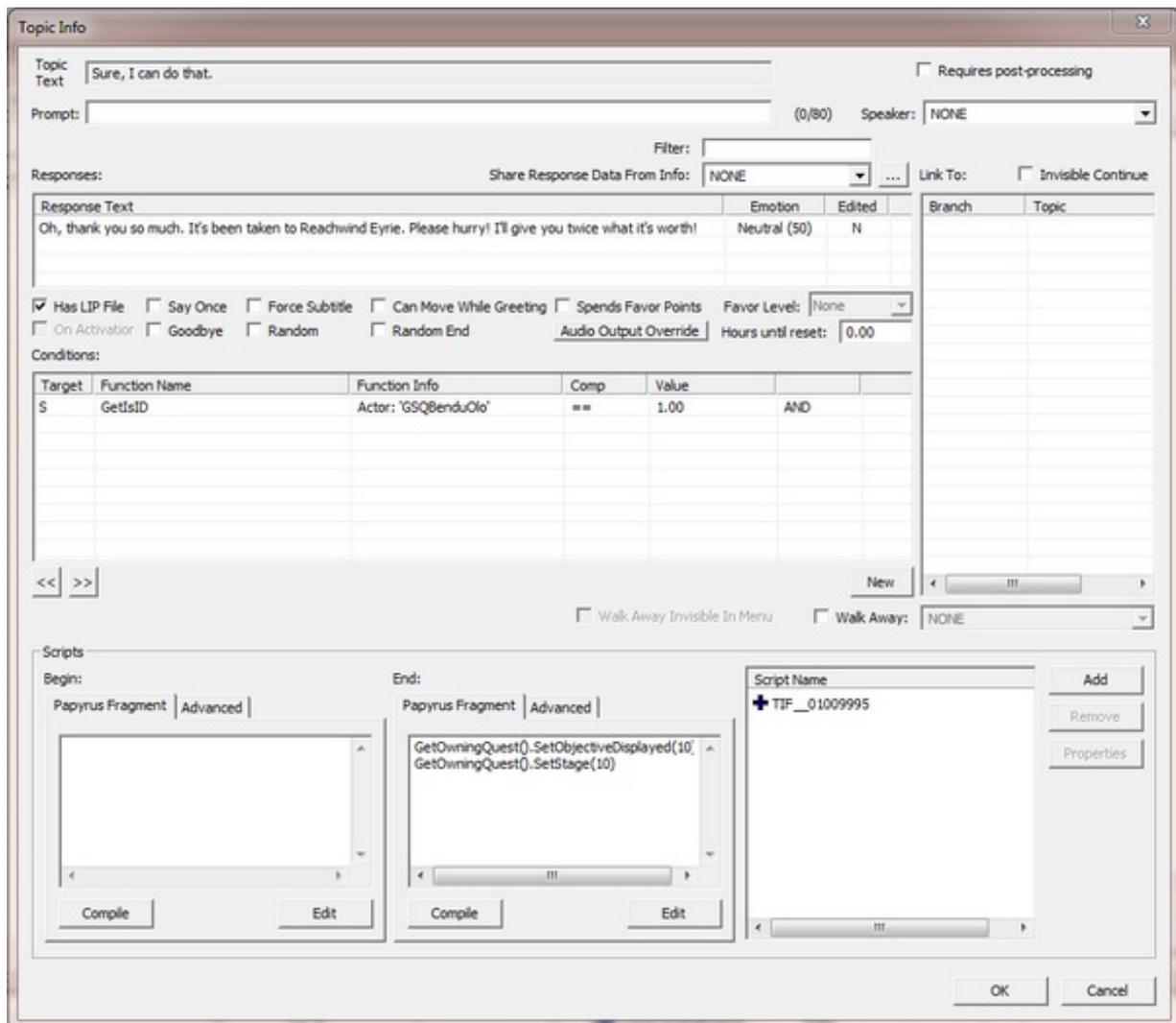
In the End field, enter this script command:

```
GetOwningQuest().SetObjectiveDisplayed(10)
GetOwningQuest().SetStage(10)
```

The `SetStage(10)` portion, as you might expect, sets a quest's current stage to 10. The `GetOwningQuest()` part just says that the quest we care about is the one that contains this dialogue. (Sometimes you might want a line in your quest to muck around in a different quest, so we have to be specific.)

If you press "Compile" you'll see some gobbledygook pop up under "Script Name" to the right. You can ignore that. (If you typed something wrong, though, the Editor will complain at you.)

When you're ready to close the window, it should look like this:



And that's it! Close the window, and you're good to go. You'll note that in the dialogue view, this info's text now has "[S]" at the front of it, so we can easily spot which bits of dialogue have game logic attached to them without having to open each one.

Branches and Conditions

What we've just done is craft a Branch of dialogue. By default it's a "Top-Level" branch (which is why it's colored orange in the view). You don't need to worry about the other branch types right now; just know that a top-level branch is available as one of the first things an actor is able to say to the player.

The topic that has a yellow-orange bar at the top of it has been flagged as the branch's Starting Topic. When an actor is determining what choices to offer the player, it looks for starting topics and displays the prompts for all of them which are valid. If you remember, we put a condition on our starting topic that the quest stage had to be less than 10. Once the player chooses to accept the quest, we set the quest stage to 10, which means that starting topic is no longer valid; it won't appear in the actor's topic list anymore.

Non-starting topics can have much smaller lists of conditions, since they are typically only accessed if the starting topic for their branch is valid. That's why we just limited them to a specific NPC so we wouldn't have extra voice files. From a gameplay perspective, it wouldn't have mattered.

Once the player chooses a topic with no links out of it, the game will again look for valid top-level topics to give the player choices. If none are valid, the NPC will release the player from dialogue.

Try It Out

- ◆ **Important!** As of 1.7 the game uses SEQ files in order to track *Start-Game Enabled* quests you've added, and possibly ones you've altered. These files are needed for your dialogue and scenes to work properly!

See this [tutorial](#) on how to Create Start-Game Enabled Quest (SEQ) Files

Load the game with your plugin and go meet Bendu. (Remember, you can hop right there with `COC MixwaterMillWorkersHouse`.) He should now have something to say to you.



There are a few problems with the dialogue as it currently stands:

1. It flashes directly to the next choice list. (Since there's no audio, if you don't have captions on, you won't have any idea what he said. Even if you do, they probably flashed by so quickly you couldn't read.)

2. If you refuse to help, he'll just go right back to the same top level topic, which feels odd.

Luckily these are easily addressed.

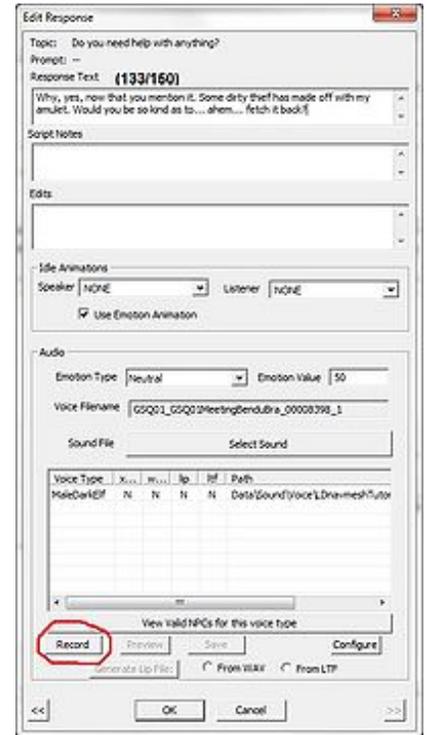
Recording a Temp Track

Open up the GSQ01MeetingBenduBranchTopic and navigate through to its Response window. At the bottom of the window, you'll see a "Record" button.

Press this button and, if you have a microphone hooked up to your computer, you can record a performance of the line yourself. If you don't (or you're self-conscious and don't like the sound of your own voice) you can just record a few seconds of silence, enough to give the captions time to be read. (Note: you MUST have a microphone to record anything. If no microphone was plugged in at the time you started the Creation Kit, the record button will not do anything) Hit the Save button to commit this file to disk.

Do the same thing for the other two topics we made. (While you're in there, consider setting the emotions for those lines, so that Bendu will show Disgust or Happiness depending on your choice.)

In the "No" topic, click the box that says "Goodbye." This will drop the player out of dialogue when it's selected, so you won't end up back at the topic list after the line is done. (You should also make the "Yes" topic a goodbye. Since Bendu only has one top-level topic right now, that's not necessary, because we know he'll have nothing else to say when he's done here. But if another quest had dialogue for him, it might be confusing to the player.)



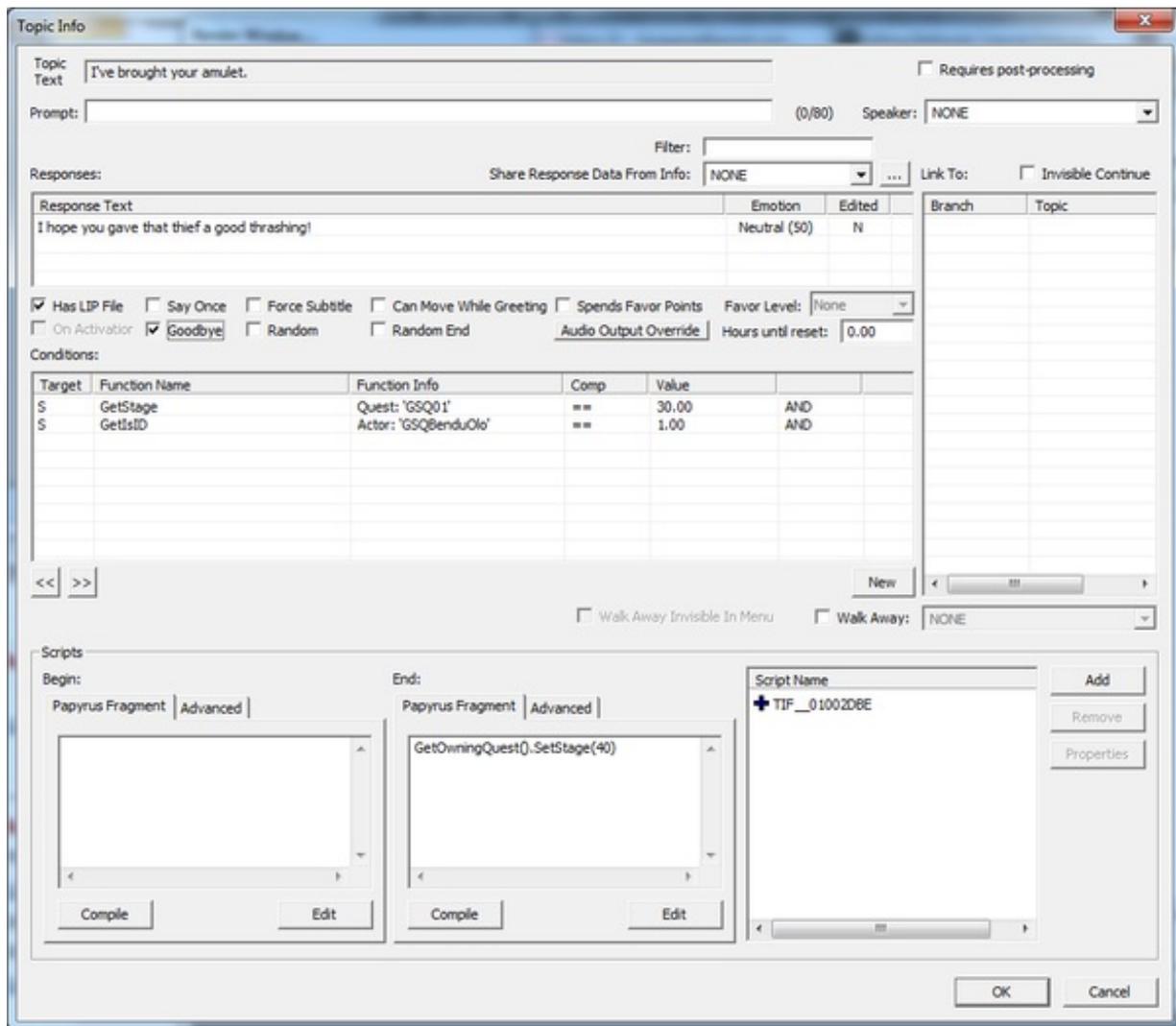
- ◆ The game engine occasionally fails to playback audio files saved in the WAV/XMW format for reasons yet to be deciphered. To workaround this bug, FUZE equivalents of the voice assets need to be created. This can be done by converting the source WAV files (when applicable) to the XWM format, and subsequently to FUZ by pairing them with their LIP files. More information on the conversion process may be found [here](#).

Additional Dialogue

While we're here, let's also make the dialogue that will close out the quest when we come back with the amulet.

In the dialogue view, right-click to create a new **branch** (not topic), and call it "GSQ01BenduFinishBranch." Accept the default name for the starting topic, and make the topic text, "I've brought your amulet." Give him something good as a response (I went with "I hope you gave that thief a good thrashing!"). It will want two conditions; one that restricts it to Bendu, and another that limits it to stage 30 of the quest. Finally, set it to be a Goodbye, and set the quest to stage 40 when it's said.

You should be able to do all this by now. It will look like this when you're done.



We still haven't dealt with the amulet itself yet, and the quest has no way to proceed from stage 10 to 30, so we have work to do!

- ☺ This might seem like a lot of work and unnecessary data-fumbling to just get characters to say a few lines. Part of the complexity is historical -- because it evolved from the old Morrowind text-based dialogue system, the organization can seem a little sideways and non-directed. But the same system also handles multiple characters saying the same line, combat barks, player greetings, hit responses, etc. Flexibility sometimes means complexity.

With some practice, all of this becomes second nature, and you'll be able to mechanically work the dialogue system as fast as you can write the lines themselves.