

CHAPTER 6: QUESTS, DIALOGUE AND VOICE ACTING

THE BASICS OF SETTING UP QUESTS

This section will cover the basics of setting up quests in the Creation Kit.

Before opening the Creation Kit, I would strongly recommend that you plan out your quest first. Start with a draft outline of your story and use that to help define the ‘beats’ of the quest and the dialogue between the player and the NPCs involved.

A simple fetch quest, for example, is generally comprised of the following four stages:

Stage 0 - The NPC is ready to give the player a quest.

Stage 10 - The NPC has told the player what to do or what to find and what kind of reward they should expect to receive.

Stage 20 - The player completes the task and needs to return to the quest-giver NPC.

Stage 30 - The player tells the quest-giver NPC that the task is done and has received a reward.

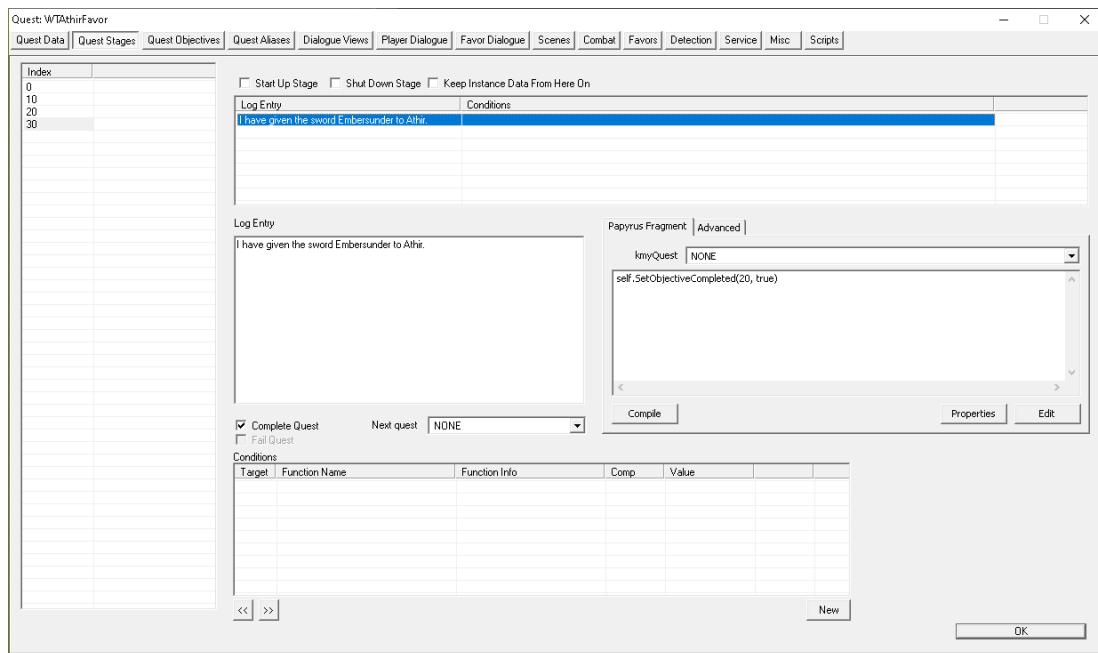


Figure 664 - A simple fetch quest.

You should also consider how your quest immerses the player in the game world, and what kind of reward the player will receive for completing it.

Teaching you how to write a questline is well beyond the scope of this tutorial, but I would highly recommend watching Josh Sawyer’s GDC talk on [Choice Architecture, Player Expression, and Narrative Design in Fallout: New Vegas](#).

Drawing the player's route on a screenshot of the world map can be a useful way to help plan out your quest.

The first quest in Wyrmsooth can be diagrammed as follows:

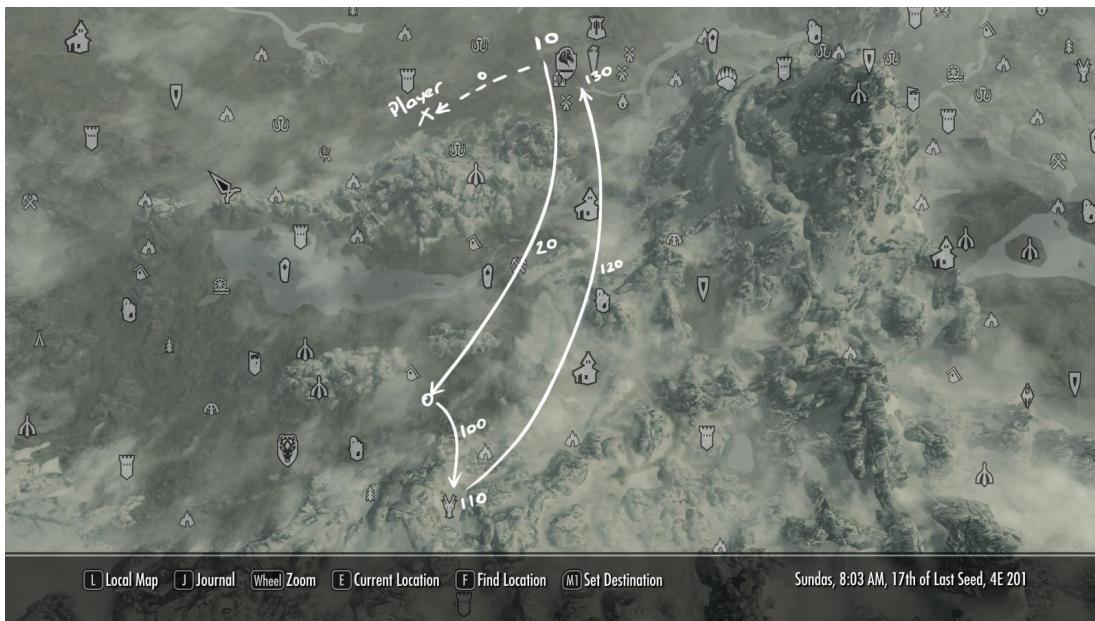


Figure 665 - Quest beat diagram of the first quest in Wyrmsooth.

Once the player meets quest starting requirements, an NPC named Theodyn Bienne spawns in at the Bannered Mare in Whiterun and travels to the player's current position to deliver a message (stage 0).

Once the player has talked to Theodyn, the current objective is to speak to his employer, Lurius Liore, currently staying at the Bannered Mare (stage 10).

Lurius will divulge more information about the threat and will instruct the player to meet up with some mercenaries along the road to Falkreath so that together you can team up and put the dragon to the sword (stage 20).

The player travels to Ancient's Ascent where the dragon is roosting (stage 100) and confronts the dragon. But the dragon knows the player is dragonborn and catches the player off-guard with a shout (stage 110).

While the player is disabled by the shout, the dragon delivers his exposition - an invitation of sorts to the island of Wyrmsooth, then flies away. One of the mercenaries instructs the player to return to the Bannered Mare to let Lurius know what transpired (stage 120).

The player returns to the Bannered Mare and speaks to Lurius (stage 130).

Note: There's a gap in the quest stages between stage 20 and 110 because originally I had the player hunt down each mercenary individually.

So let's now look at setting up a quest in the Creation Kit.

For this section, I'll be recreating a simple side quest from Wyrmstooth called 'Retrieving Embersunder'. It's a simple fetch quest that should teach you the basics of setting up quests in the Creation Kit.

In the Object Window, go to Character > Quest.

Right-click on one of the existing quests and select New.

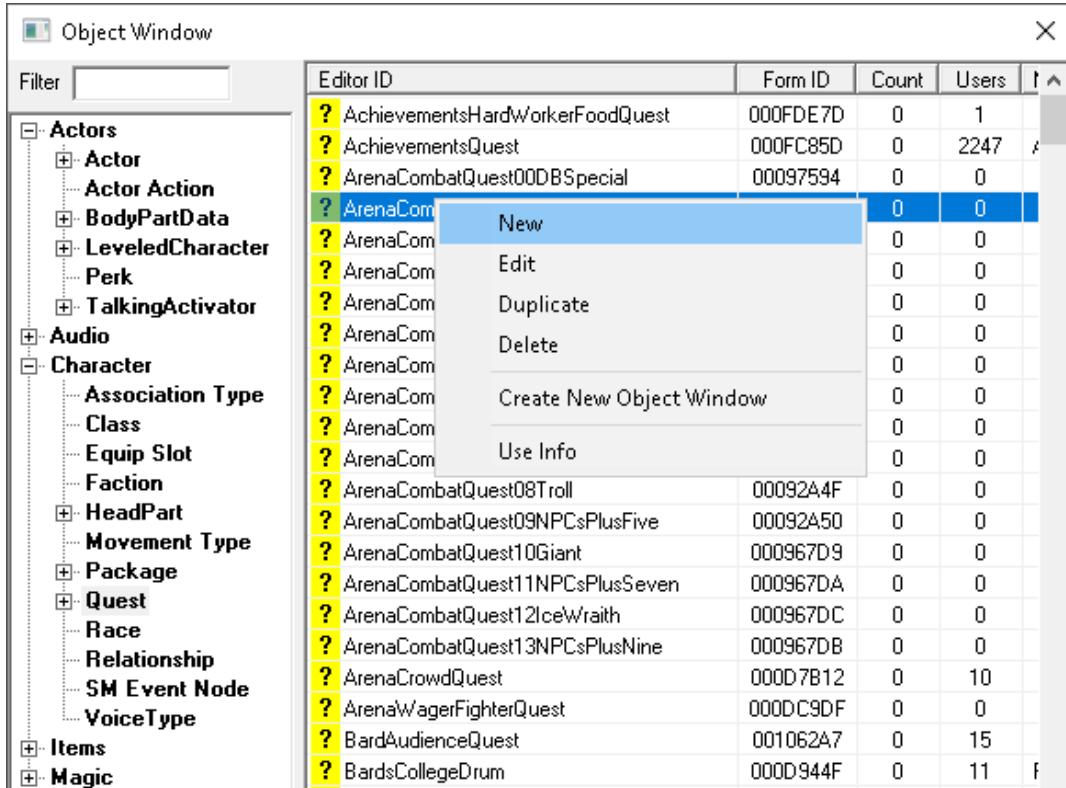


Figure 666 - Creating a new quest.

In the Quest Data tab, set the ID, Quest Name and Type. For my example, I just set the ID to WTTTestQuest, the Quest Name to Test Quest and the Type to Side-Quests.

I also set the Priority to 60. This determines dialogue precedence. Dialogue in the DialogueViews tab from quests with a higher priority will have precedence over dialogue from quests with a lower priority.

Leave ‘Start Game Enabled’ and ‘Run Once’ ticked.

Start Game Enabled means this quest will start up as soon as the game loads. This means any quest dialogue defined in that quest will become available as soon as the game loads.

Run Once means this quest can only be completed once. For repeatable radiant quests, you’re going to want to untick that.

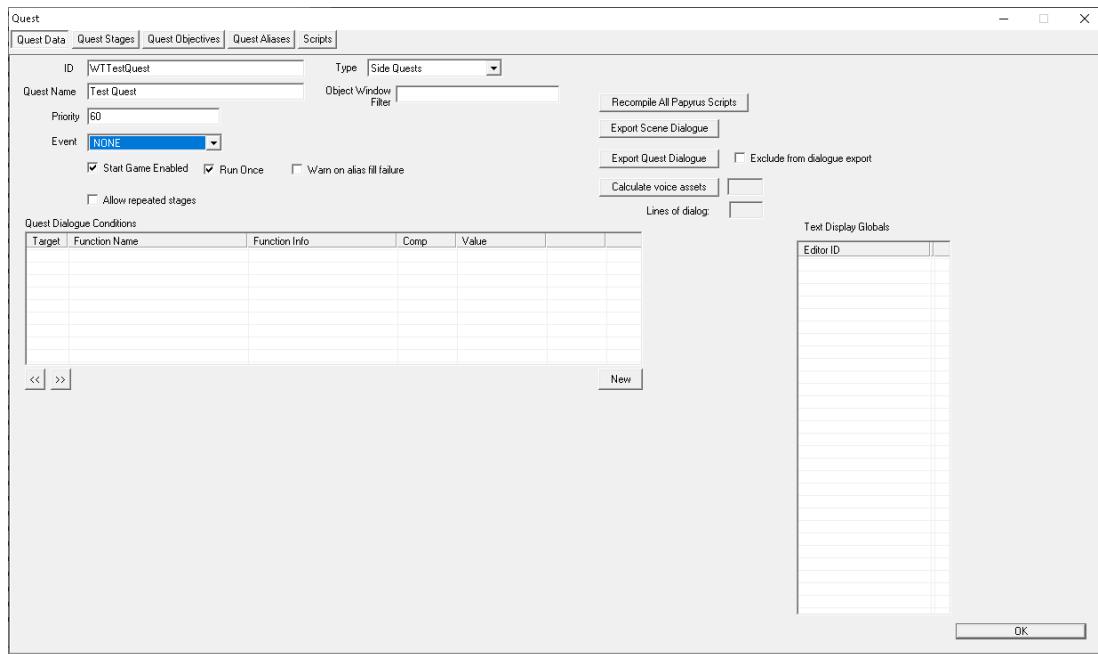


Figure 667 - Quest Data tab.

Important: Before doing anything else, click OK to close out of the Quest properties, then re-open the quest.

You should now see more tabs relating to quest dialogue.

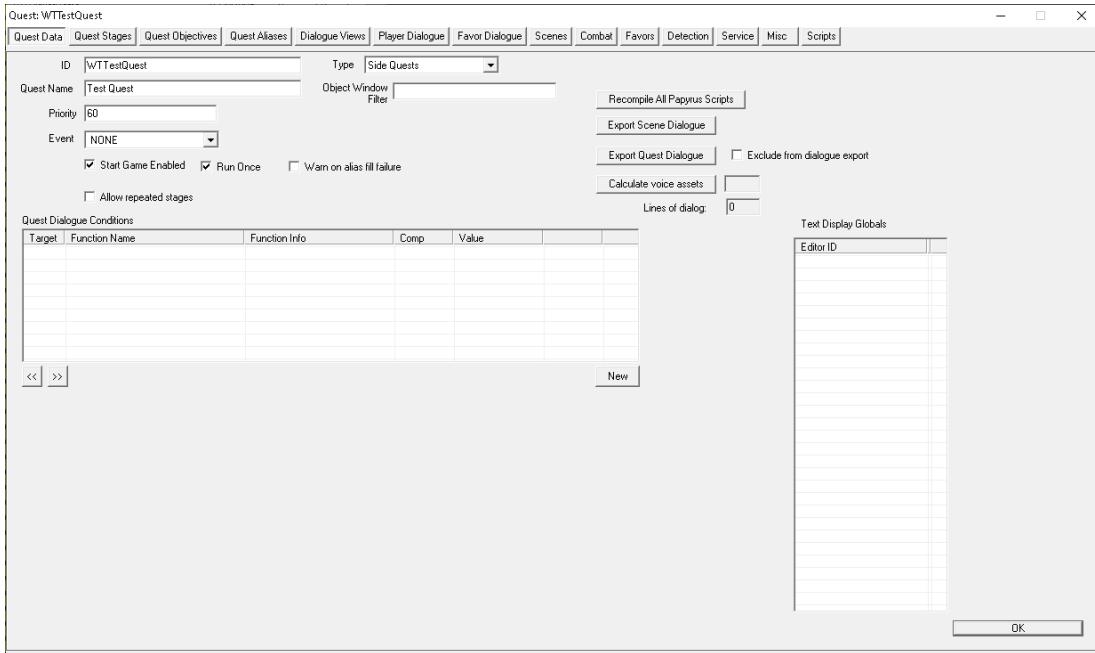


Figure 668 - More tabs in Quest properties.

Go to the Quest Stages tab. Right click on the quest stage list on the left hand side and select New to add a new stage.

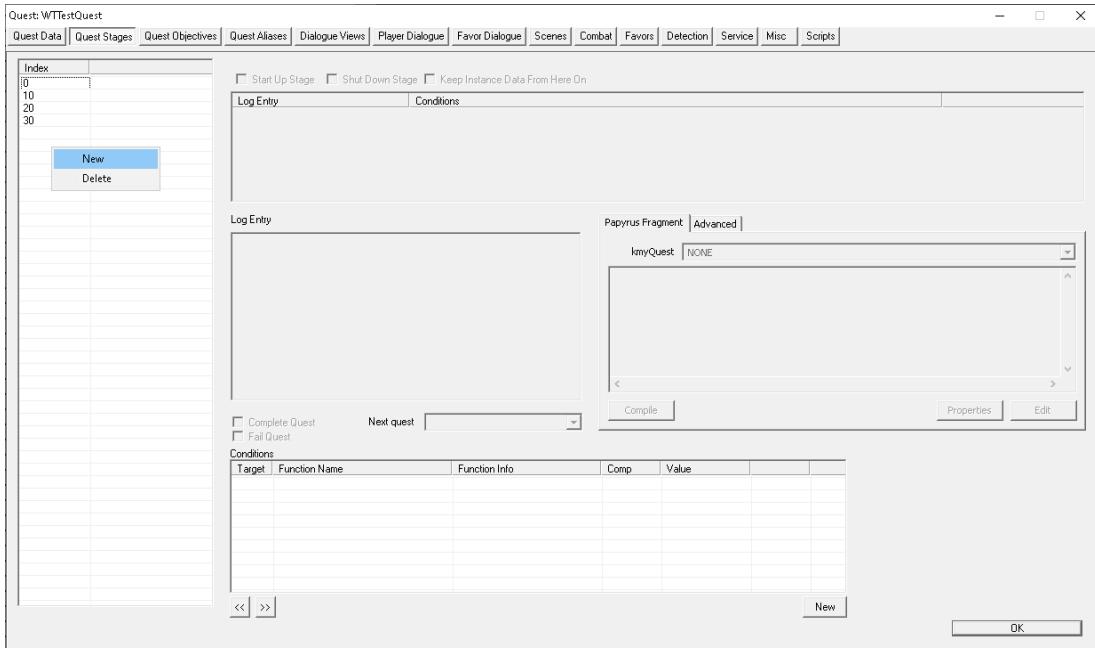


Figure 669 - Setting up quest stages.

I created four new stages: 0, 10, 20, and 30.

Right-click in the area under the ‘Start Up Stage’ tickbox and select New.

This will create a new log entry that will appear in your journal at that stage in the quest.

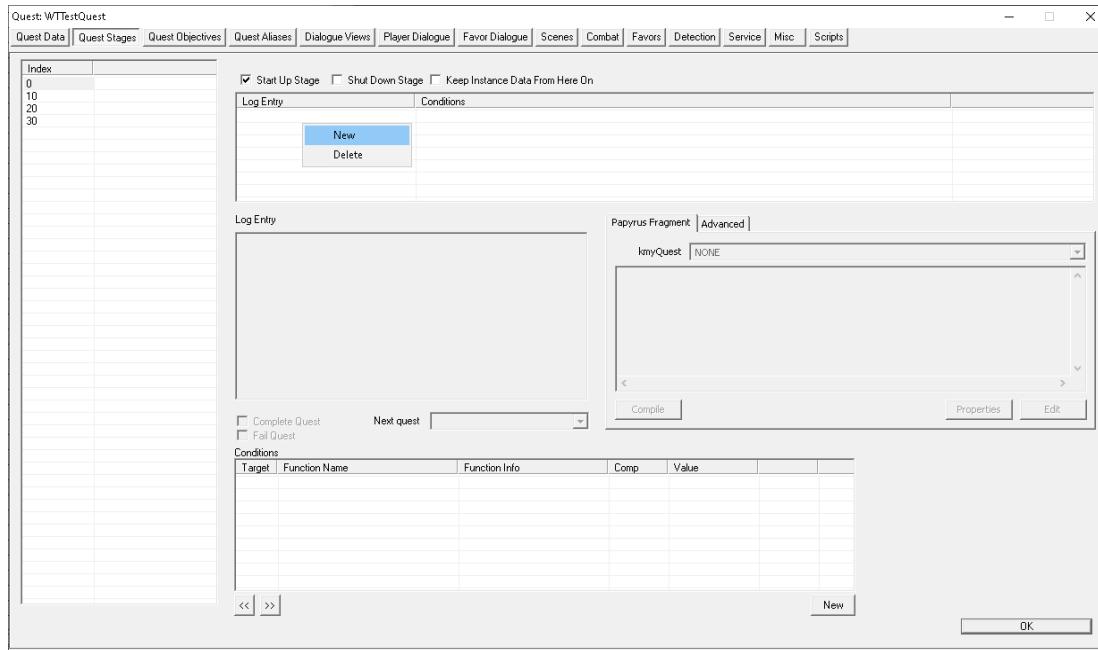


Figure 670 - Setting up the start-up stage.

It will also allow you to run papyrus on that stage.

Create a new log entry for stages 10, 20 and 30 as well.

We can leave the Log Entry fields blank for now.

Go to Stage 30.

Tick ‘Complete Quest’. Advancing to this stage will flag the quest as completed in the player’s journal.

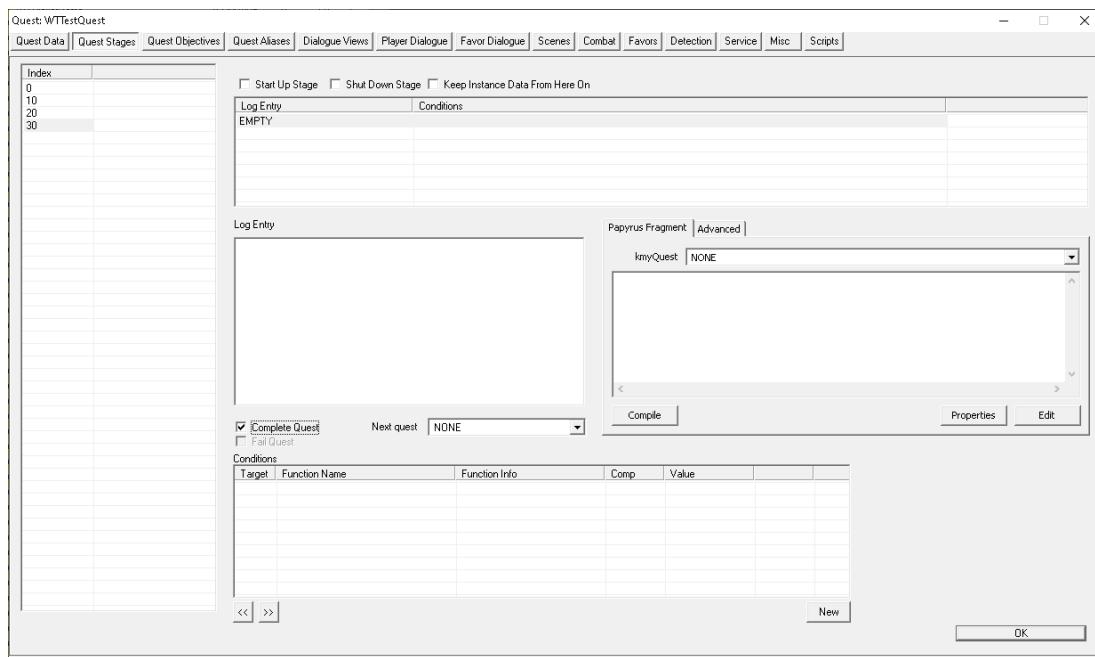


Figure 671 - Setting up the quest complete stage.

Go to the Quest Aliases tab.

Quest Aliases can be items, objects, NPCs or even locations that are involved in the quest.

For example, for a fetch quest, the item the player has to retrieve and the NPC giving the quest will both need to be added to the quest as reference aliases.

To add a reference alias, right-click in the aliases list and select New Reference Alias.

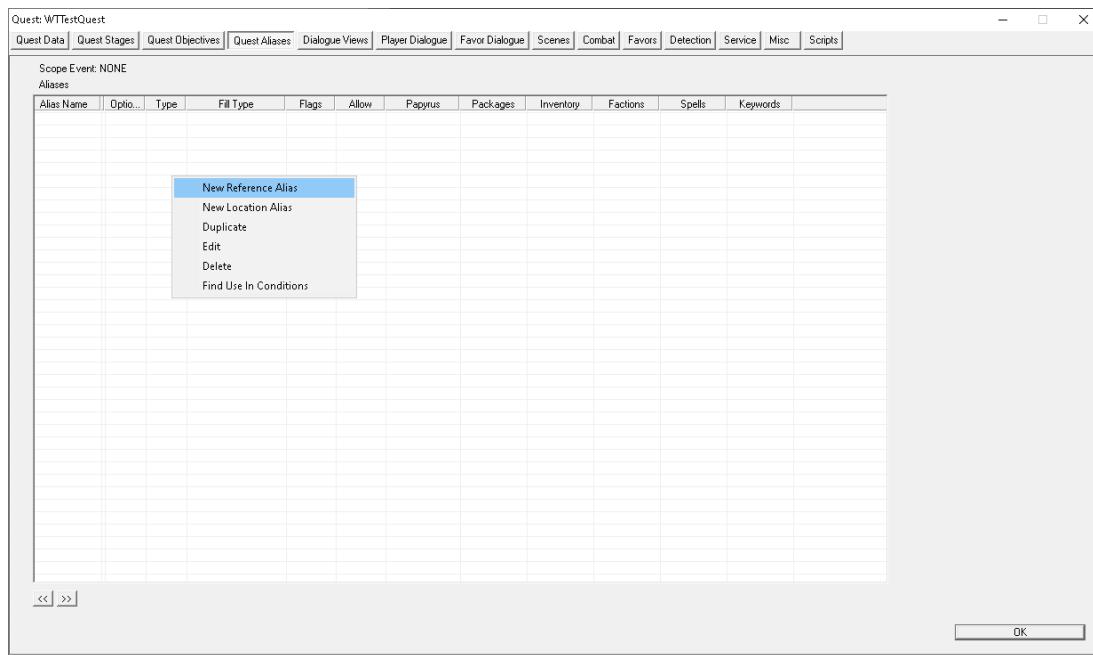


Figure 672 - Adding a new Reference Alias.

Enter in the name of the alias in the Alias Name field. For this example, I'm going to add in Athir, a unique NPC from Wyrmsooth.

Set the Fill Type to 'Unique Actor' then select the NPC from the drop-down to the right.

I also ticked the 'Essential' tickbox as I don't want this NPC to die while this quest is active.

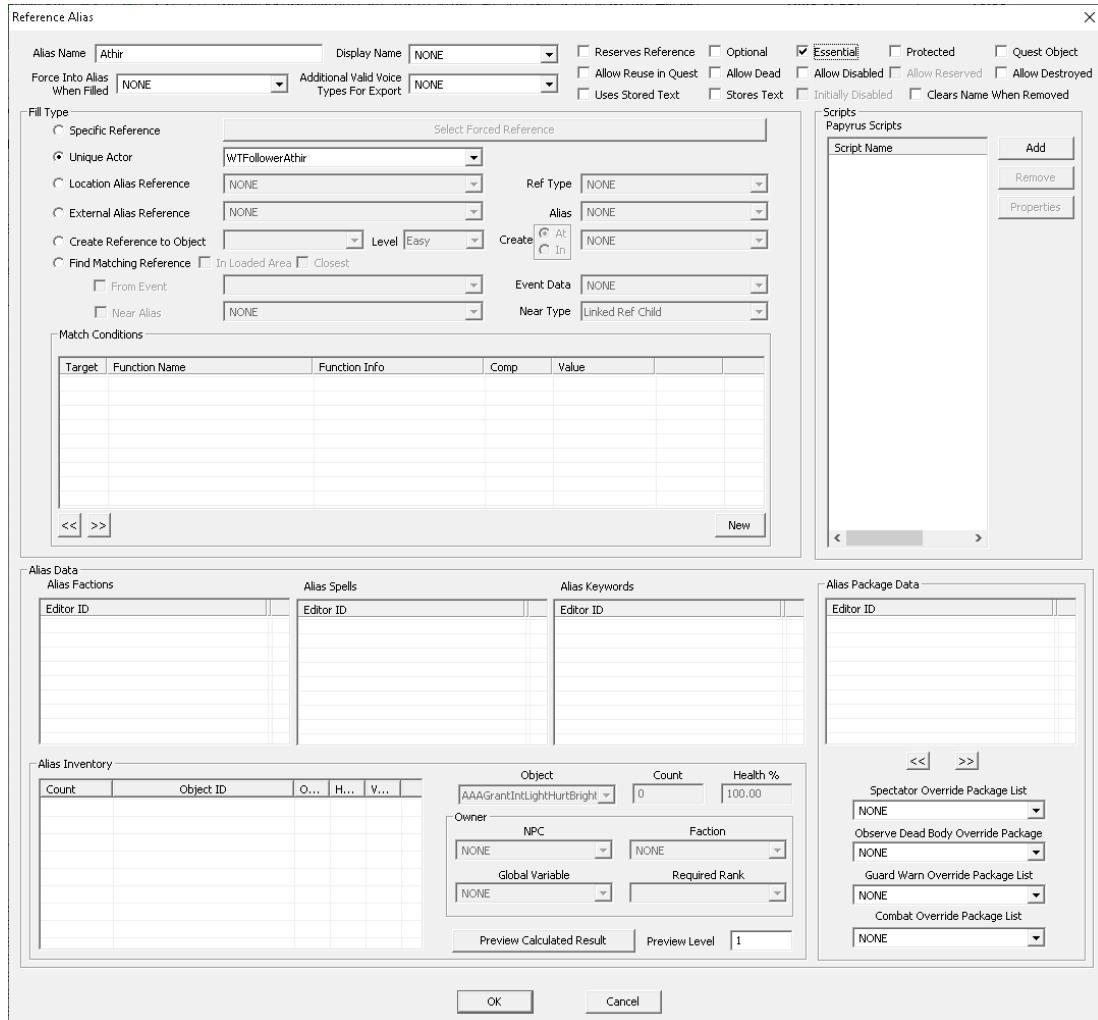


Figure 673 - Reference alias for Athir.

A 'unique' NPC is generally an NPC that has its own name, not something generic like 'bandit' or 'fox'.

Note: If you're selecting an NPC, use 'Unique Actor' where possible for the sake of compatibility. Selecting an NPC using Specific Reference will create an NPC edit, however selecting the same NPC through Unique Actor doesn't.

Next, let's select an item for the player to retrieve.

Right-click on the alias list again and select New Reference Alias.

Tick 'Quest Object' then set the Fill Type to 'Specific Reference'.

Quest Object flags this item as un-droppable.

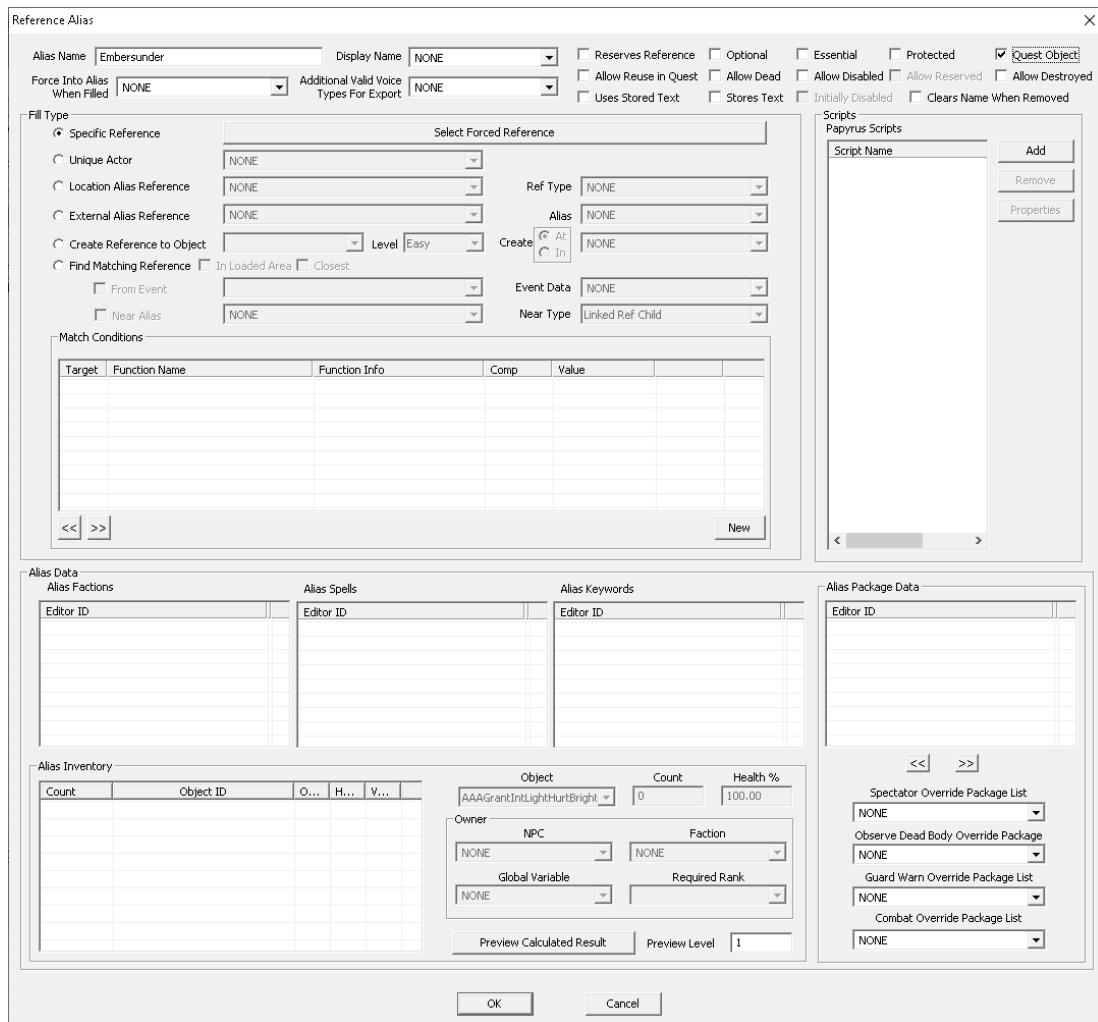


Figure 674 - Reference Alias for Embersunder.

To select the reference, click on the button next to it that currently reads 'Select Forced Reference'.

In the Choose Reference pop-up, click on ‘Select Reference in Render Window’

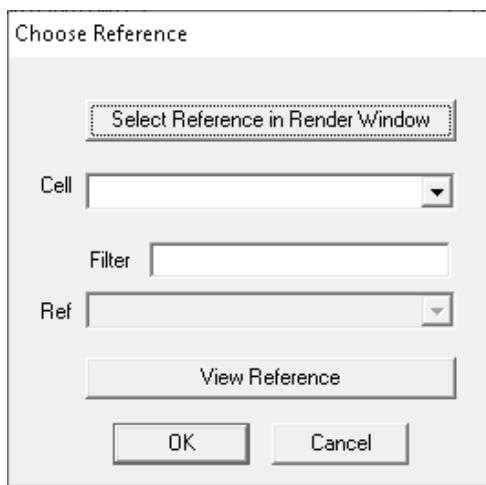


Figure 675 - Choose Reference pop-up.

This will change your mouse cursor to a crosshair in the render window.

The sword I want the player to retrieve currently exists in a special holding cell, which contains quest-related objects and NPCs related to Wyrmstooth that I want to have ‘off-screen’ until they’re ready.

When the quest starts up, I’ll be scripting the sword into place.

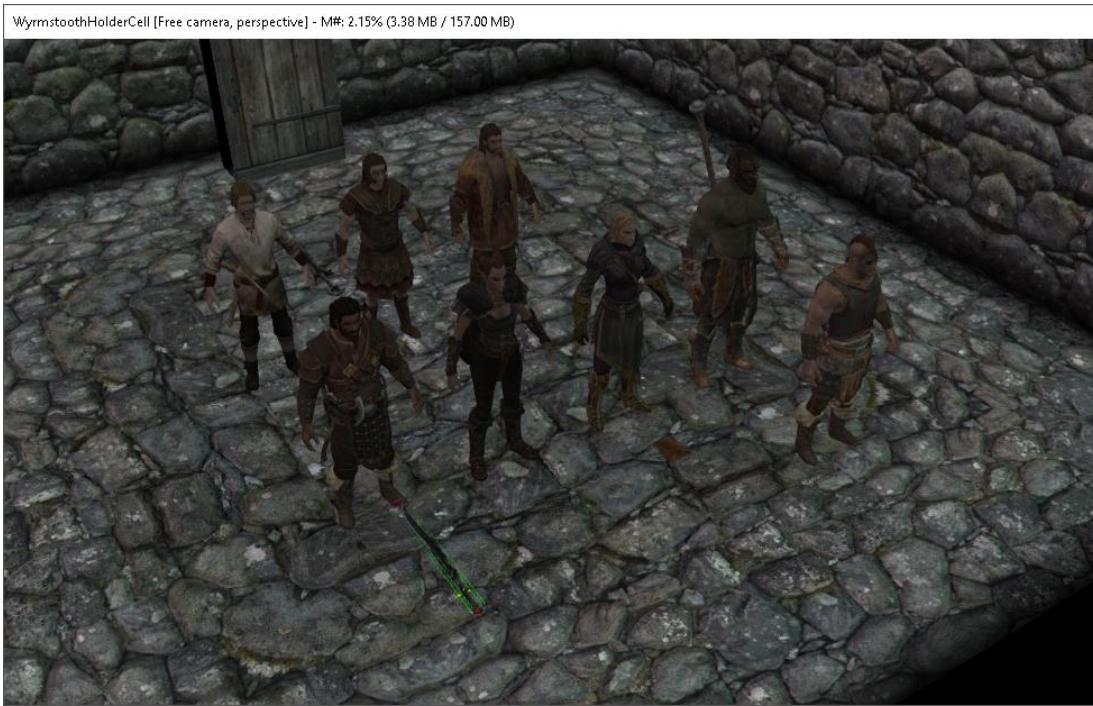


Figure 676 - A holding cell with quest-related NPCs and objects.

Double-click on the sword or the item you want the player to retrieve in the render window to select it.

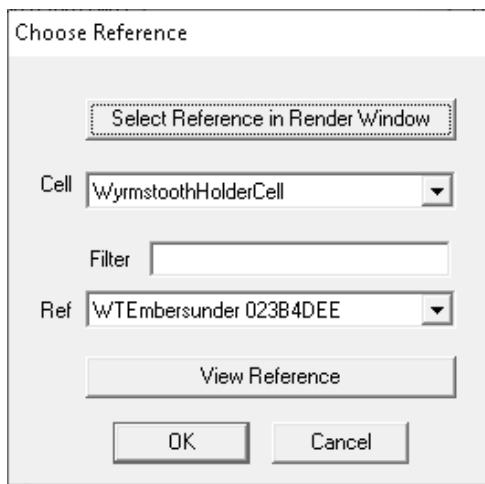


Figure 677 - Forced Reference.

Click OK to the Choose Reference pop-up.

Click OK to the Reference Alias properties.

Lastly, we need to define the location to move the sword to once the quest starts up.

Right-click in the alias list again and select New Reference Alias.

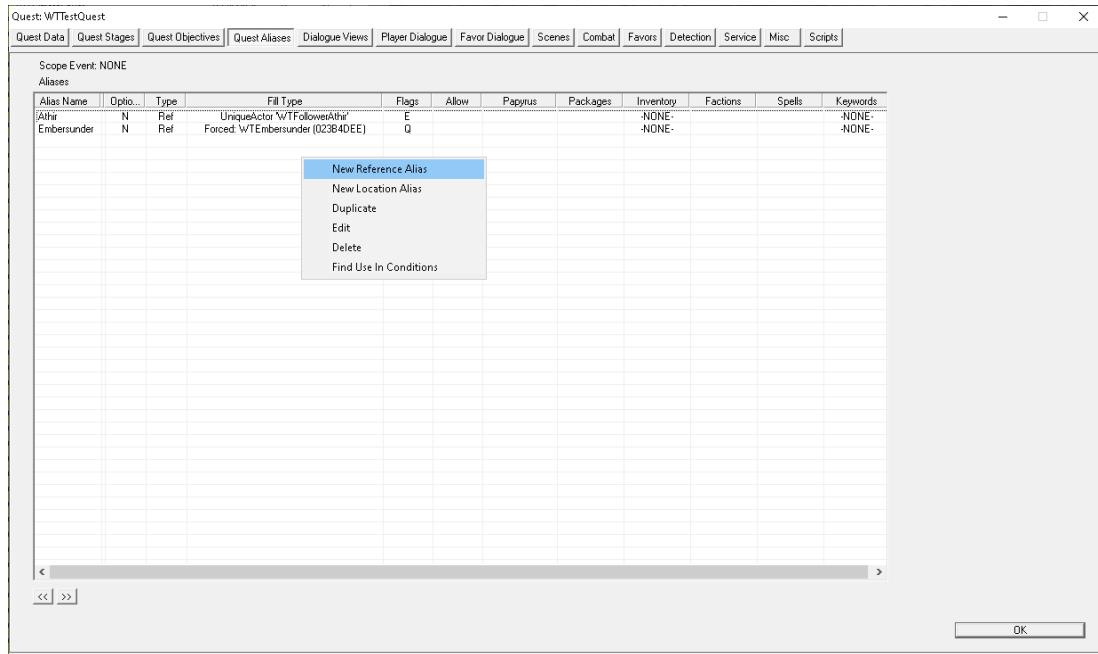


Figure 678 - Adding a third reference alias.

Set the Alias Name to BossLocation.

Set the Fill Type to Specific Reference then click on ‘Select Forced Reference’.

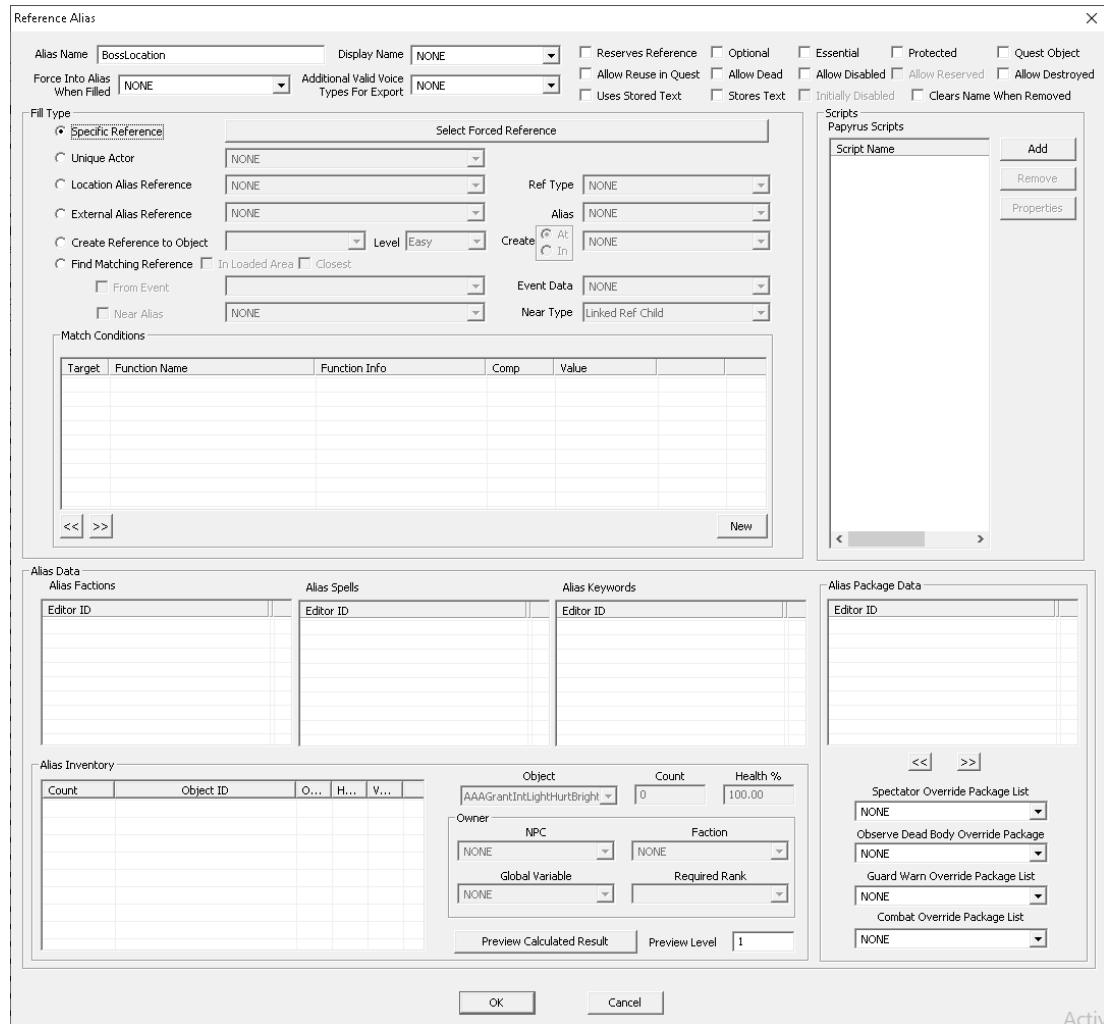


Figure 679 - Setting the BossLocation reference alias.

On the Choose Reference pop-up, click on ‘Select Reference in Render Window’.

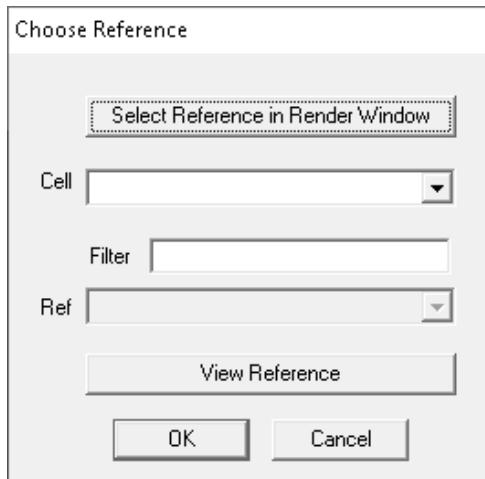


Figure 680 - Selecting the reference alias from the render window.

For this example, I'm going to select the BossTreasureMarker next to the boss chest at Hag's End. This is in the HagsEndOrigin cell of the DeepwoodRedoubtWorld world space.



Figure 681 - Selecting the location to move the sword to.

Click OK to the Choose Reference pop-up.

Click OK to the Reference Alias properties.

We should now have three quest aliases as per the screenshot below.

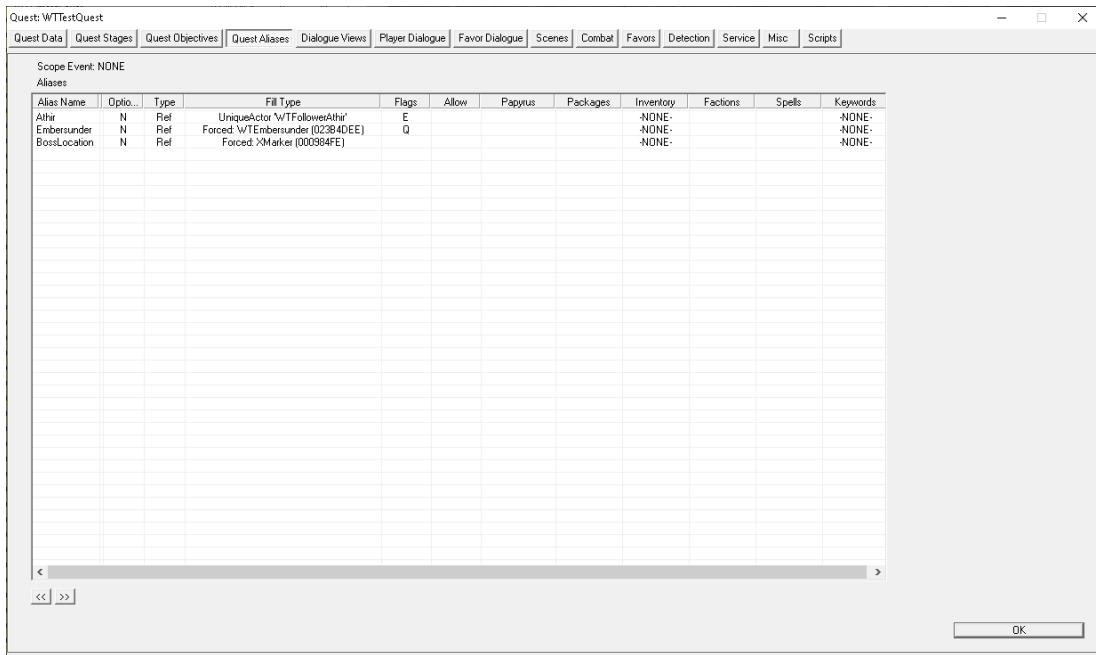


Figure 682 - Our quest aliases added.

Go to the Quest Objectives tab.

Right-click in the objectives list and select New.

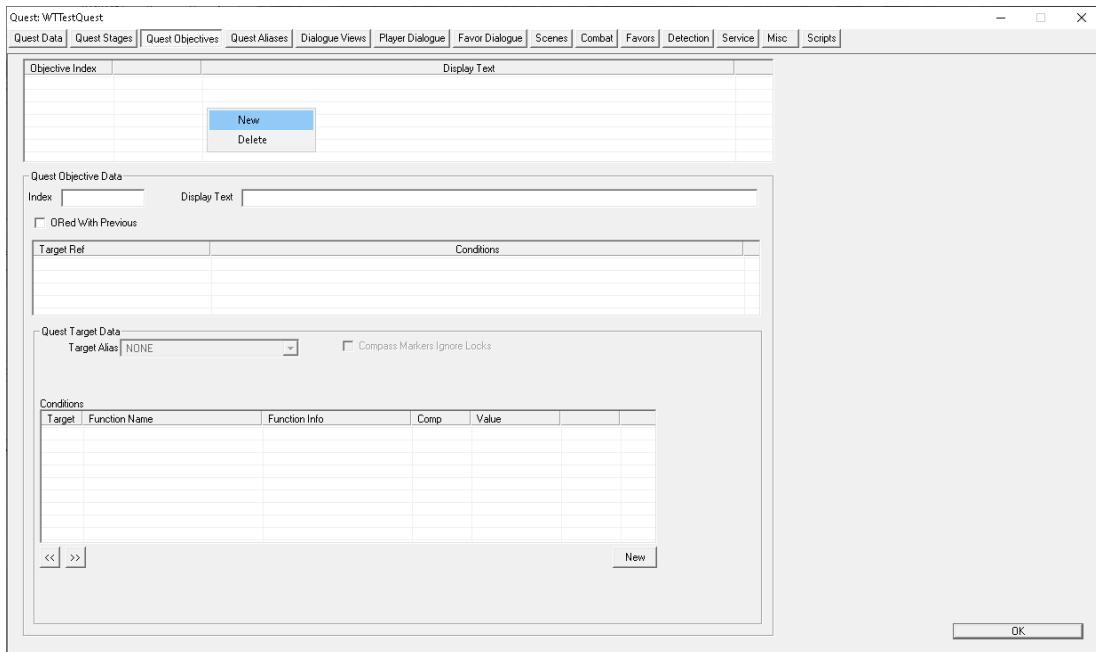


Figure 683 - Adding a new quest objective.

Set the Index to '10' and the Display Text to 'Retrieve Embersunder'.

This will be the objective for stage 10 of this quest. The display text is the objective that appears in the player's journal.

Right-click on the target list below and select New.

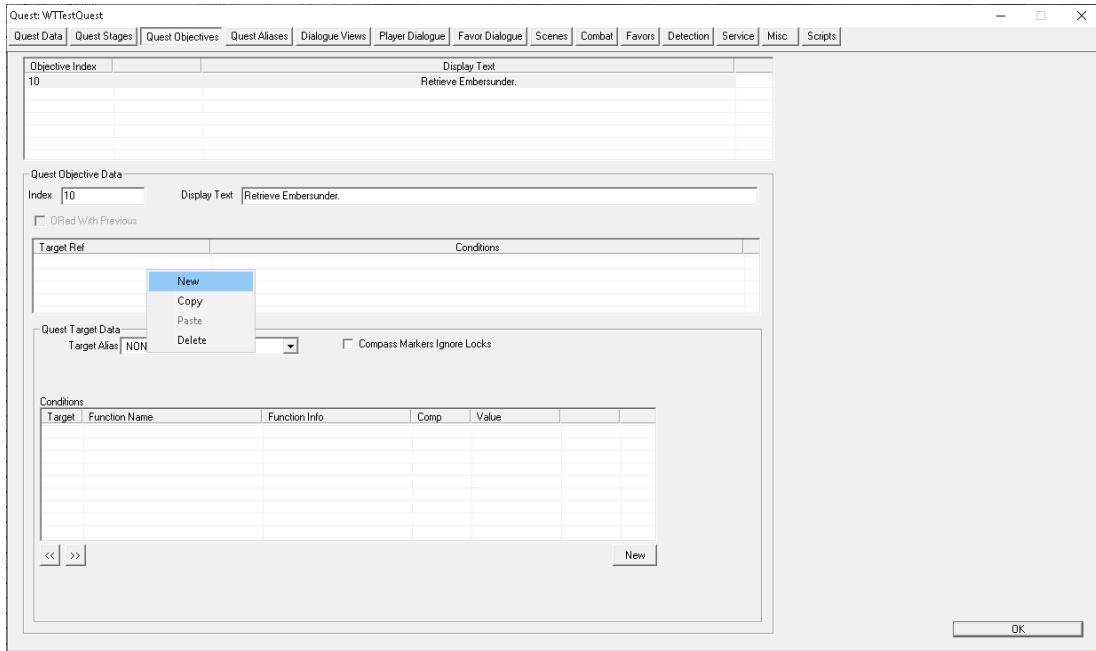


Figure 684 - Adding a quest objective.

Under Quest Target Data, set the Target Alias to Embersunder.

The reference selected here will have an objective pointer above it in-game at that stage of the quest.

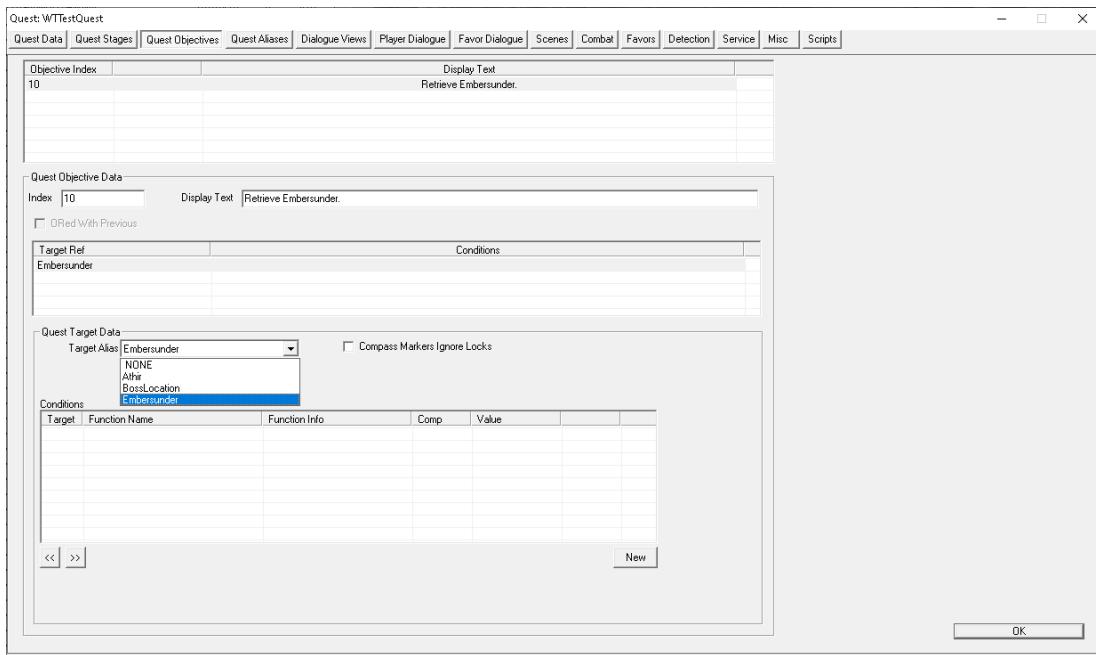


Figure 685 - Setting the quest target.

Create a second objective for stage 20. Set the Display Text to 'Return to Athir.'

Set the Target Ref to Athir.

This will place an objective over Athir once the player has obtained the sword.

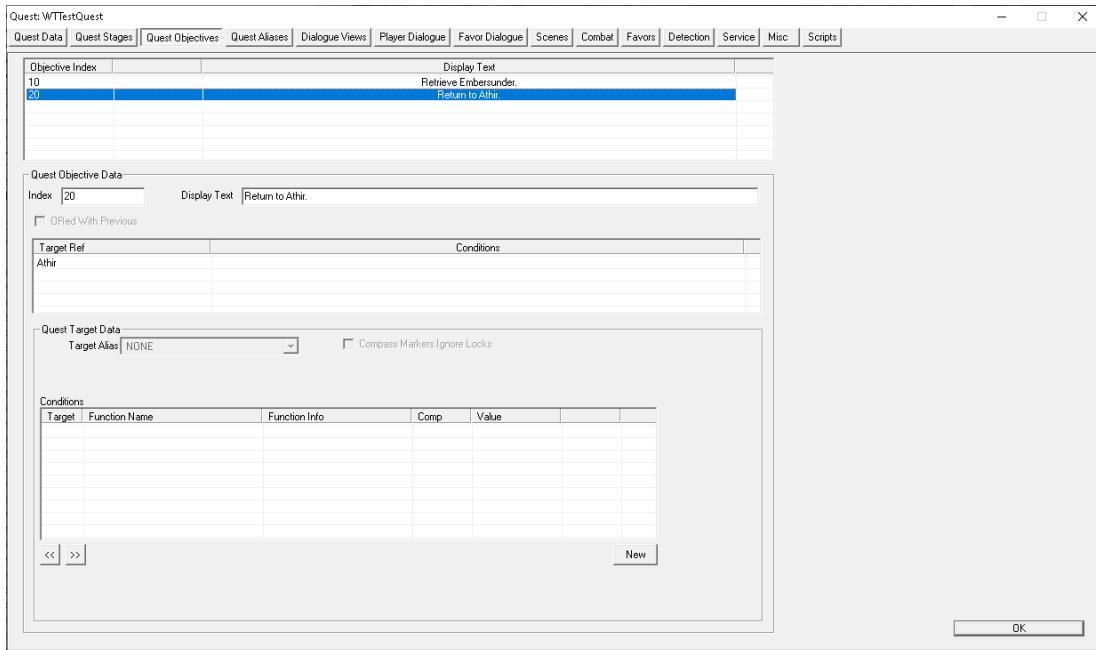


Figure 686 - Setting the second objective.

Go back to the Quest Stages tab.

Now we can enter a Log Entry for stage 10. I just went with:

Athir has asked me to find a sword called Embersunder. I should search the forswn camp at Hag's End as Athir suggested.

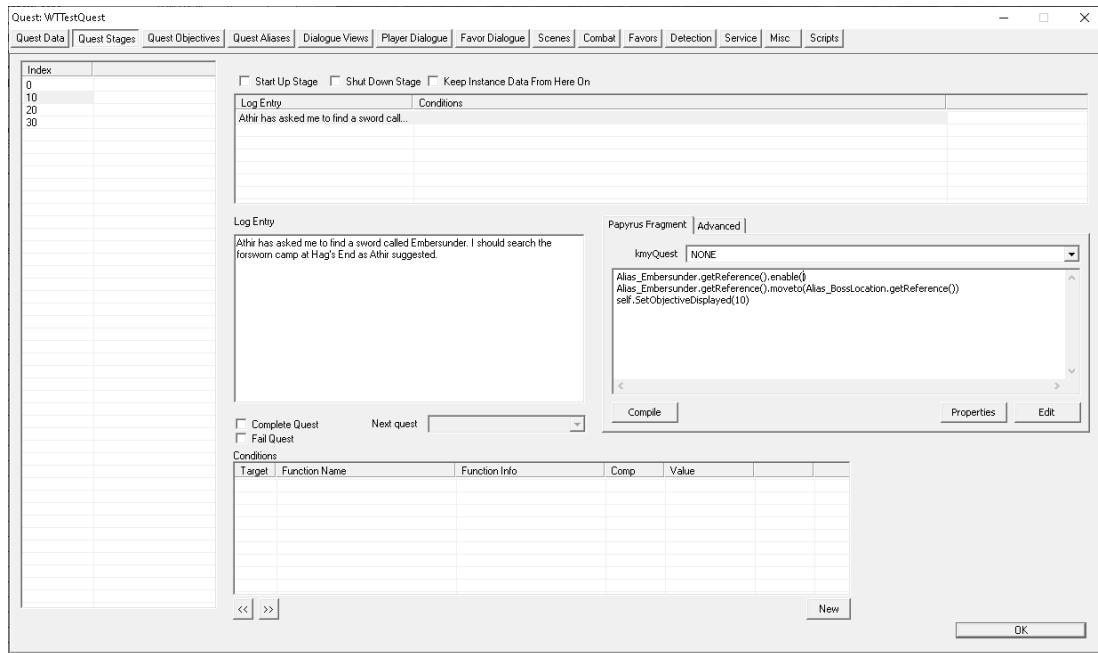


Figure 687 - Stage 10 of the sample fetch quest.

Under the Papyrus Fragment tab, enter in the following code:

```

Alias_Embesunder.getReference().enable()
Alias_Embesunder.getReference().moveto(Alias_BossLocation.getReference())
self.SetObjectiveDisplayed(10)

```

The sword is flagged as disabled by default so we need to enable it first.

It's then moved to the BossMarker reference alias and the stage 10 quest objective is displayed in the journal with a quest marker pointing to the sword.

For stage 20, I set the Log Entry to:

I have found the sword Embersunder for Athir. I should bring it to him when I get the chance.

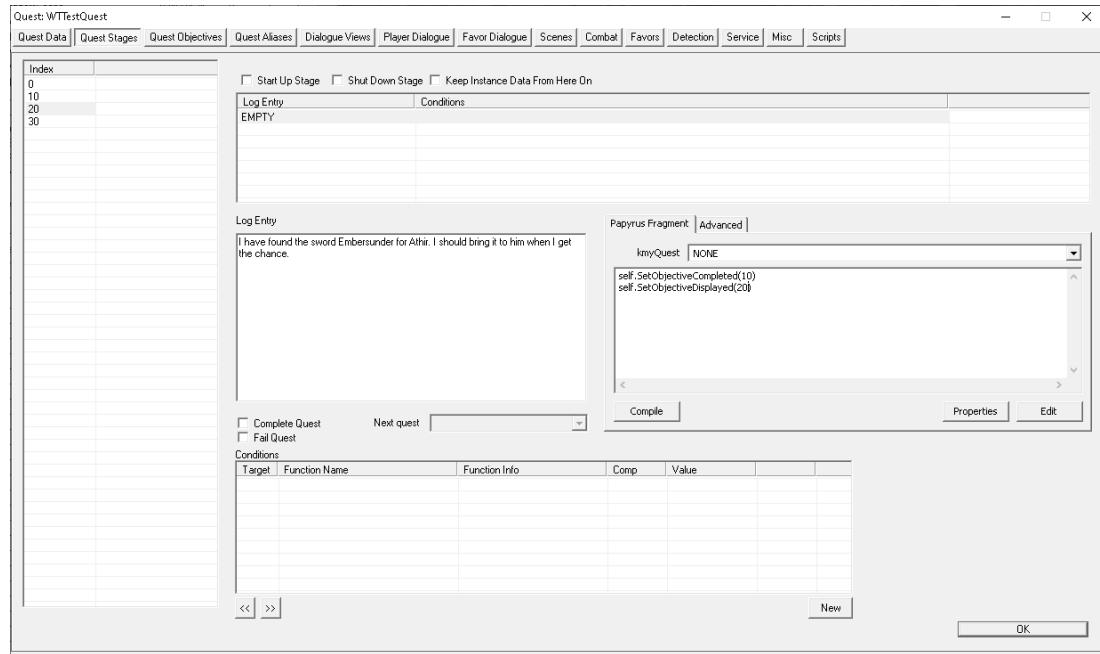


Figure 688 - Stage 20 of the sample fetch quest.

Under the Papyrus Fragment tab, enter in the following code:

```
self.SetObjectiveCompleted(10)
self.SetObjectiveDisplayed(20)
```

Once the player successfully retrieves the sword, we need to clear the objective to obtain the sword and replace it with the objective to return to our quest-giver Athir.

Finally, for stage 30, I set the Log Entry to:

I have given the sword Embersunder to Athir.

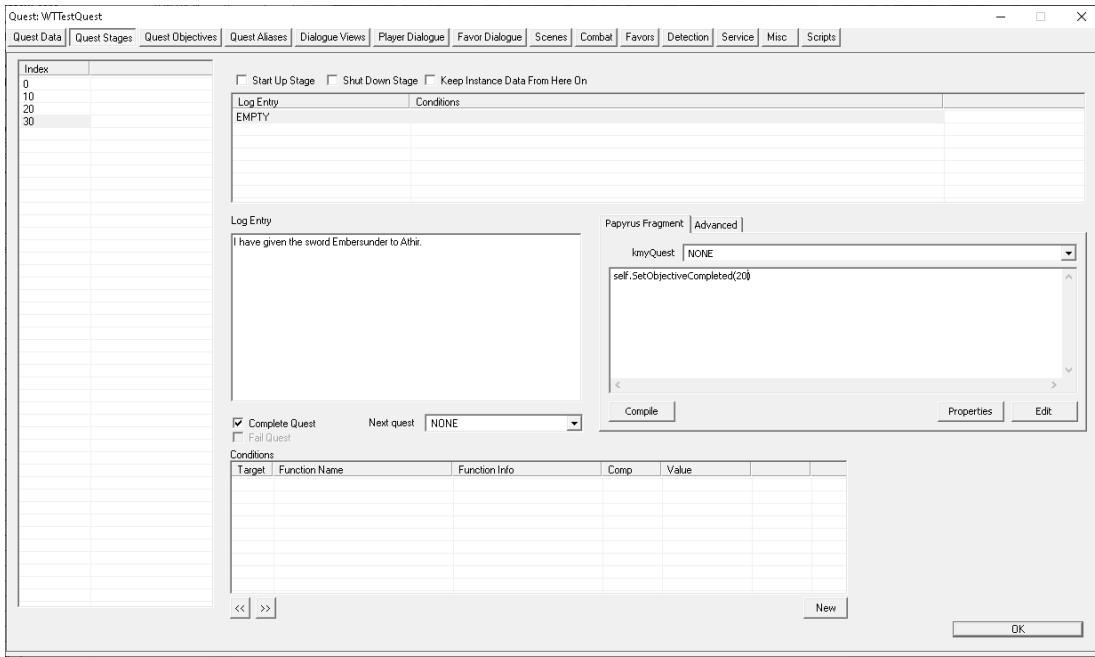


Figure 689 - Stage 30 of the sample fetch quest.

Under the Papyrus Fragment tab, enter in the following code:

```
self.SetObjectiveCompleted(20)
```

Now that we've given the sword to Athir we can complete the last objective and shut down the quest.

You can use the `stop()` command to manually shut down the quest or tick 'Shut Down Stage' on stage 30.

Important: Quest dialogue will become unavailable once the quest shuts down.

Lastly, we need some way of detecting when the player has acquired the sword. To do this, double-click or right-click on the sword in the render view and select Edit.

Go to the Scripts tab and select Add.

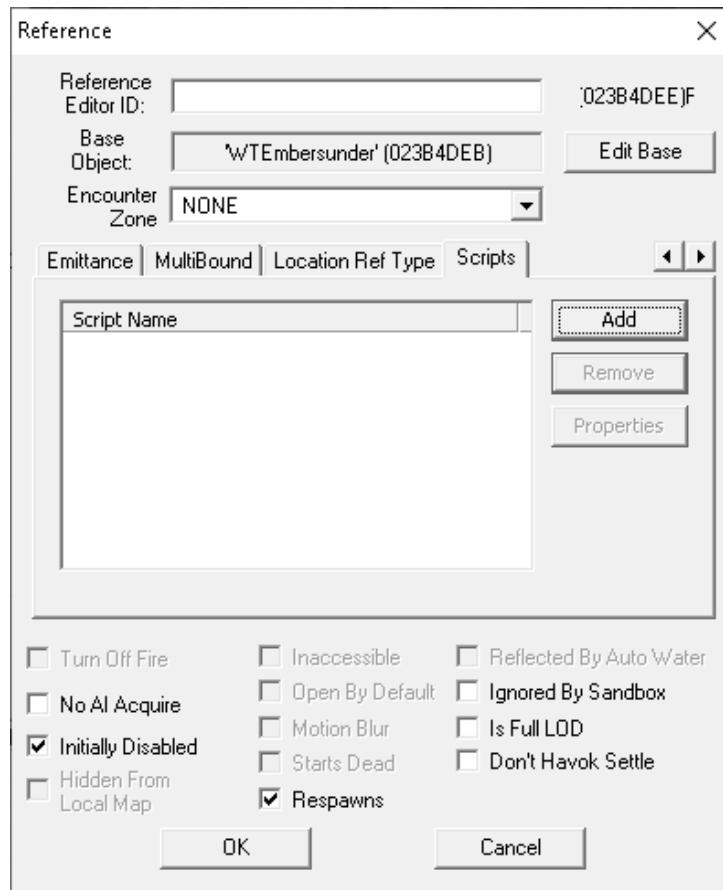


Figure 690 - Scripts tab.

Filter by ‘defaultsetstageonplayeracquireitem’.

Highlight that script in the script list and click OK.

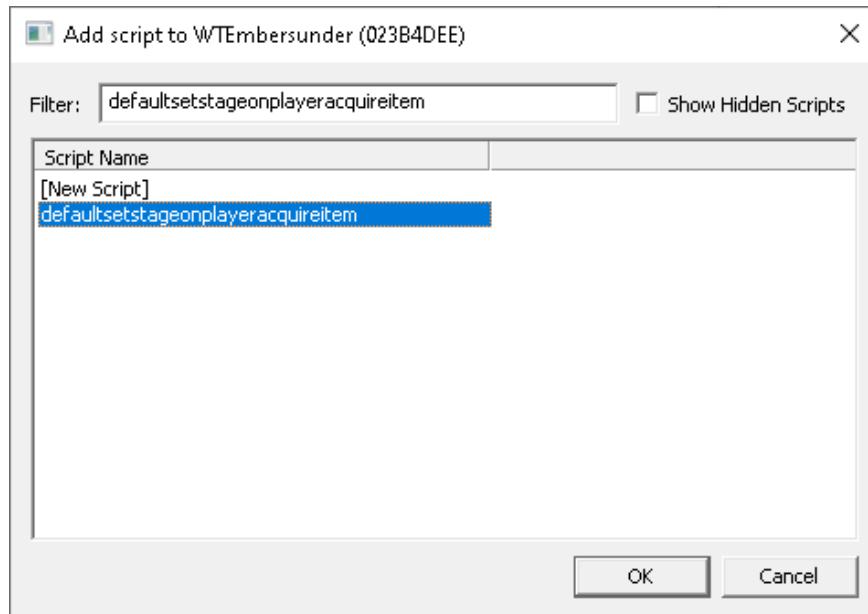


Figure 691 - Add script pop-up.

Highlight 'defaultsetstageonplayeracquireitem' in the Scripts tab then click Properties.

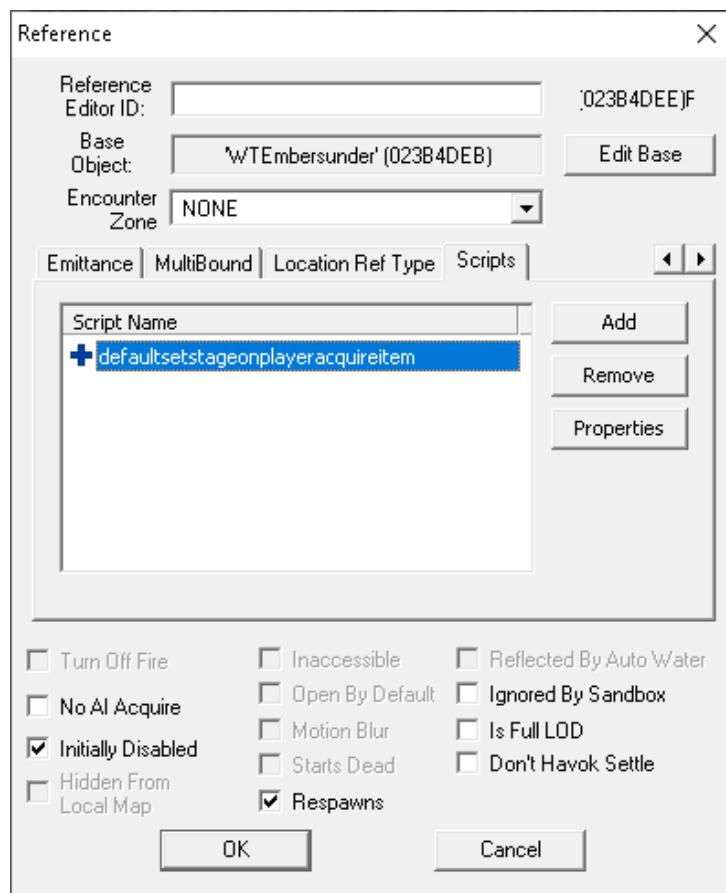


Figure 692 - Script added to the object reference.

Highlight myQST and set the quest value from the Pick Object drop-down. For this example, that's going to be WTTTestQuest.

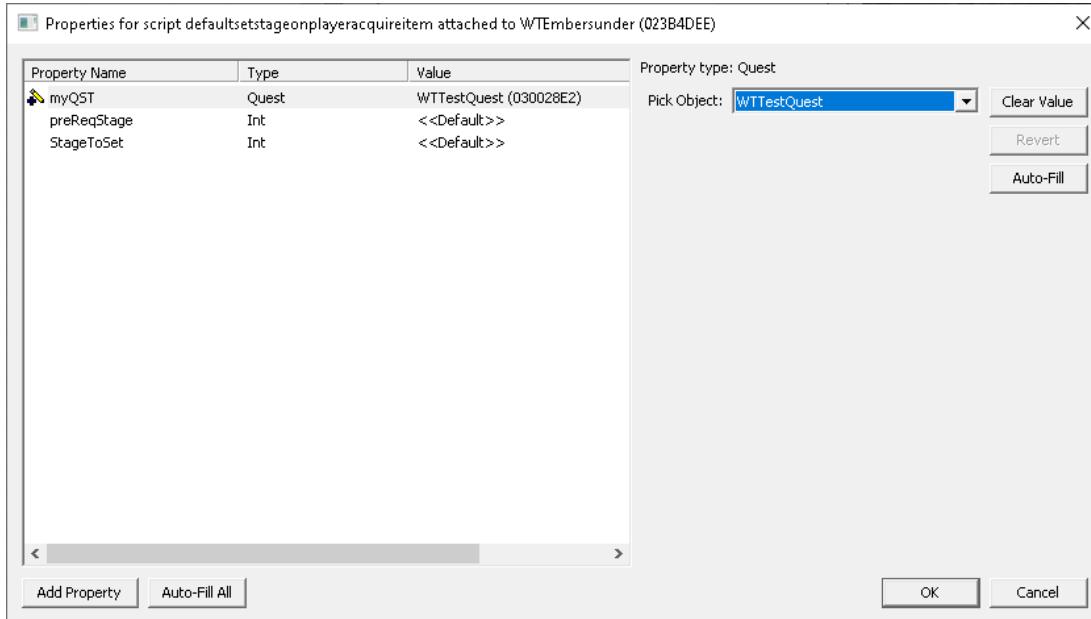


Figure 693 - Setting the *myQST* variable value.

Set preReqStage and click on Edit Value. Set preReqStage to 10.

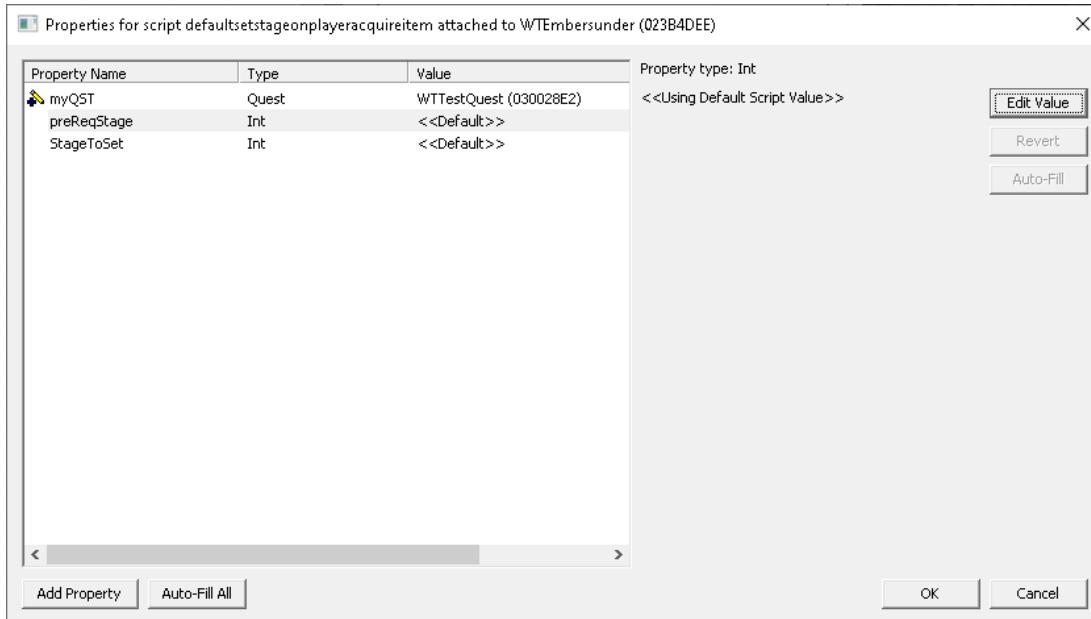


Figure 694 - Setting the prerequisite quest stage.

Now go to StageToSet and set it to 20.

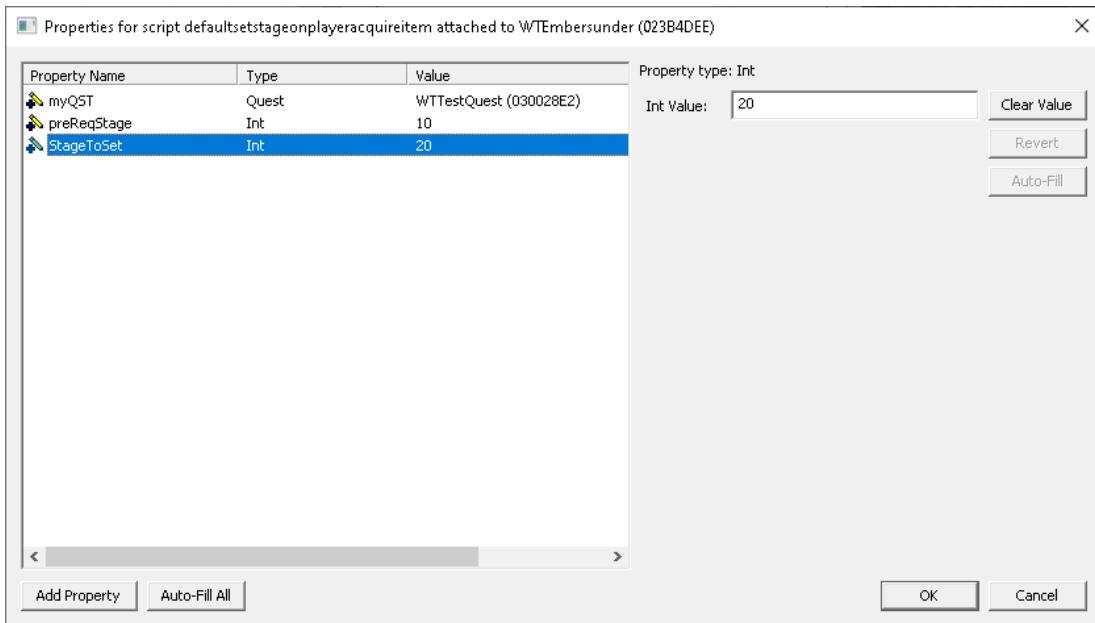


Figure 695 - Setting StageToSet to 20.

Click OK to close the script properties.

Click OK to close the Reference properties.

When the player picks up this item it'll set the quest stage to 20. The papyrus on stage 20 changes the objective to 'Return to Athir' and the quest marker will be set to Athir's current position.

The papyrus for setting stage 10 and stage 30 of this sample quest will be added to the dialogue itself which I'll be covering in the next section.

ADDING DIALOGUE TO A QUEST

All dialogue in the game is attached to quests. Some quests, such as DialogueGeneric, serve the singular purpose of providing NPCs with their generic hello, goodbye, combat and trespass dialogue.

Note: For the sake of compatibility, I'd recommend adding new dialogue to your own quests where possible rather than modify existing quests.

In this section, I'll be continuing where I left off in the previous section and will show you how to add in dialogue for our sample fetch quest.

Back in our quest properties, go to the Dialogue Views tab. Right-click in the Dialog Views list and select New.

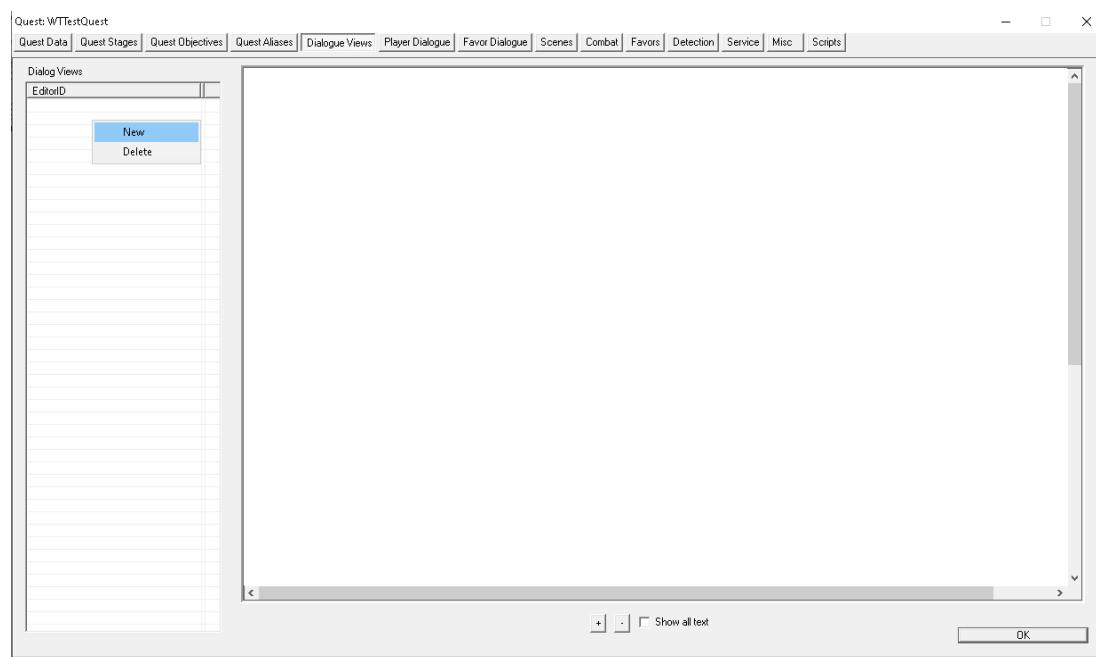


Figure 696 - Adding a new dialogue view.

Give the dialogue view a unique ID. For this example, I'm going to call it WTTTestQuestAthirDialogView.



Figure 697 - Giving our dialogue view a unique ID.

Click OK.

Highlight the new dialogue view. Right-click in the area to the right and select ‘Create branch’.

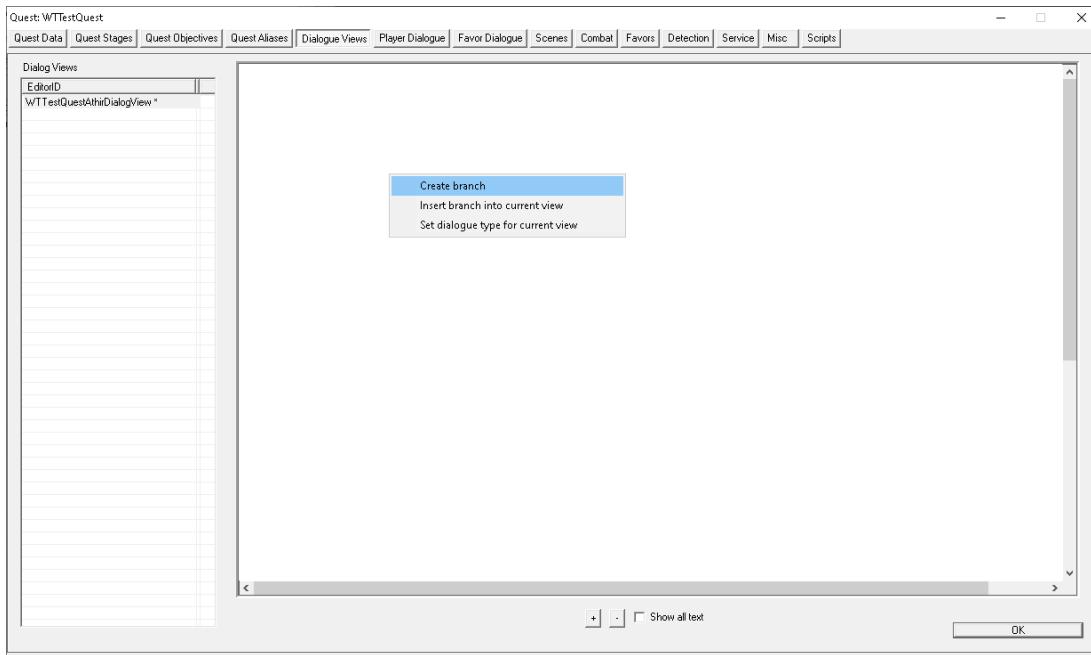


Figure 698 - Creating a new branch.

Enter in a branch ID and click OK. I'm going to call it WTTTestQuestAthirBranch1.

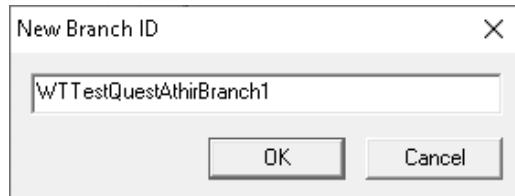


Figure 699 - Adding a branch ID.

You'll also be prompted to enter in the ID for the first topic. Generally this should include the branch ID. In the following example I went with WTTTestQuestAthirBranch1Topic1.

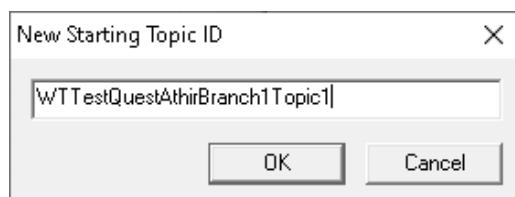


Figure 700 - Adding the first topic ID.

The new branch and topic should appear as per the screenshot below.

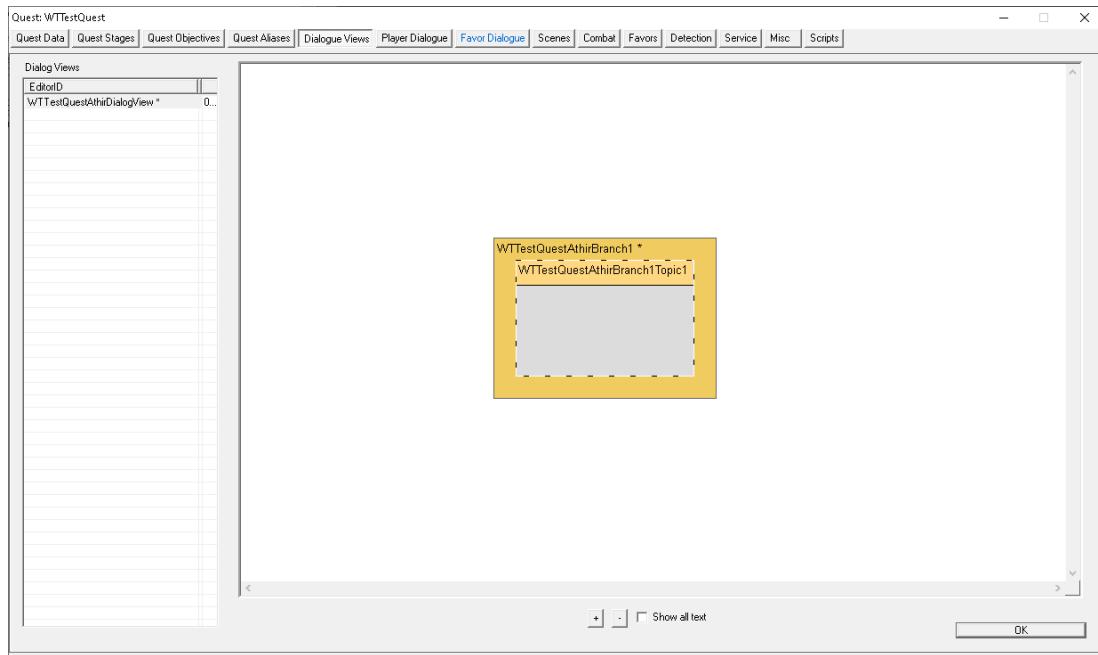


Figure 701 - New branch and topic.

Double-click on the topic within the branch to open it.

Enter in the Topic Text. The topic text is a player's dialogue option that is available when speaking to an NPC.

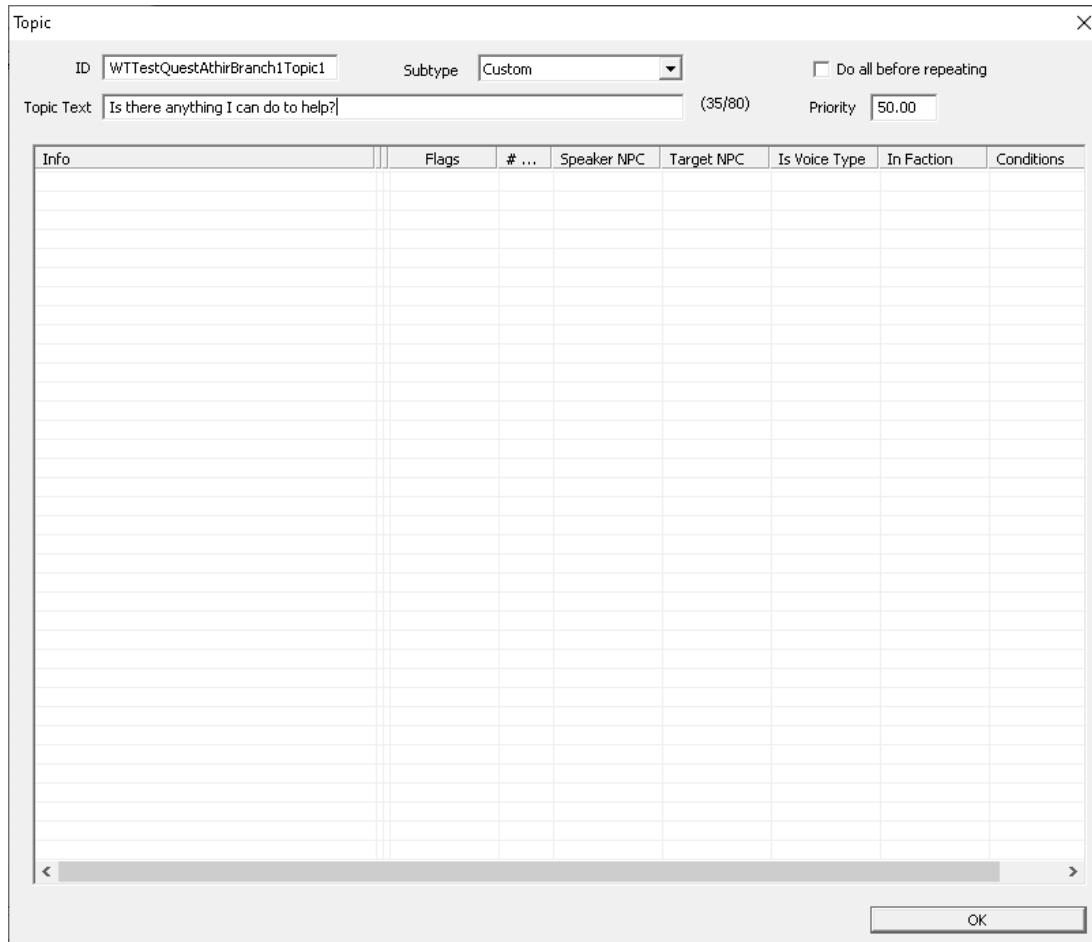


Figure 702 - Adding topic text.

Right-click in the space beneath the topic text and select New to add a new NPC response.

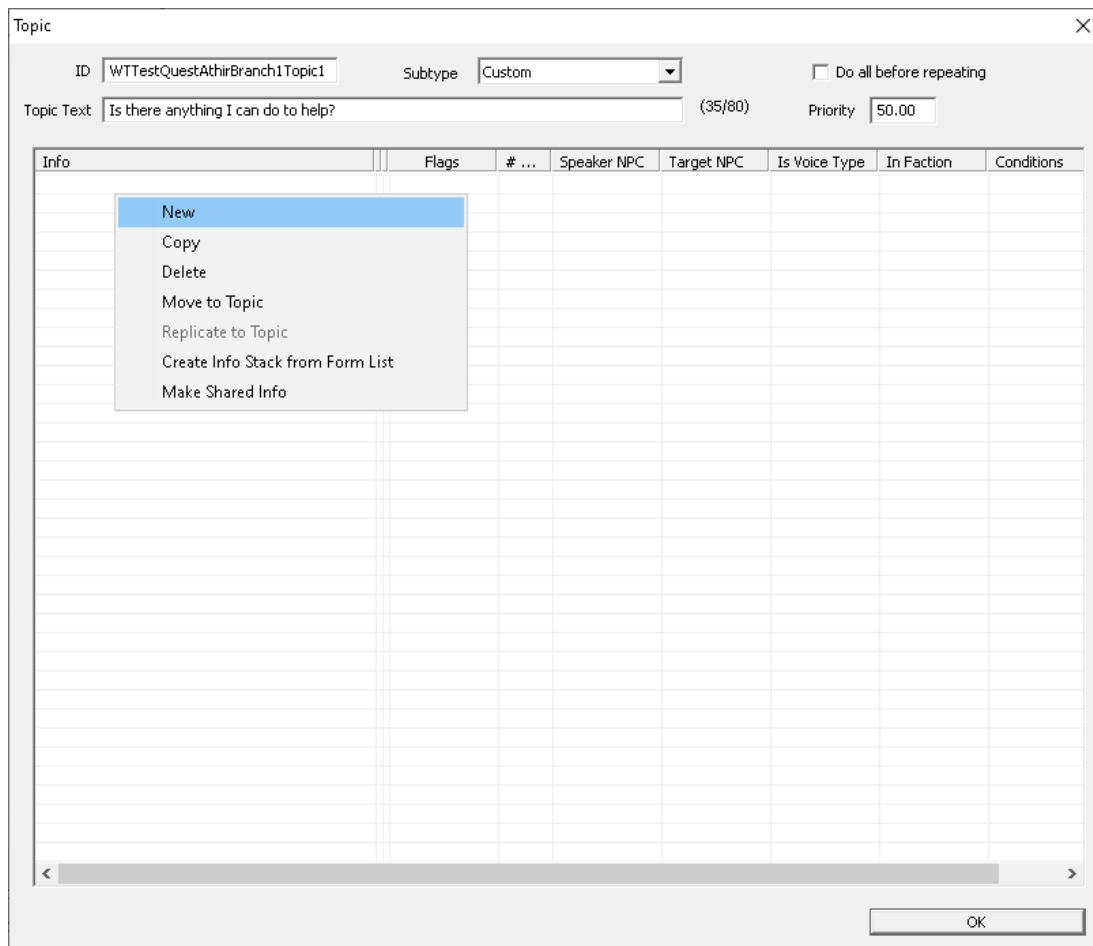


Figure 703 - Adding a new NPC response.

Enter in the NPC's response in the Response Text field. This will be used as the subtitle for the line of the dialogue that the NPC will say.

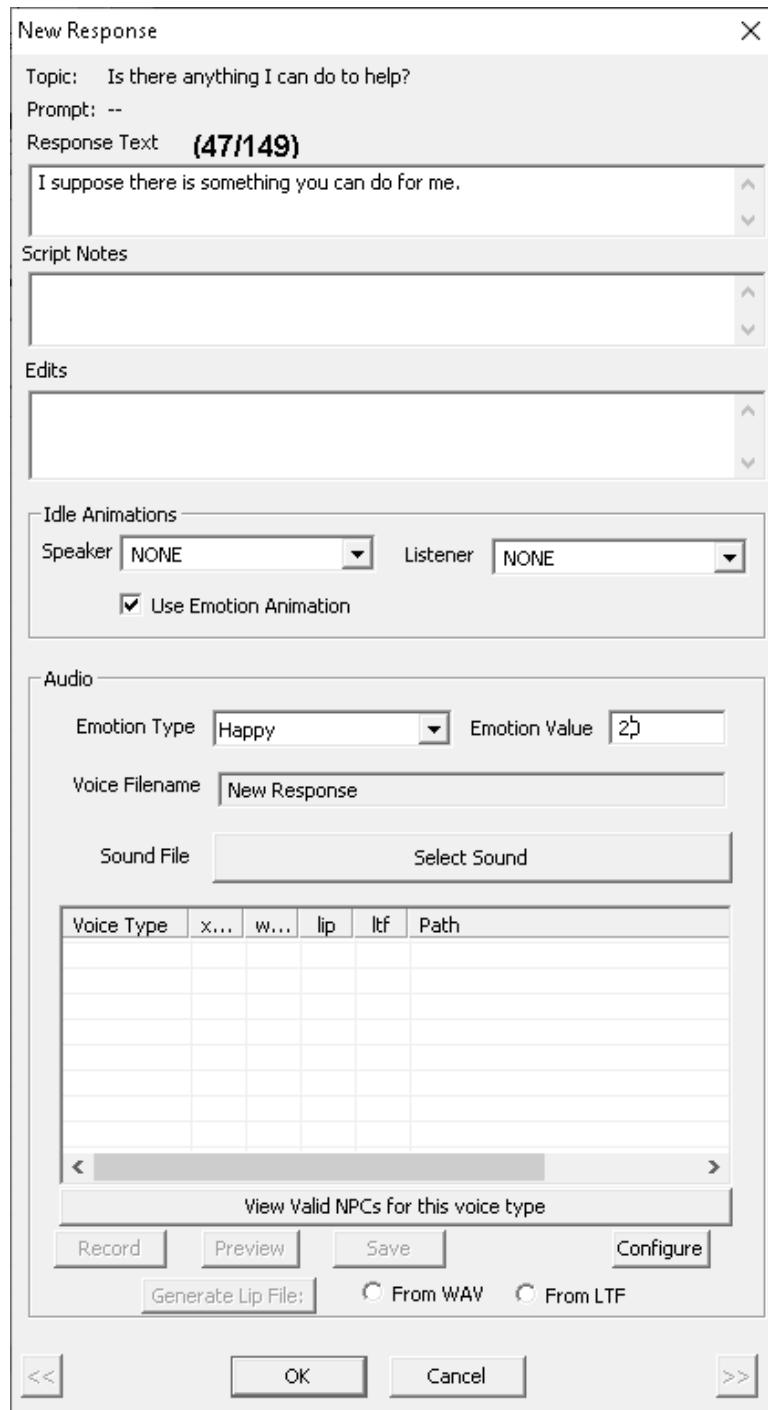


Figure 704 - Configuring the NPC's response.

If you want the NPC to play an animation while speaking, select the animation in the Speaker drop-down under Idle Animations.

The Emotion Type drop-down can be used to influence an NPC's facial animations while they're speaking a line of dialogue. In the previous example, I set the Emotion Type to Happy. You can control the amount of emotion they show in the Emotion Value field. The lower the number, the closer to neutral their expression will be.

Click OK to close the New Response screen.

Our Topic Info screen should now appear as per the screenshot below:

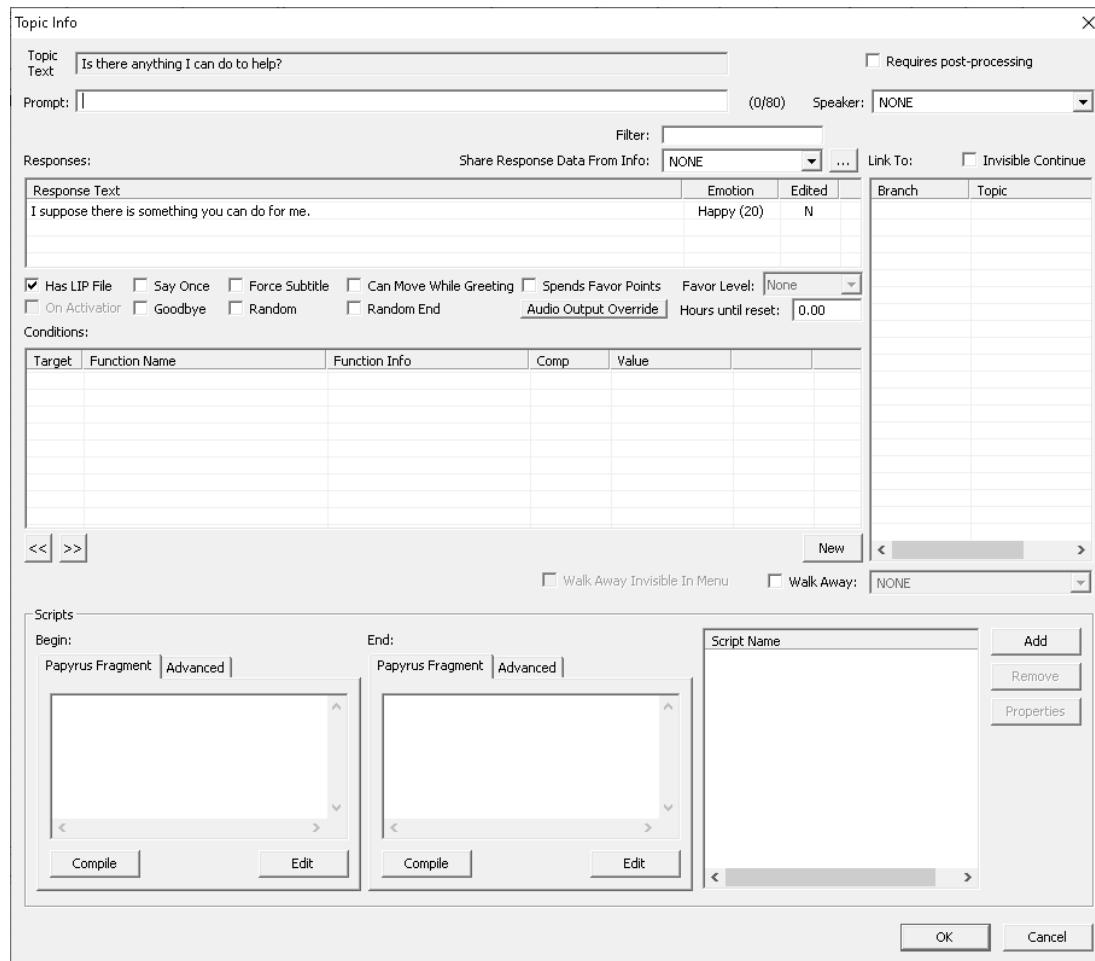


Figure 705 - Topic Info.

To add successive lines of dialogue, right click in the Responses list and select New.

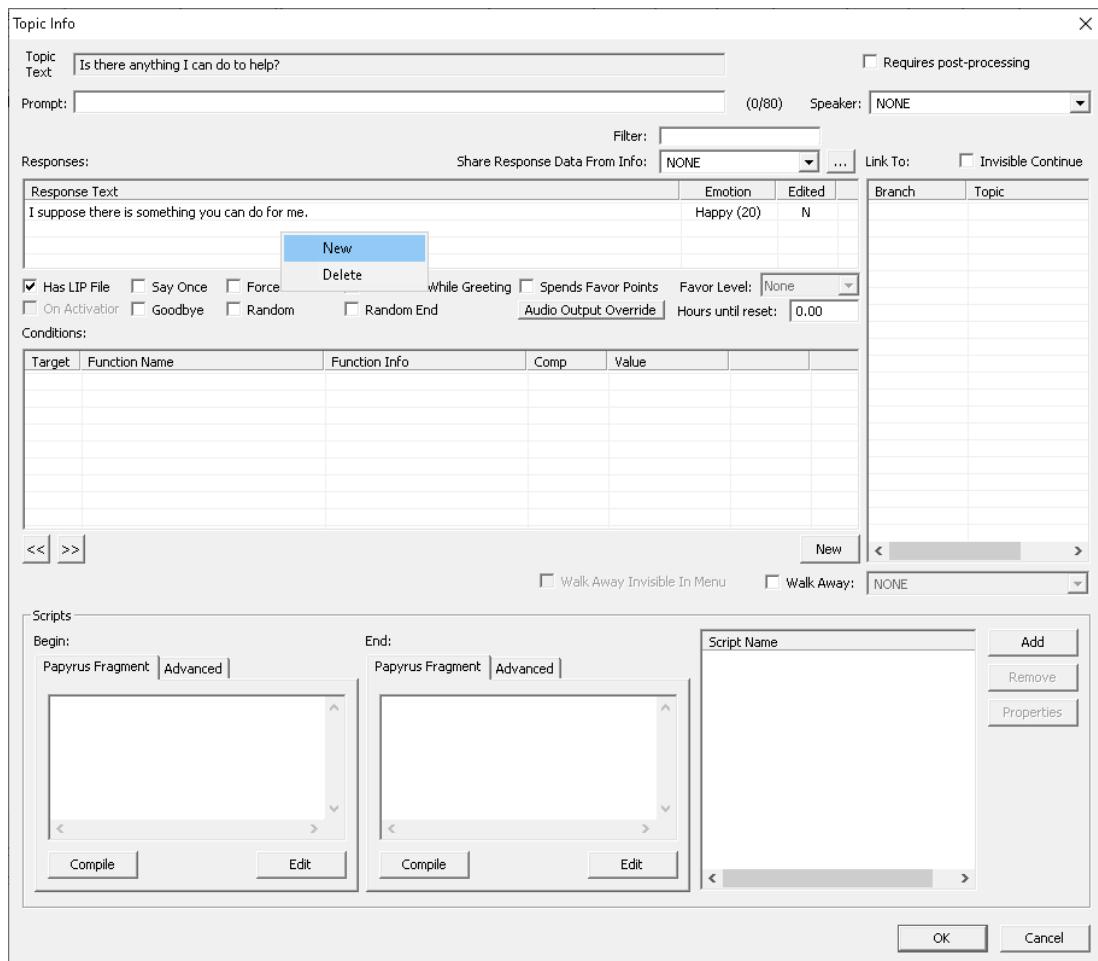


Figure 706 - Adding more lines of dialogue to this topic.

I went ahead and added a few more lines for this example.

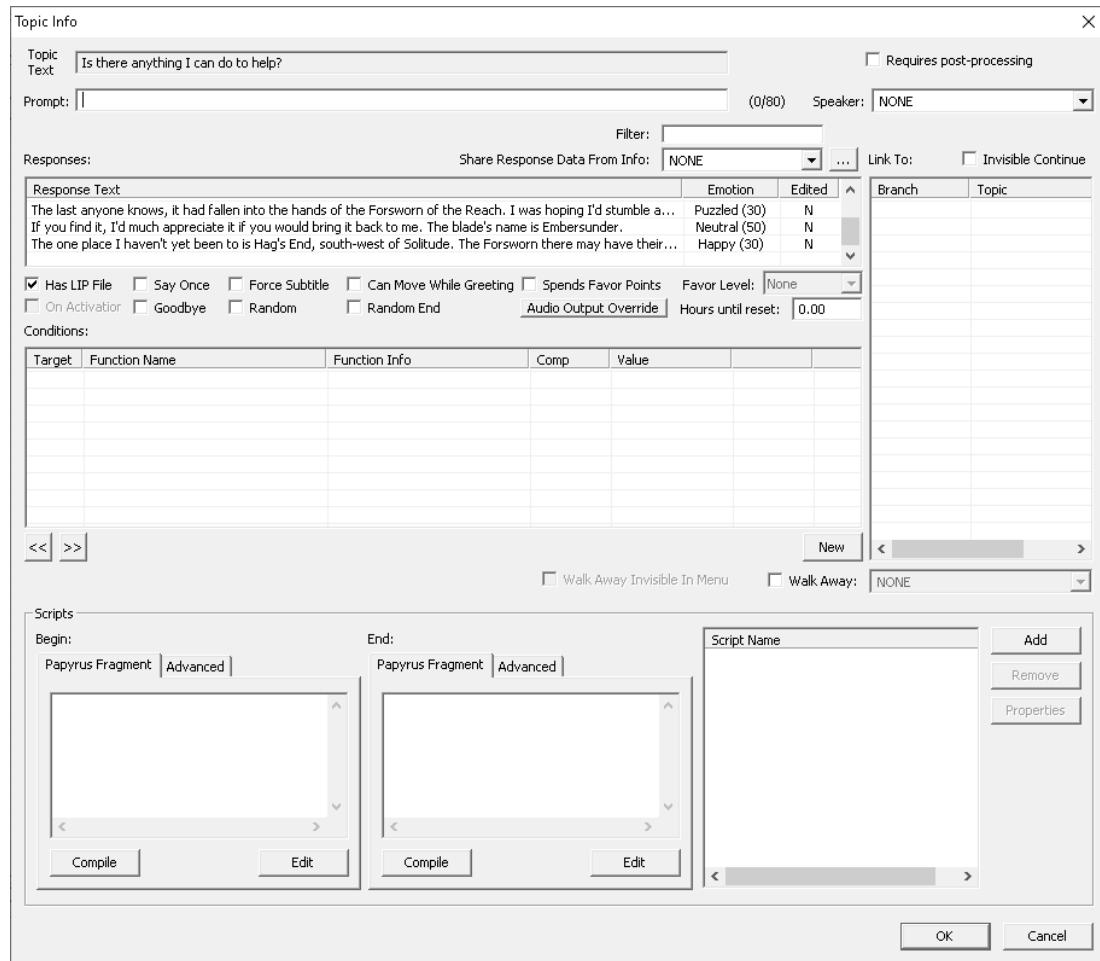


Figure 707 - Added multiple responses.

Next, we need to select the NPC that these lines will apply to.

Right-click in the Conditions list and select New.

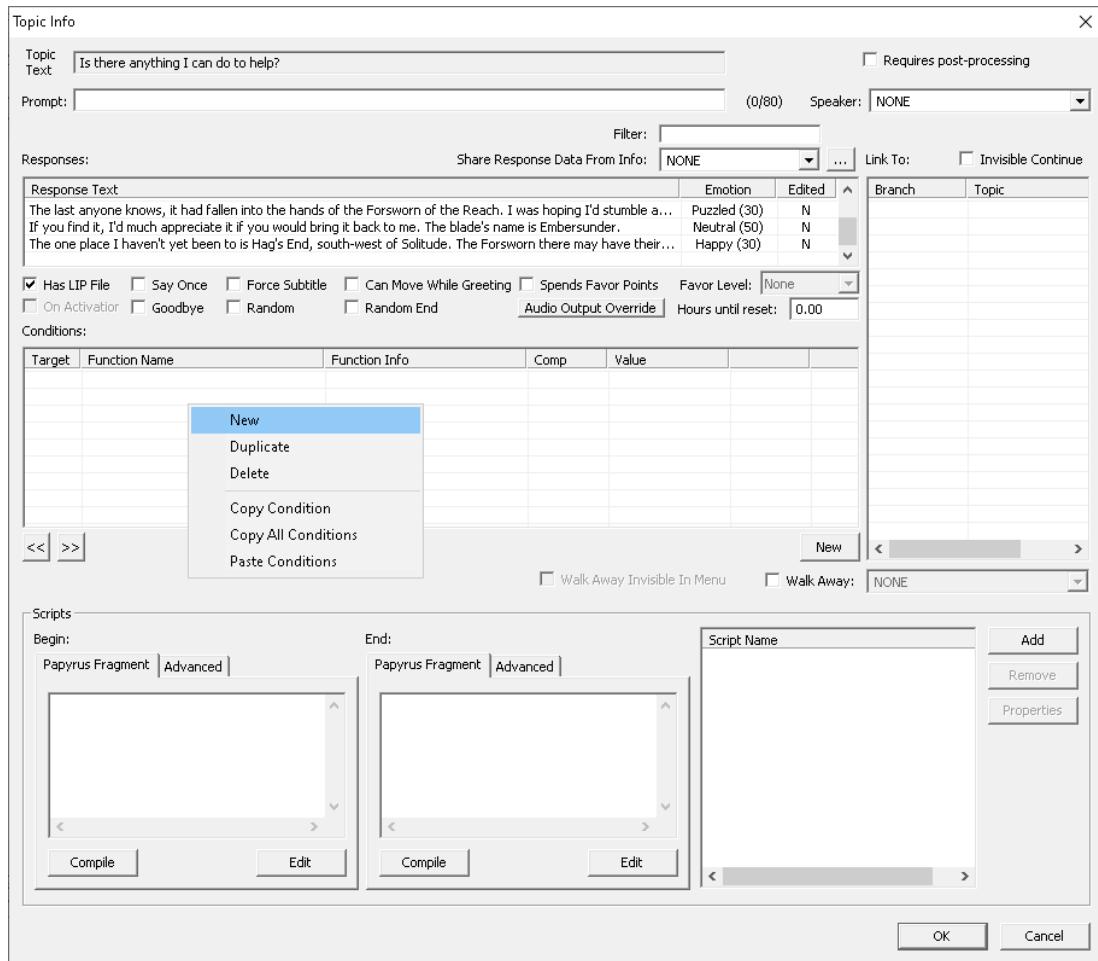


Figure 708 - Adding the first dialogue condition.

Ensure the Condition Function is set to GetIsID then click on the button next to it that currently says 'INVALID'.

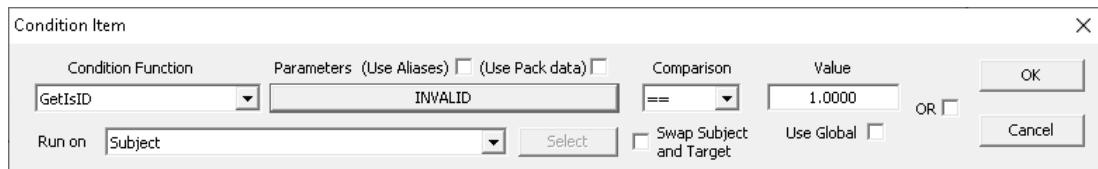


Figure 709 - Setting an NPC ID condition.

Select the form ID of the NPC. In this example, that's going to be our quest-giver Athir.

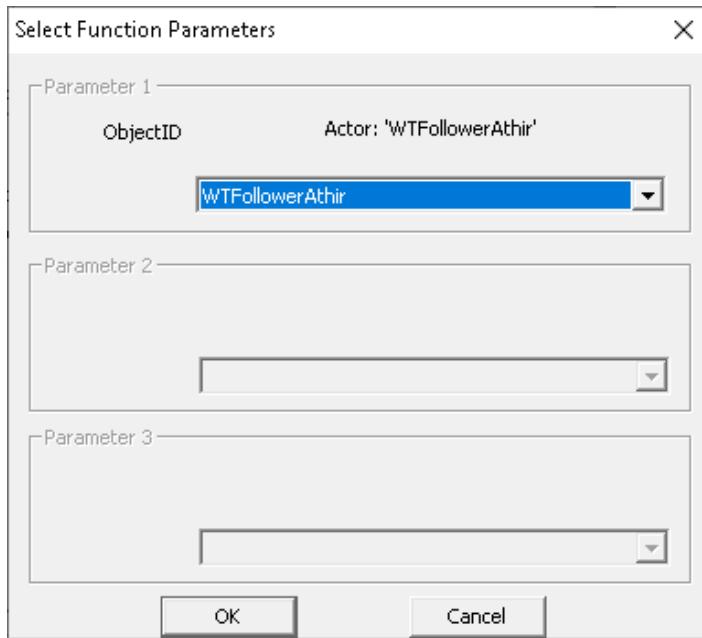


Figure 710 - Selecting the NPC ID.

Click OK to close Select Function Parameters.

Click OK to close Condition Item properties.



Figure 711 - NPC ID has been set.

Next, we need to add another condition to only show this topic if we haven't yet received this quest from our quest-giver.

Right-click again in the Conditions list and select New.

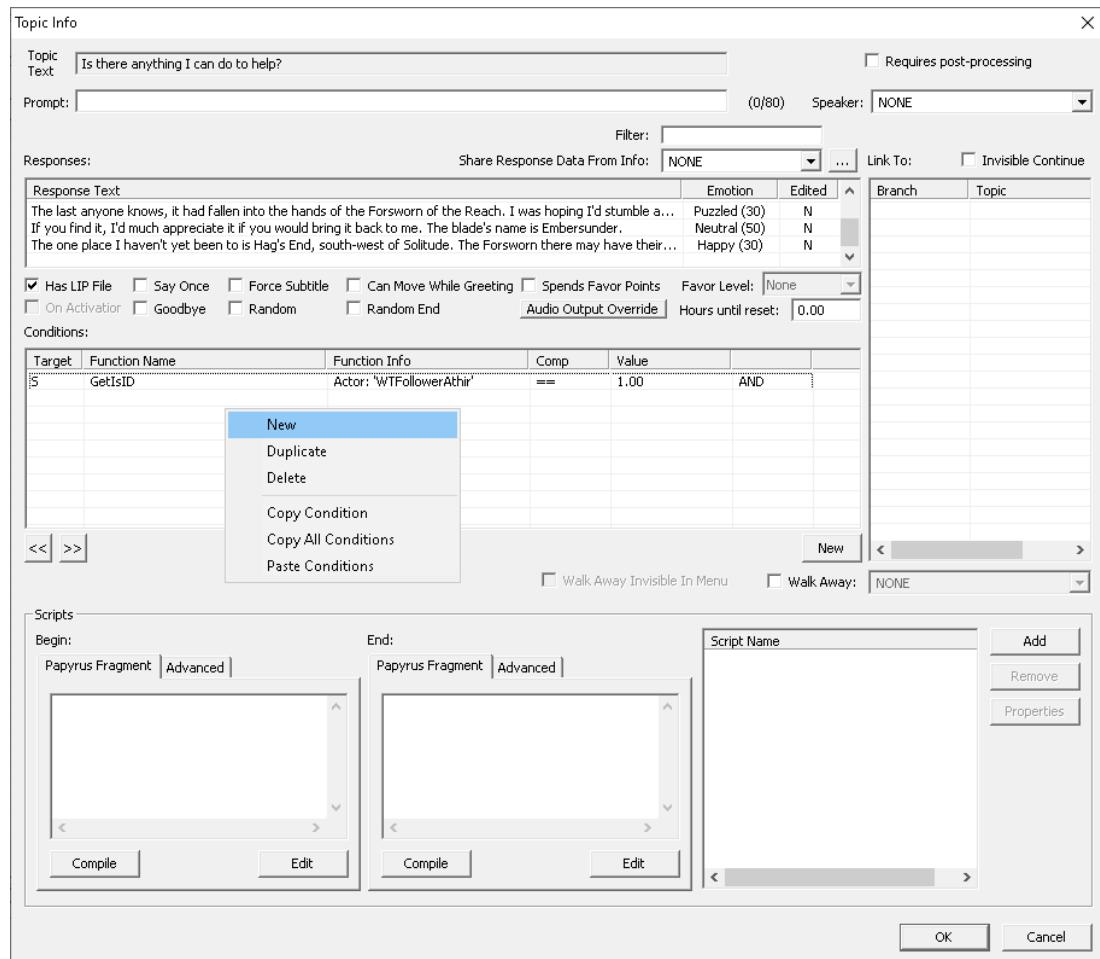


Figure 712 - Adding a GetStage condition.

Set the Condition Function to GetStage then click on the button next to it that currently says 'INVALID'.

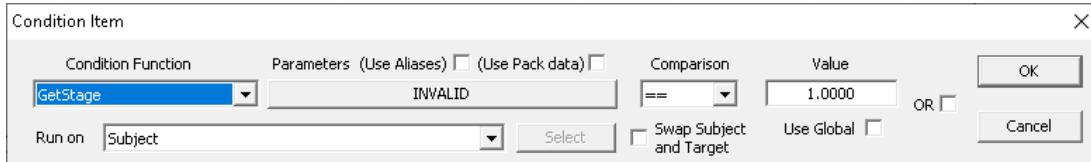


Figure 713 - Set the Condition Function to GetStage.

Select the Quest ID from the Quest drop-down, then click OK.

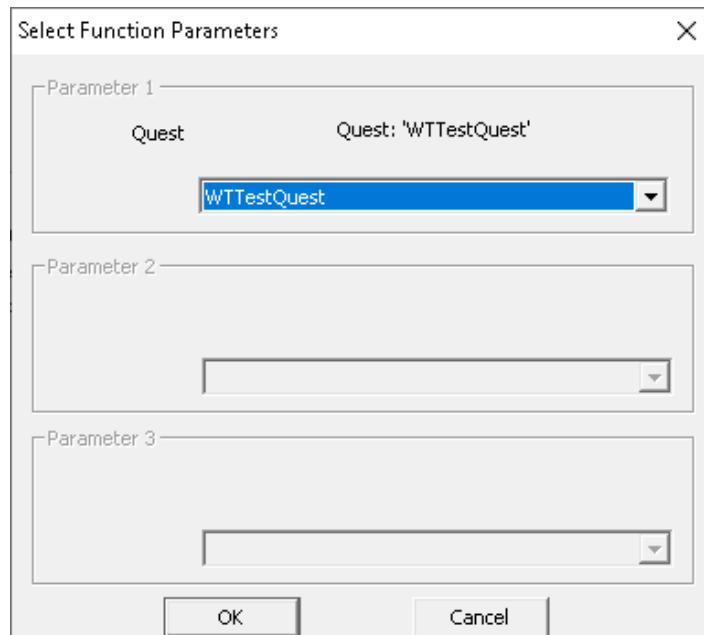


Figure 714 - Setting the Quest ID.

Set the Comparison operator to '==' and the value to 0. We only want this topic to appear if the current stage is 0.

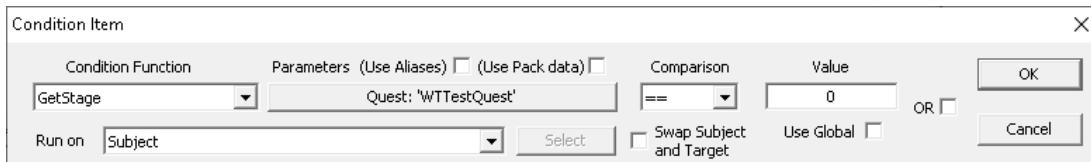


Figure 715 - Quest ID has been set.

Click OK to close out of Condition Item properties

In the Scripts section, enter in the following code into the End field and click Compile.

```
self.GetOwningQuest().SetStage(10)
```

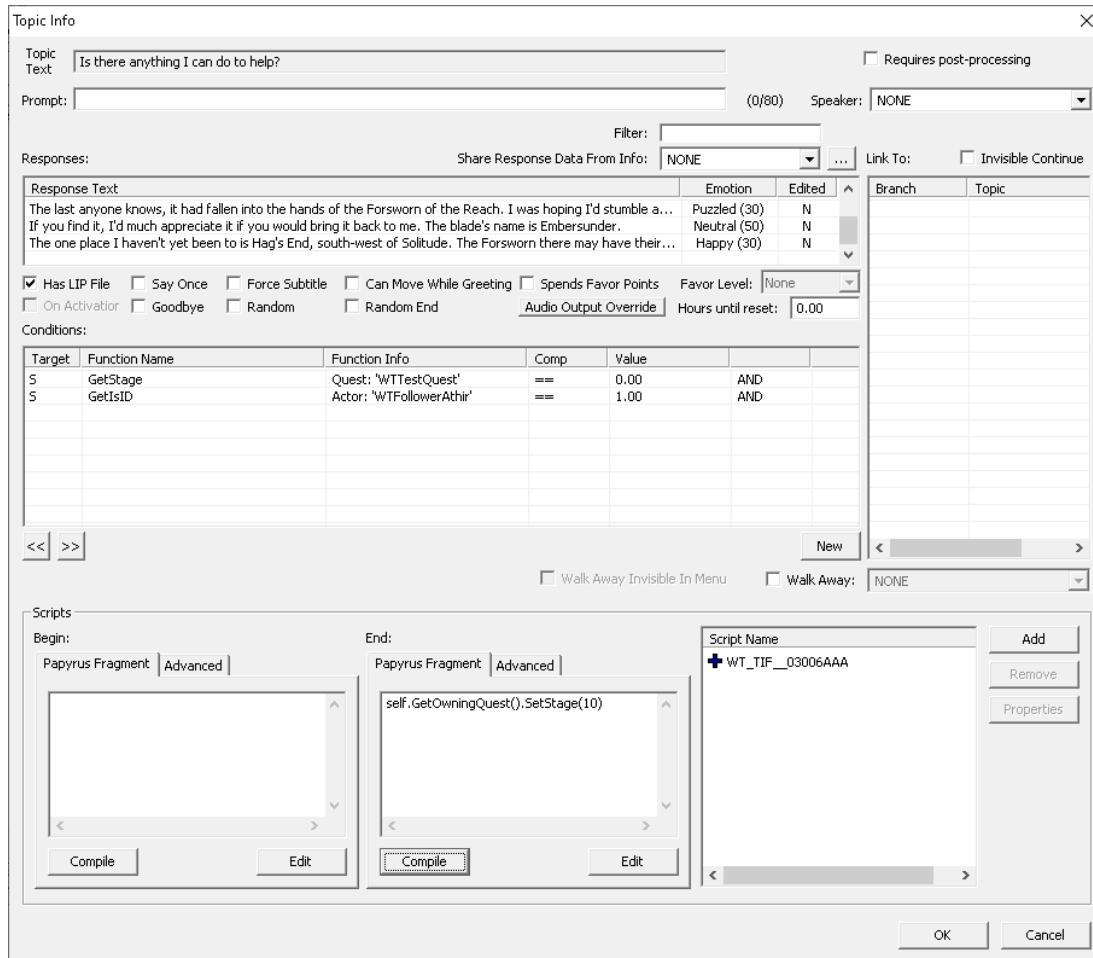


Figure 716 - Adding papyrus to execute when the NPC finishes speaking.

This code will be run when the NPC finishes speaking and will advance our sample fetch quest to stage 10.

When you press Compile, a new papyrus script fragment will be generated automatically. The papyrus fragment generated for this topic is WT_TIF_03006AAA.

Click OK to exit out of the Topic Info properties.

Our first branch and topic should now appear as per the following screenshot:

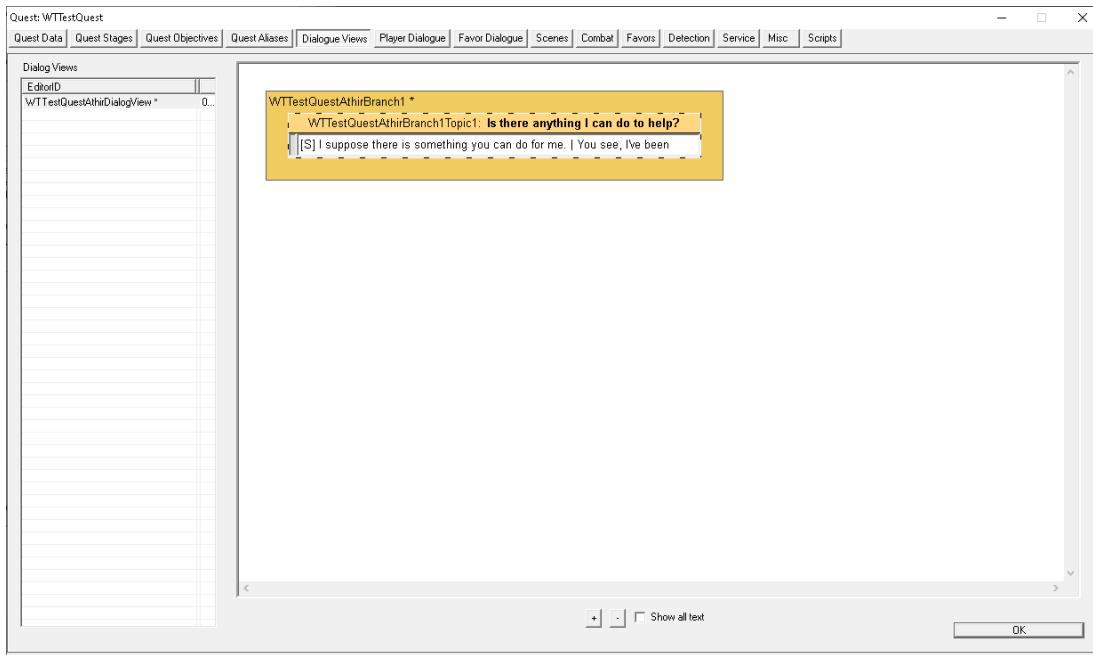


Figure 717 - First topic added.

So just to recap, our quest starts up behind the scenes at stage 0 when the mod is loaded.

After asking Athir 'Is there anything I can do to help', the quest will advance to stage 10, the sword will be moved into position, and the quest 'starts up' in the player's journal when the first objective becomes active.

When the player picks up the sword, the quest will be advanced to stage 20.

Now we need to add in a second dialogue branch to add the dialogue for returning the sword to our quest-giver.

Repeat the previous steps above to add a second branch and topic.

Note: When adding additional topics to an existing branch, the Creation Kit won't prompt us for a topic ID in a pop-up, so we need to manually change the topic ID in the ID field. For this example, I just called it WTTTestQuestDialogueChainTopic2.

In my example, I set the second branch ID to WTTTestQuestAthirBranch2 and set its topic ID to WTTTestQuestAthirBranch2Topic1.

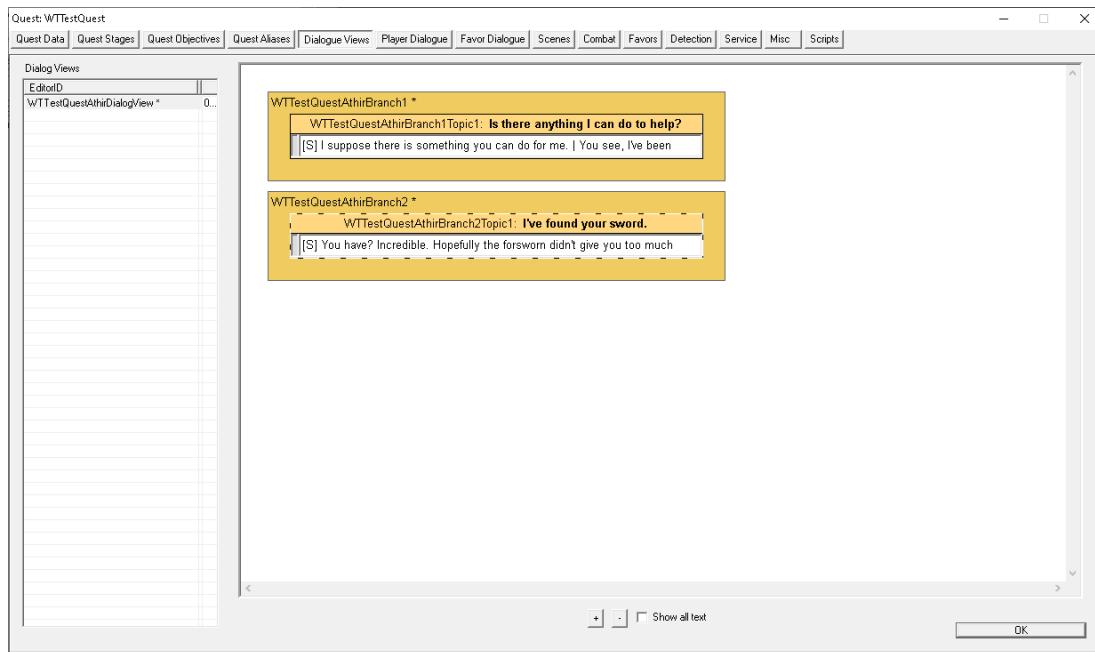


Figure 718 - Added the second branch and topic.

I set the Topic Text field for the second branch's topic to 'I've found your sword'.

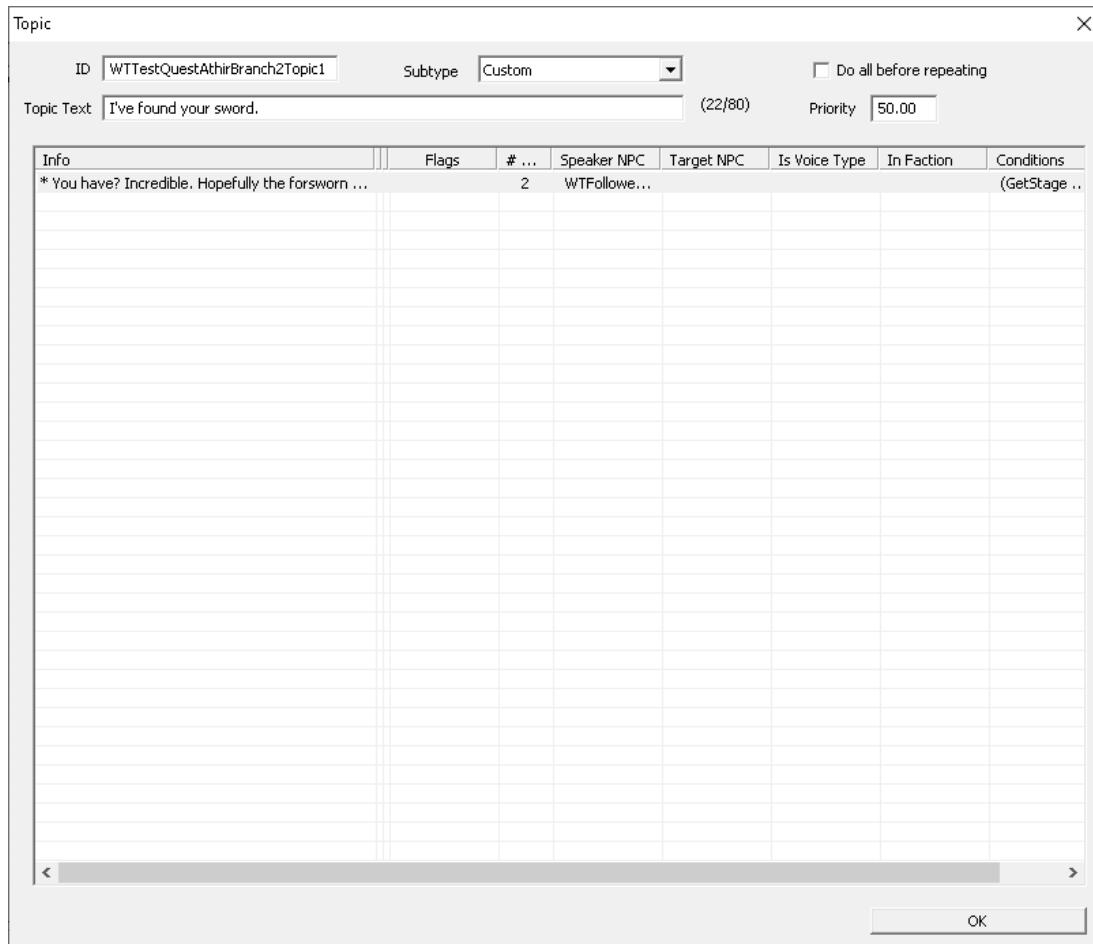


Figure 719 - The player's dialogue for the second branch.

The response for this topic was set up as per the following screenshot:

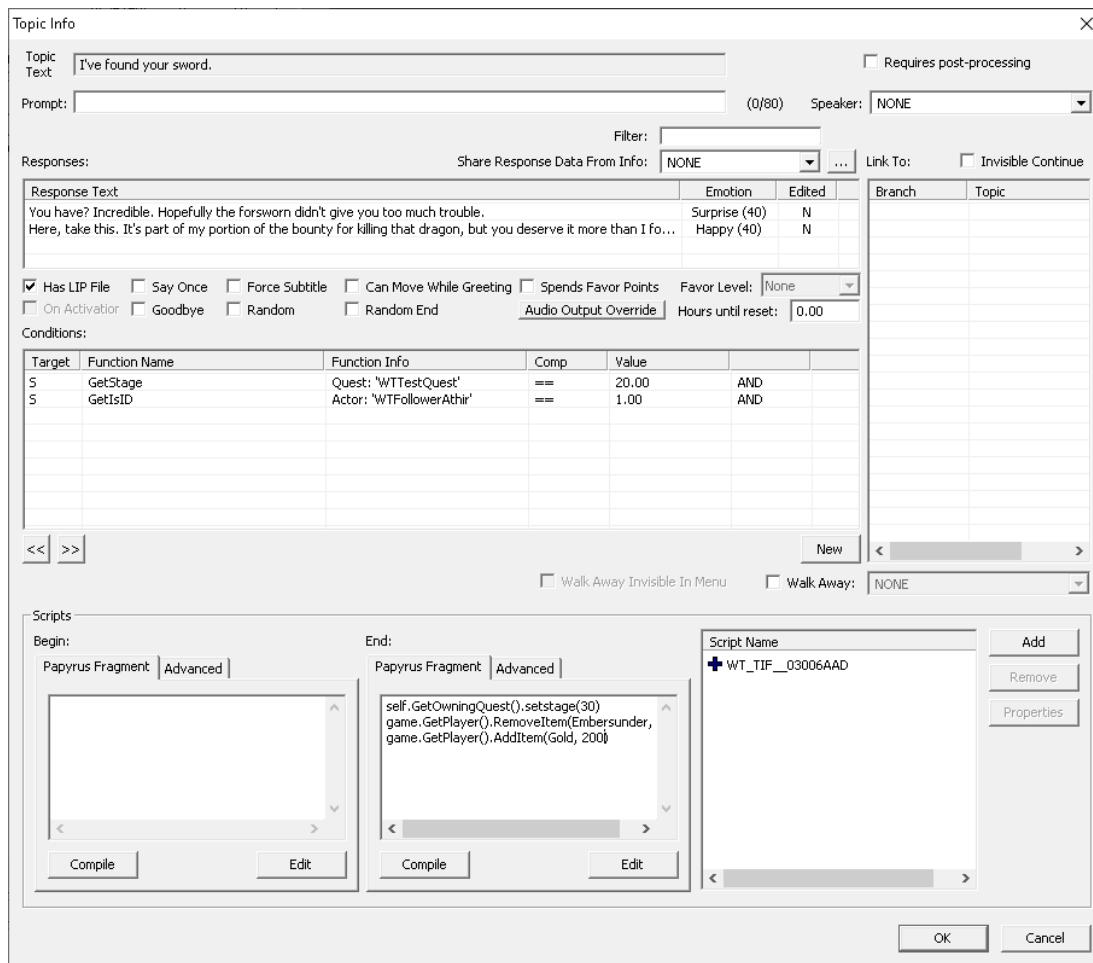


Figure 720 - NPC response for returning the sword.

In the Conditions list, this time I set the GetStage condition to '20' so this topic only appears once the player has the sword.

The papyrus added to the End field is as follows:

```
self.GetOwningQuest().setstage(30)
game.GetPlayer().RemoveItem(Embersunder, 1)
game.GetPlayer().AddItem(Gold, 200)
```

To break down this code, first we set our quest stage to 30 so we can end it. Then we remove the sword from the player's inventory. Lastly, we give the player a 200 gold reward for completing the quest.

Currently, if we try to Compile the End code, we'll get the following warning:

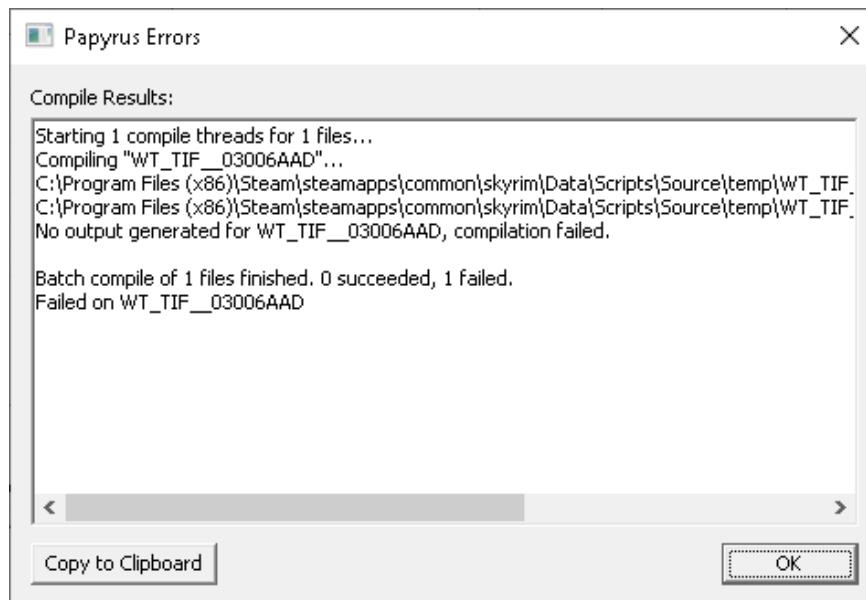


Figure 721 - Papyrus warning.

This is occurring because the 'Embersunder' and 'Gold' variables haven't yet been defined in the script fragment.

To rectify this situation, highlight the papyrus fragment and click on the Properties button.

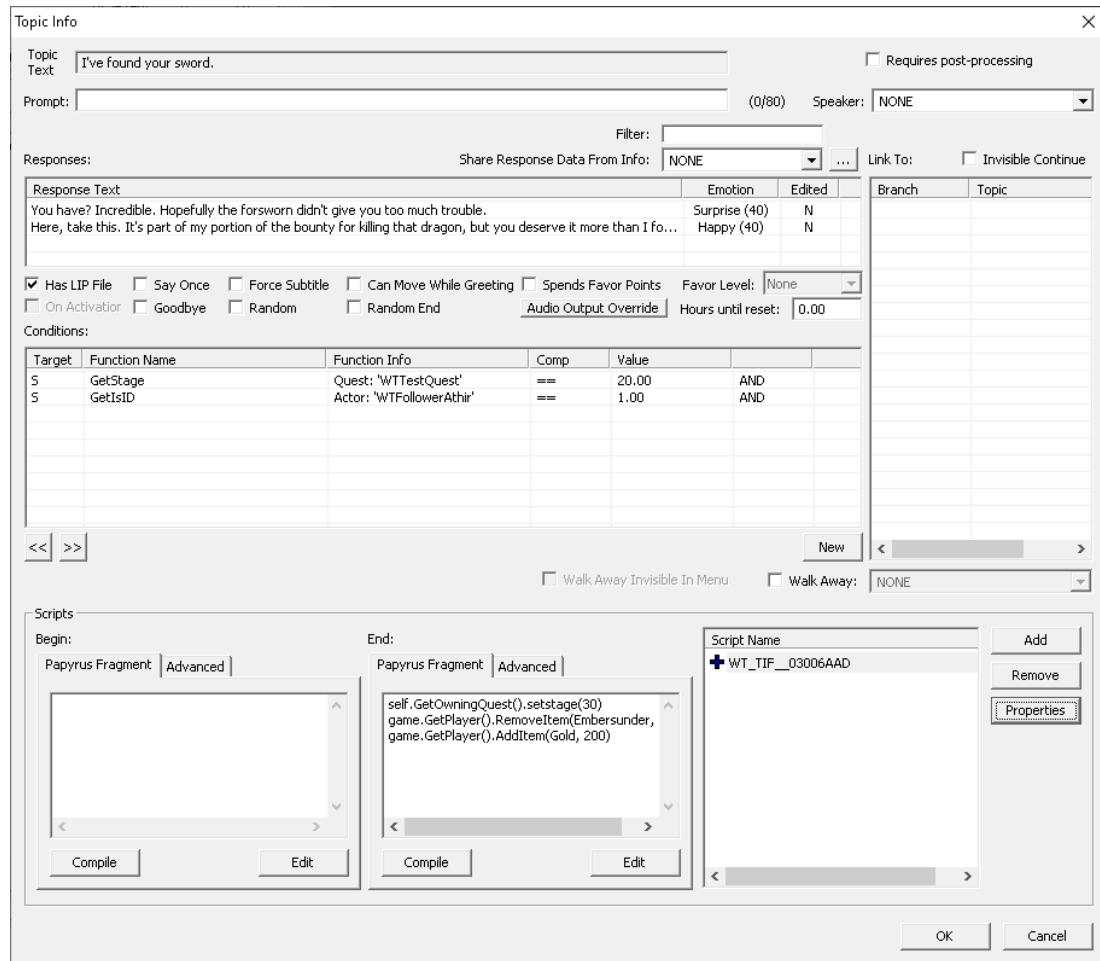


Figure 722 - Opening properties of the papyrus fragment.

If you see the following error, just click OK and ignore it.

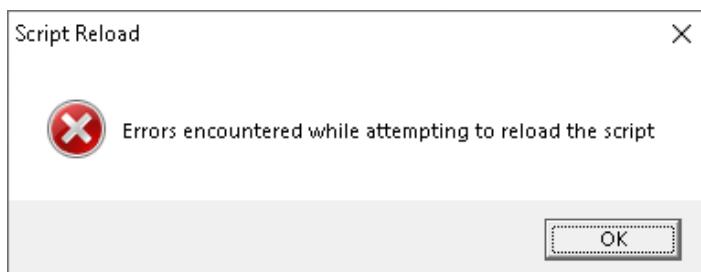


Figure 723 - Script reload error.

Click on the Add Property button.



Figure 724 - Adding a new script property.

Set the Type drop-down to MiscObject and set the Name field to ‘Gold’ then click OK.

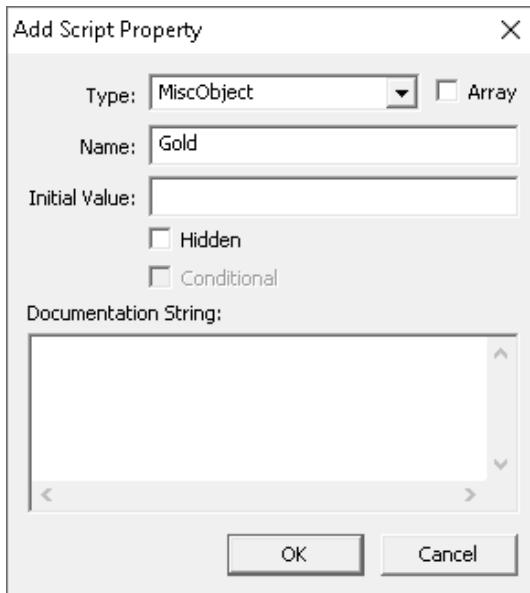


Figure 725 - Adding the Gold variable.

Click on Add Property again. This time set the Type drop-down to WEAPON and the Name field to Embersunder then click OK.

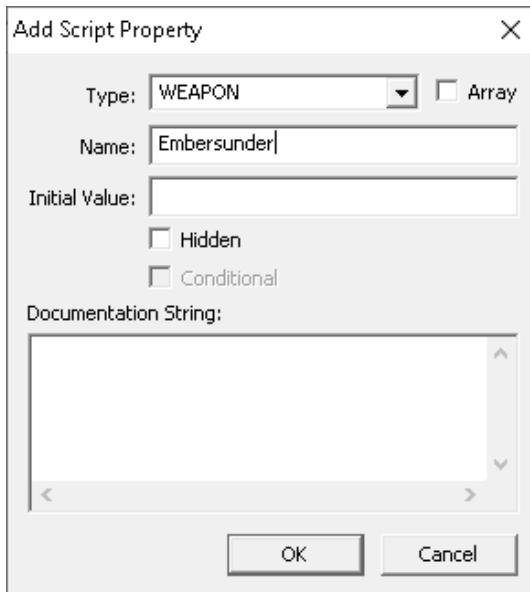


Figure 726 - Adding the Embersunder variable.

The two variables should now appear in the Properties list. Next, we need to set their values.

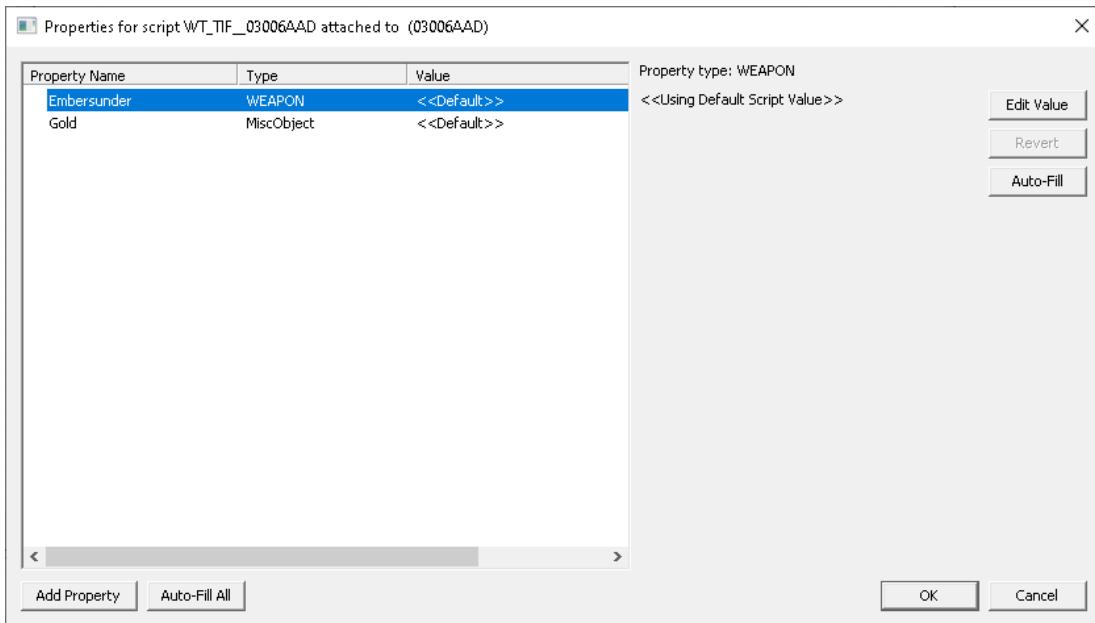


Figure 727 - The two variables added.

Highlight the Embersunder variable and click on the Edit Value button.

Set the Pick Object drop-down to point to the sword the player obtained.

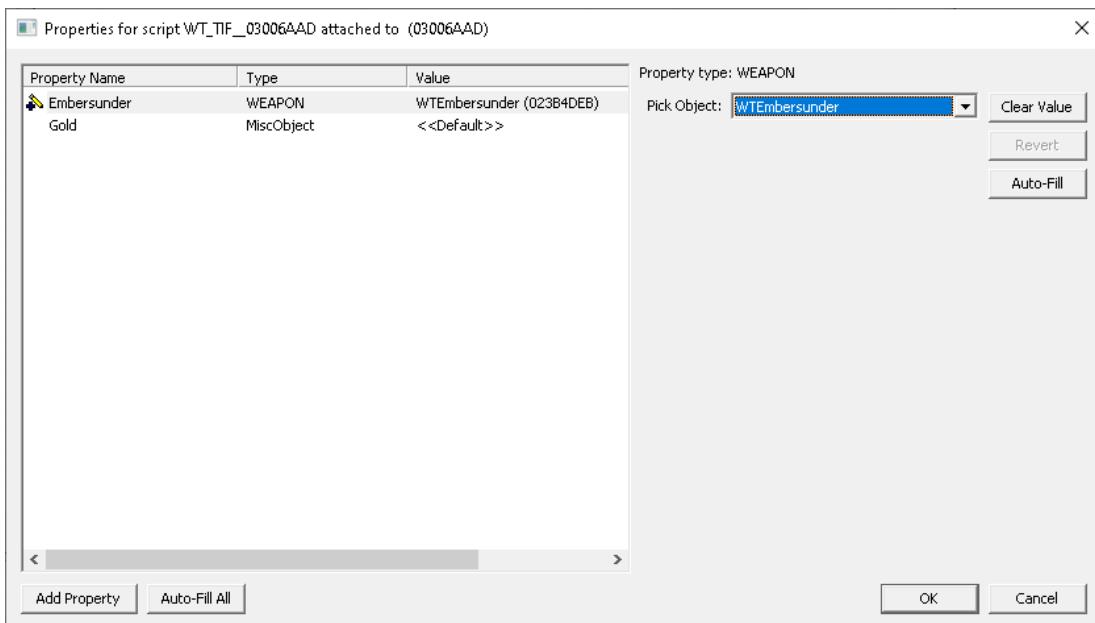


Figure 728 - Setting the Embersunder variable's value.

Next, highlight the Gold variable and click on the Edit Value button.

Set the Pick Object drop-down to point to Gold001.

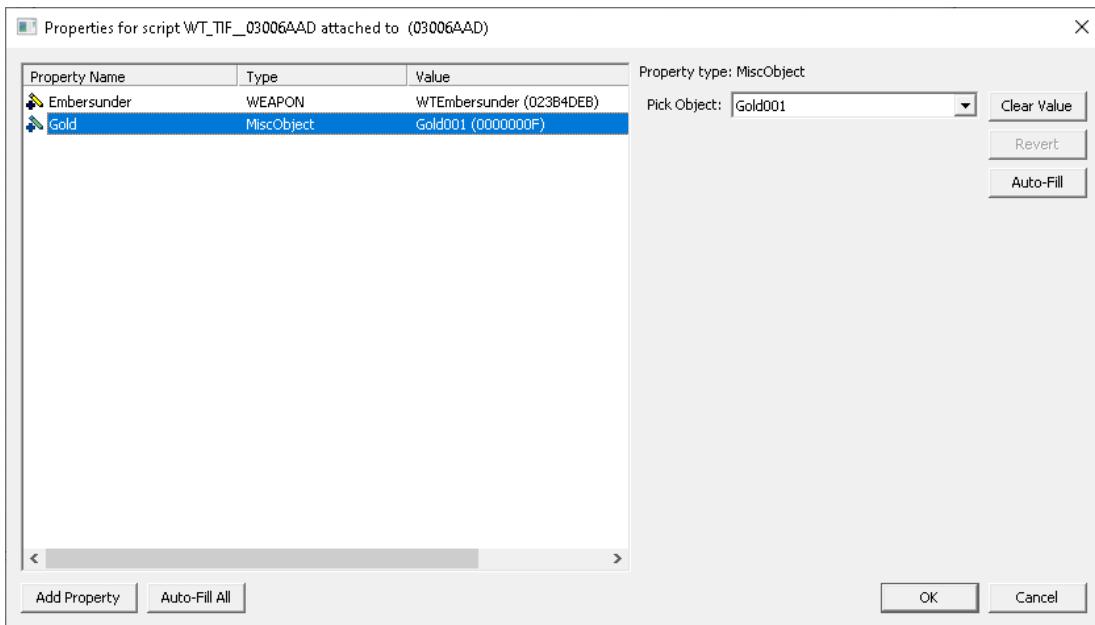


Figure 729 - Setting the Gold variable's value.

Click OK to close out of script properties.

Now when we compile the papyrus code in End, we should no longer receive an error.

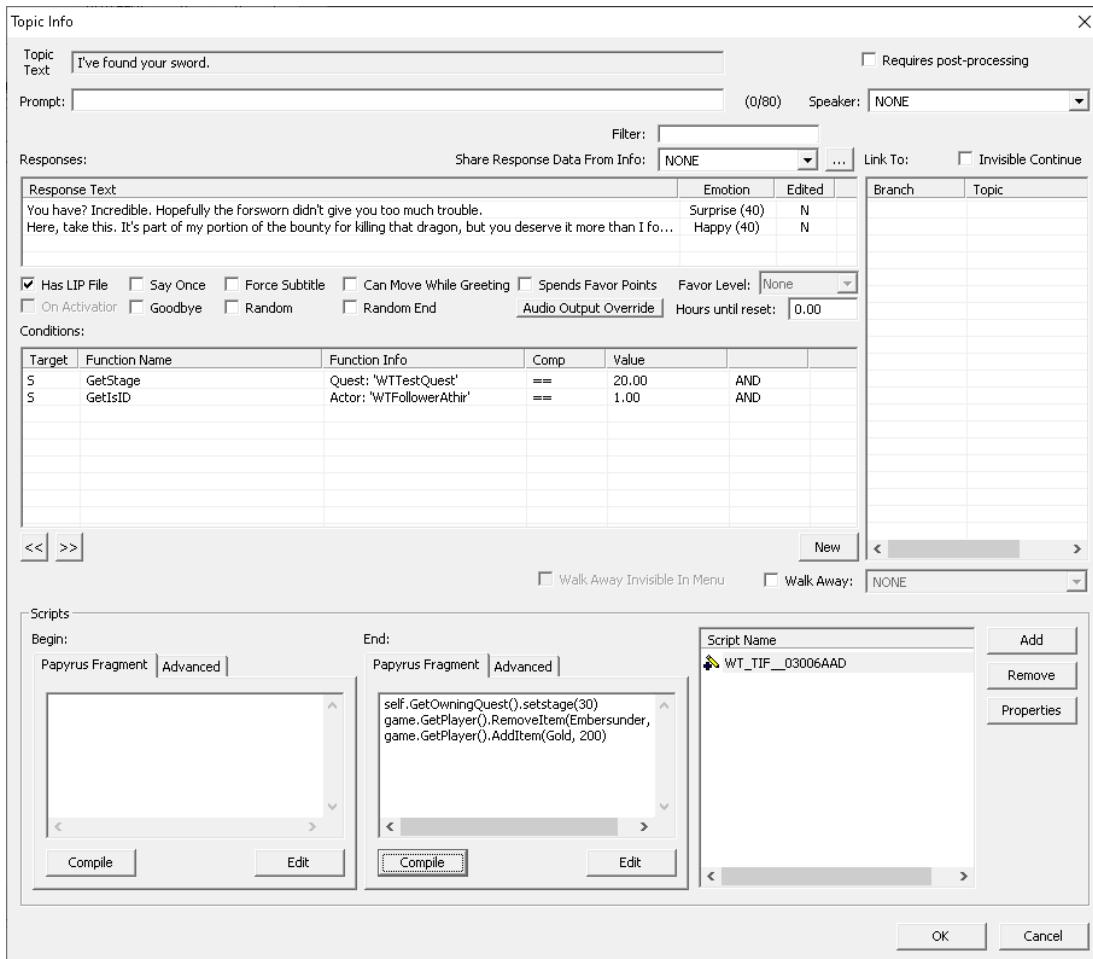


Figure 730 - Papyrus compiled successfully.

Click OK to close out of Topic Info, click OK to close out of Topic, then click OK to close out of Quest properties.

CHAINING DIALOGUE TOPICS TOGETHER

Say for example we wanted an NPC to say something and give the player multiple choices to choose from. We can accomplish this by chaining topics together in Dialogue Views.

Let's start by creating a new branch.

In the Dialogue Views tab under WITTestQuestDialogueView, right-click in the empty area and select 'Create branch'.

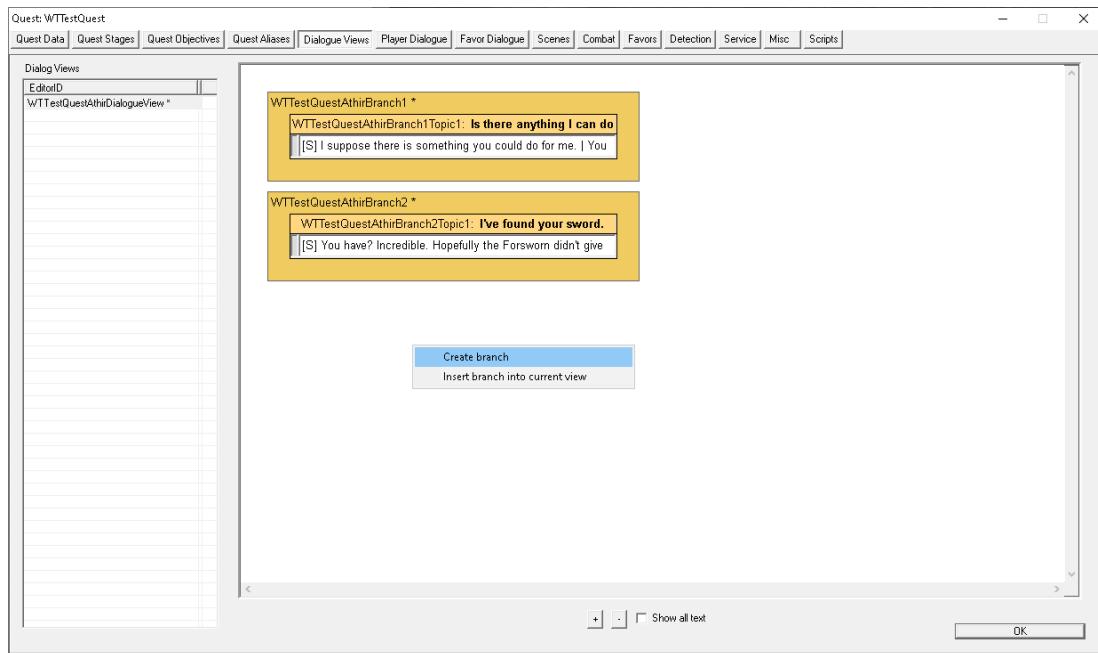


Figure 731 - Creating a new branch.

Set the branch ID then click OK. For this example I set it to WITTestQuestDialogueChain.

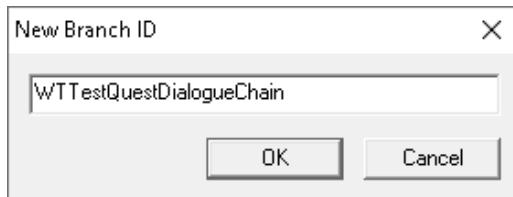


Figure 732 - Setting the branch ID.

Set the topic ID of the first topic then click OK. Again, for this example I just named it WTTTestQuestDialogueChainTopic1.



Figure 733 - Setting the topic ID of the first topic.

Double-click or right-click on the new topic and select 'Edit topic'.

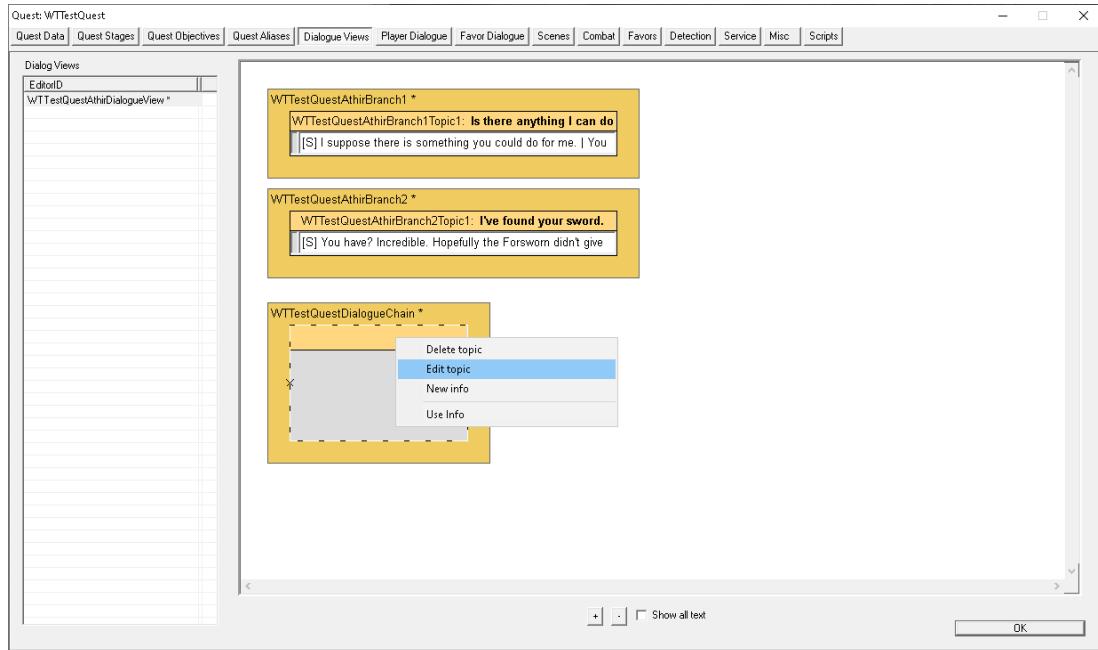


Figure 734 - Editing the first topic in the dialogue chain.

Enter topic text into the Topic Text field then right-click in the Info list below and select New to create a new NPC response.

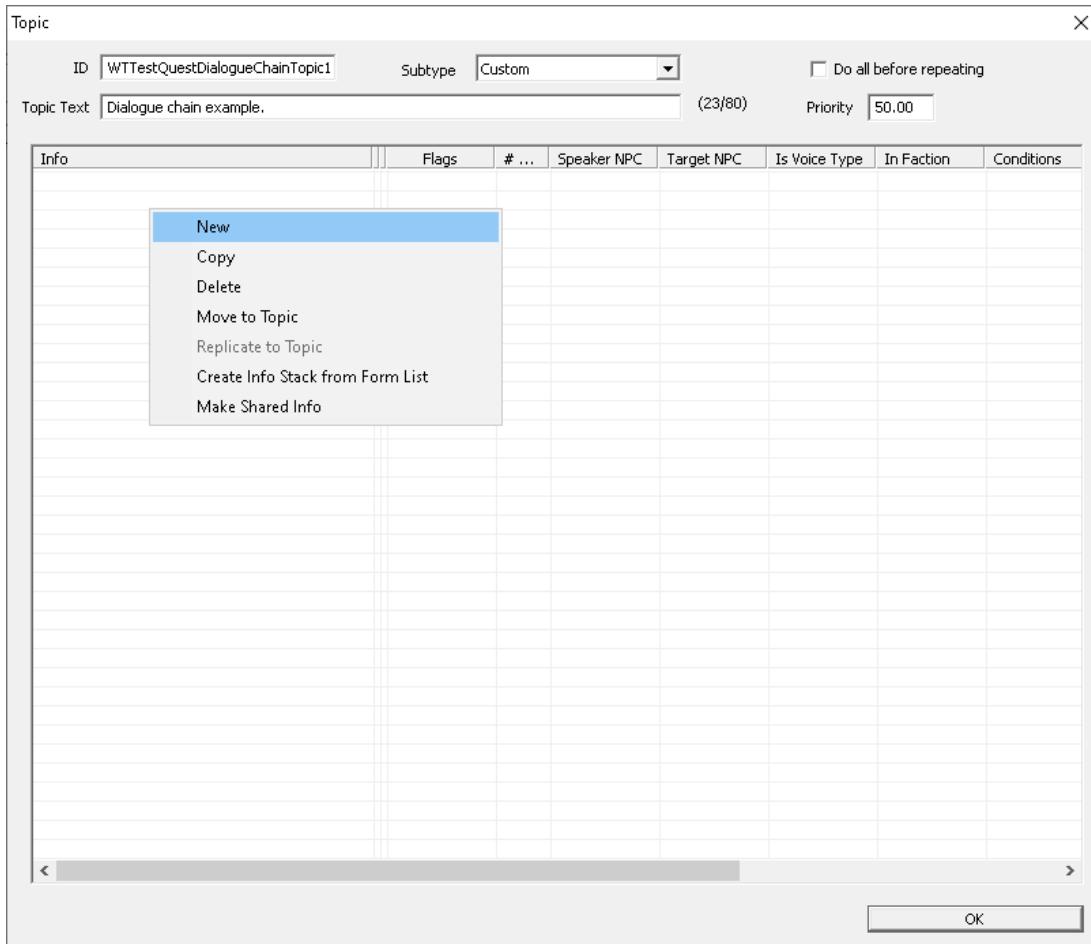


Figure 735 - Setting the topic text and adding a response.

Enter in the dialogue that the NPC will respond with in the Response Text field then click OK.

Obviously not the most riveting line of dialogue. I'm just keeping things simple here for the sake of this example.

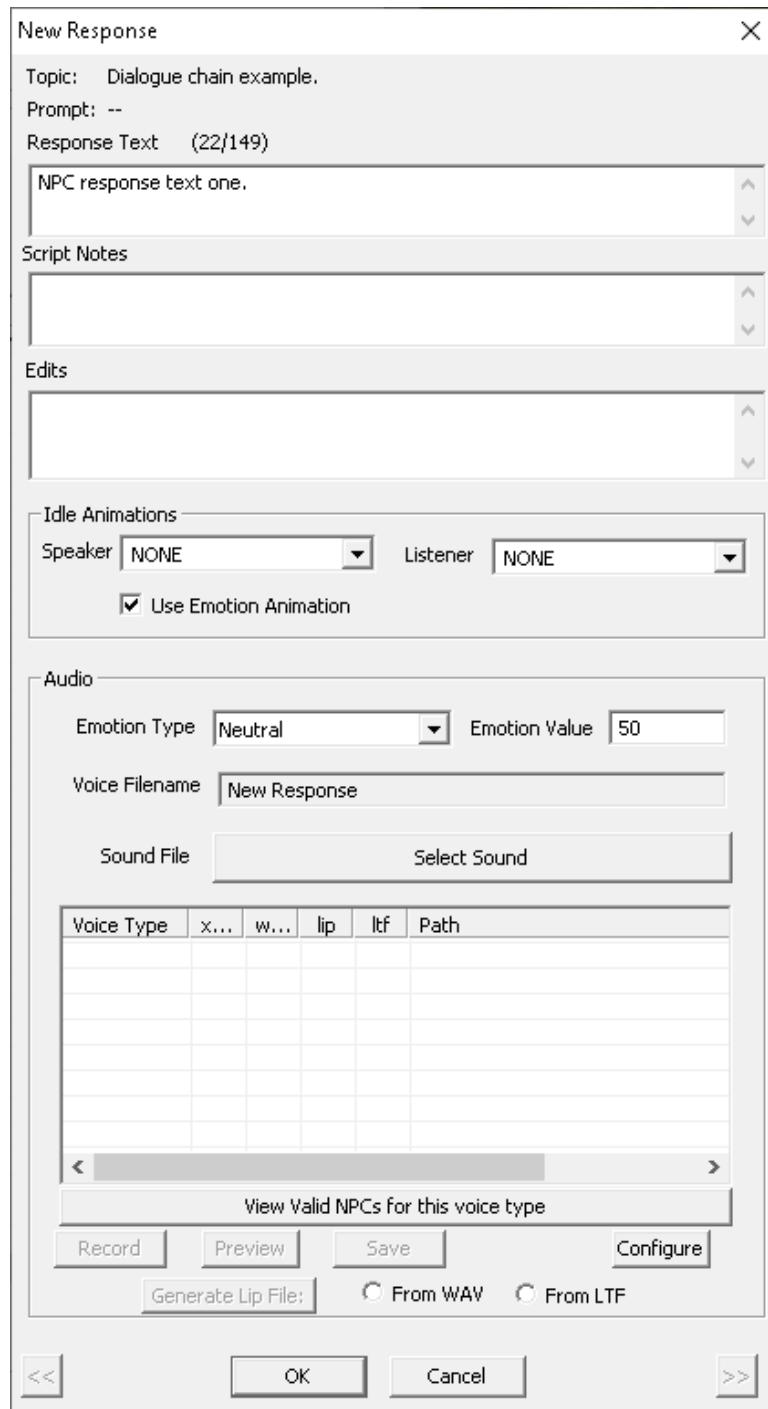


Figure 736 - The NPC's first response.

Right-click in the Conditions list and select New.

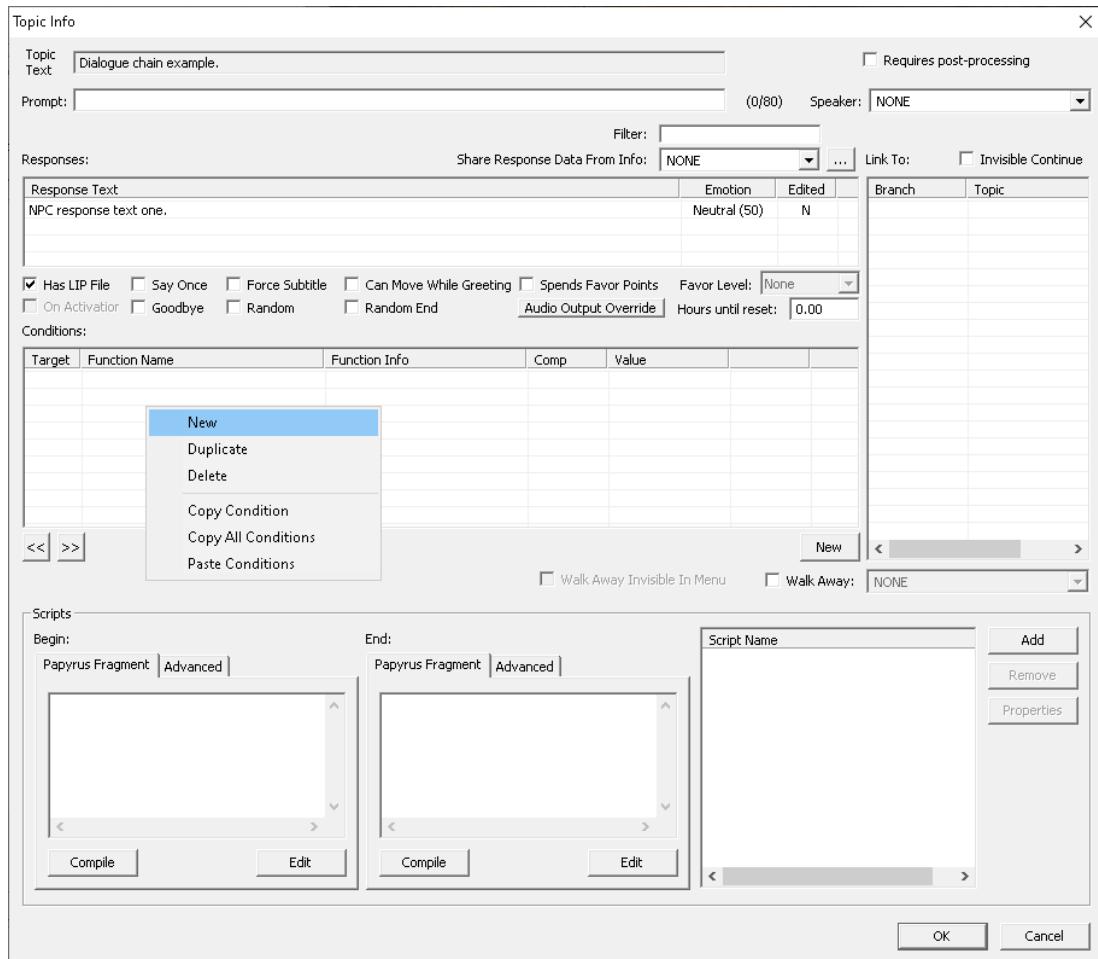


Figure 737 - Adding a new condition.

Ensure the Condition Function drop-down is set to GetIsID and set the actor. Since I'm setting this example up in the WITTestQuest example, I'm just going to reuse Athir.

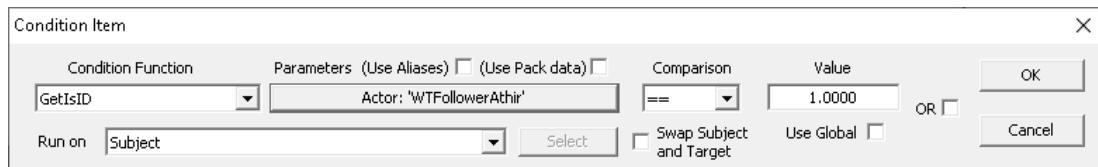


Figure 738 - Setting Athir as the speaker of this line of dialogue.

Click OK to close out of Condition Item.

Click OK to close out of Topic Info.

Click OK to close out of Topic.

Right-click in the WTTTestQuestDialogueChain branch and select 'Add topic'.

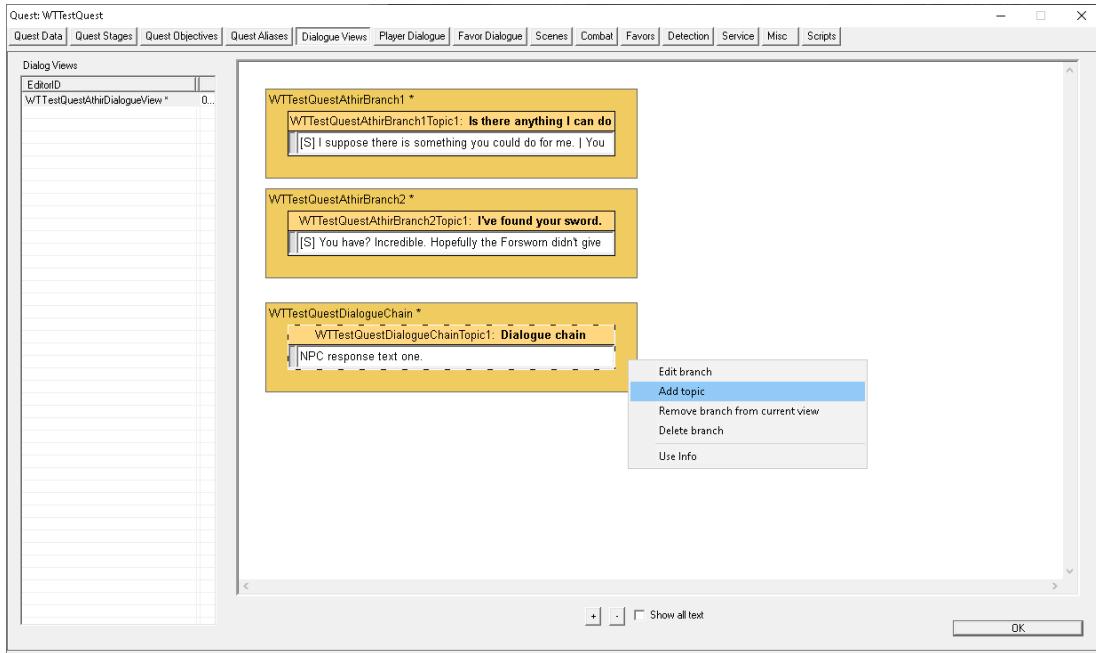


Figure 739 - Adding a second topic to the WTTTestQuestDialogueChain branch.

Set the player response in the Topic Text field then right-click in the Info list and select New.

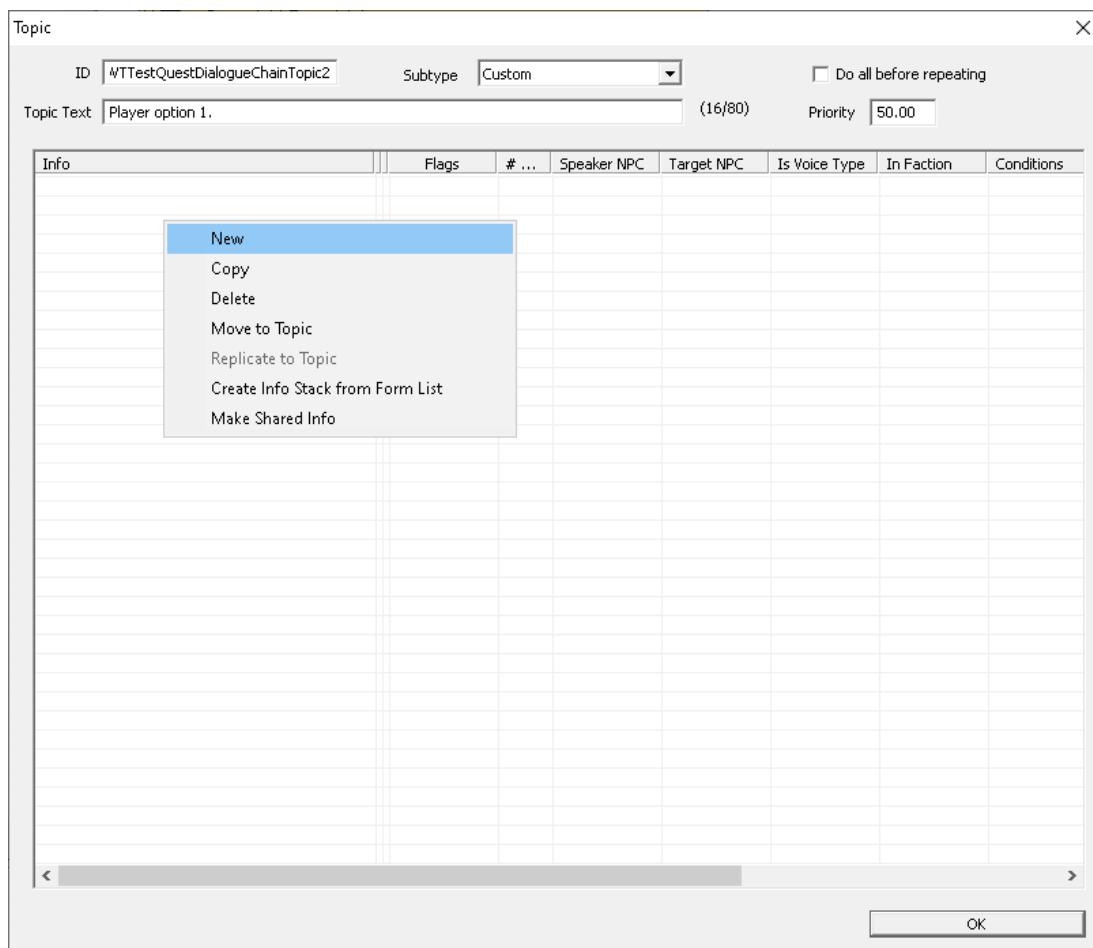


Figure 740 - Setting the ID, topic text and adding a response.

Repeat the steps mentioned earlier to add in the NPC's response.

Then add a third topic to the same branch by once again right-clicking in the WTTTestQuestDialogueChain branch and selecting ‘Add topic’.

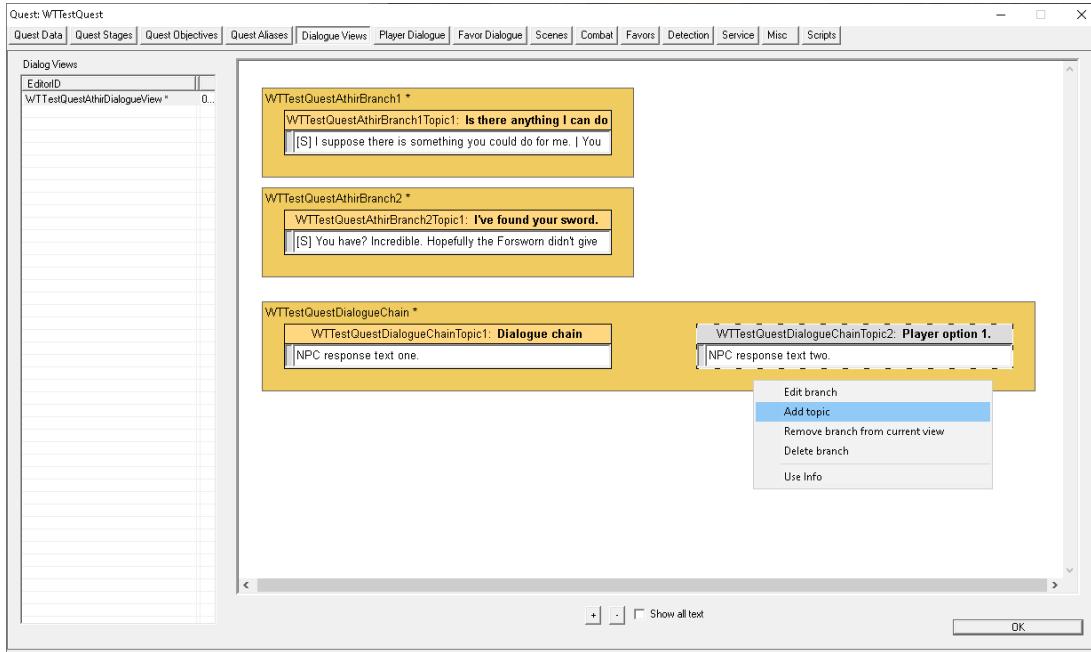


Figure 741 - Adding a third topic to WTTTestQuestDialogueChain.

We should now have three topics in the same branch. Just align them into position like they are in the screenshot below:

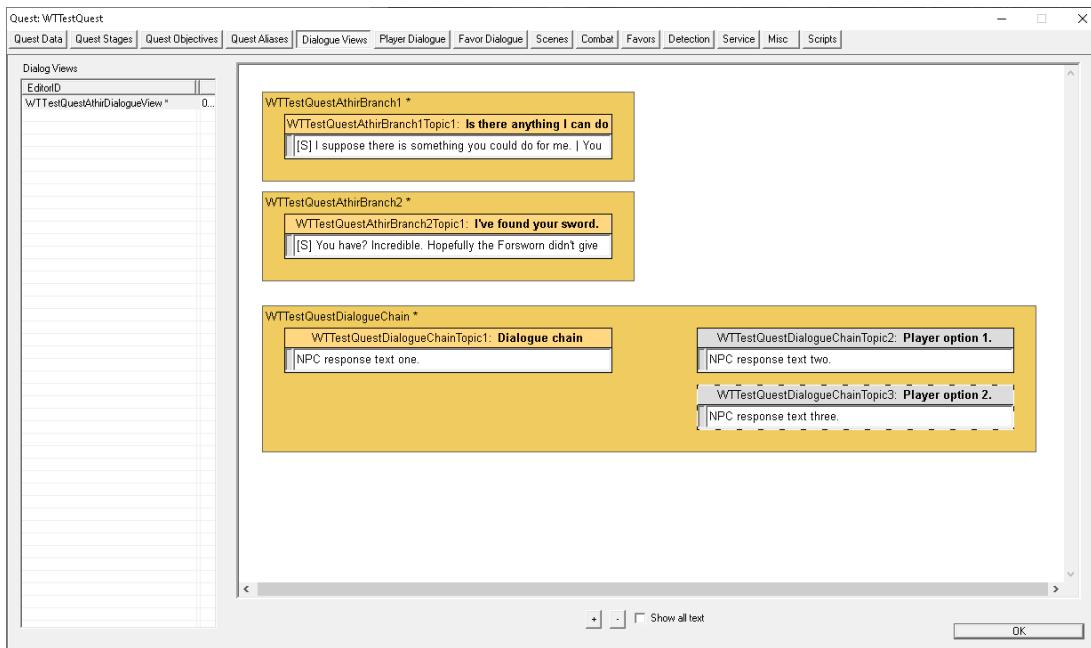


Figure 742 - Three topics in the same branch.

Now to chain them together.

Click on WTTTestQuestDialogueChainTopic1 to select it. When you hover over the NPC response text, your mouse cursor should change into a hand icon. Left-click and drag from the response text in WTTTestQuestDialogueChainTopic1 to WTTTestQuestDialogueChainTopic2 to link these two topics together.

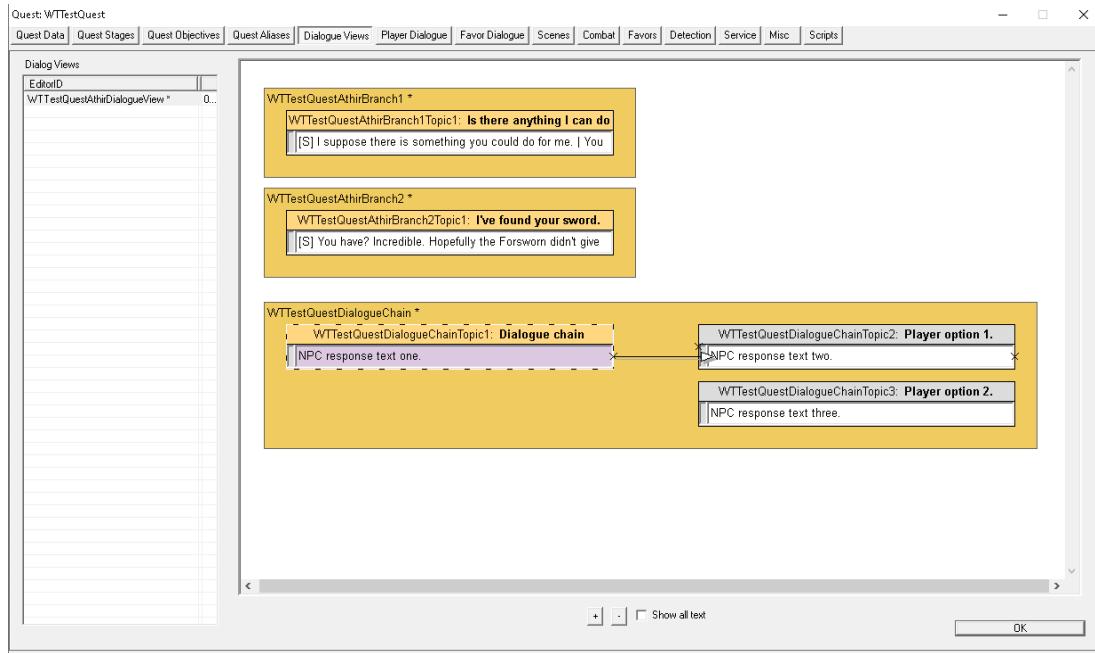


Figure 743 - Linking topic 1 to topic 2.

Repeat these steps to link WTTTestQuestDialogueChainTopic1 to WTTTestQuestDialogueChainTopic3 as well.

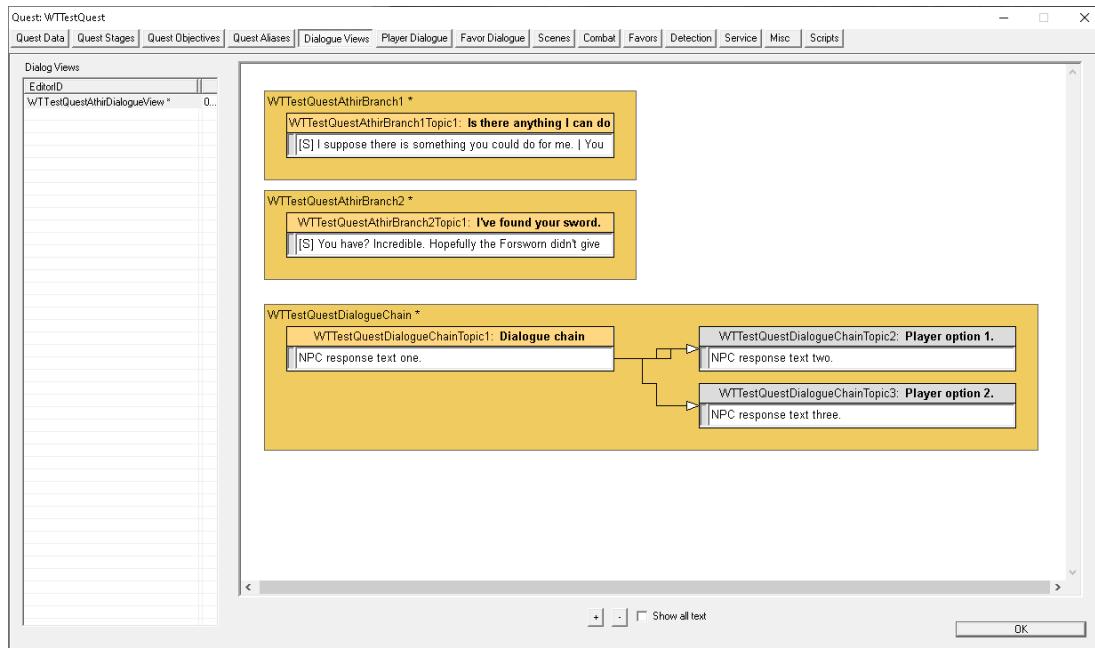


Figure 744 - Linking topic 1 to topic 3.

In-game we should now see those two choices presenting themselves when we speak to Athir after selecting 'Dialogue chain example'. But what if the player tabs out of dialogue before choosing the next response?

By default, if the player tabs out before selecting a response, dialogue with the NPC will end. But we can set up a default choice instead to continue dialogue with the NPC.

In order to do this, open the Topic Info properties of the preceding line of dialogue, tick Walk Away, then select the topic to automatically progress to should the player attempt to tab out of dialogue.

In the following example, I set the Walk Away topic to WTTTestQuestDialogueChainTopic2.

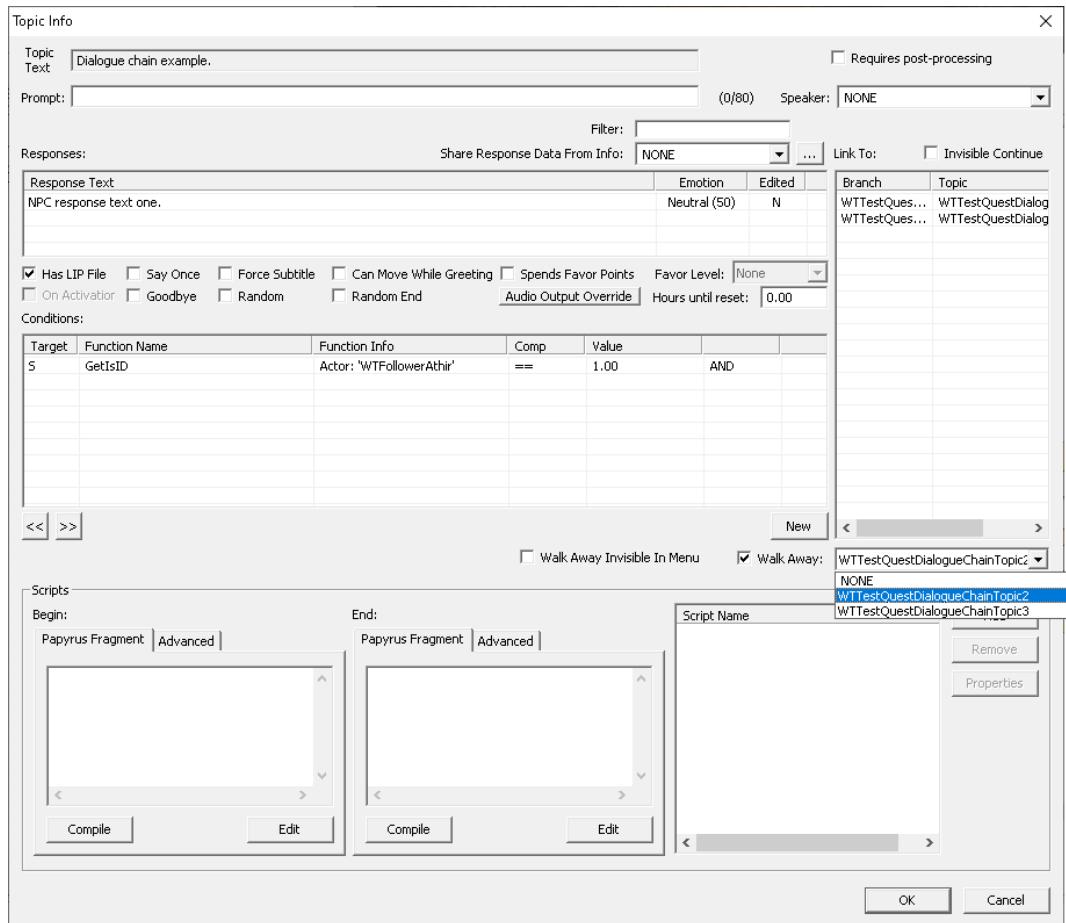


Figure 745 - Setting the walk away topic for 'Dialogue chain example'.

In the Dialogue Views tab, we should now see a red arrow between WTTTestQuestDialogueChainTopic1 and WTTTestQuestDialogueChainTopic2.

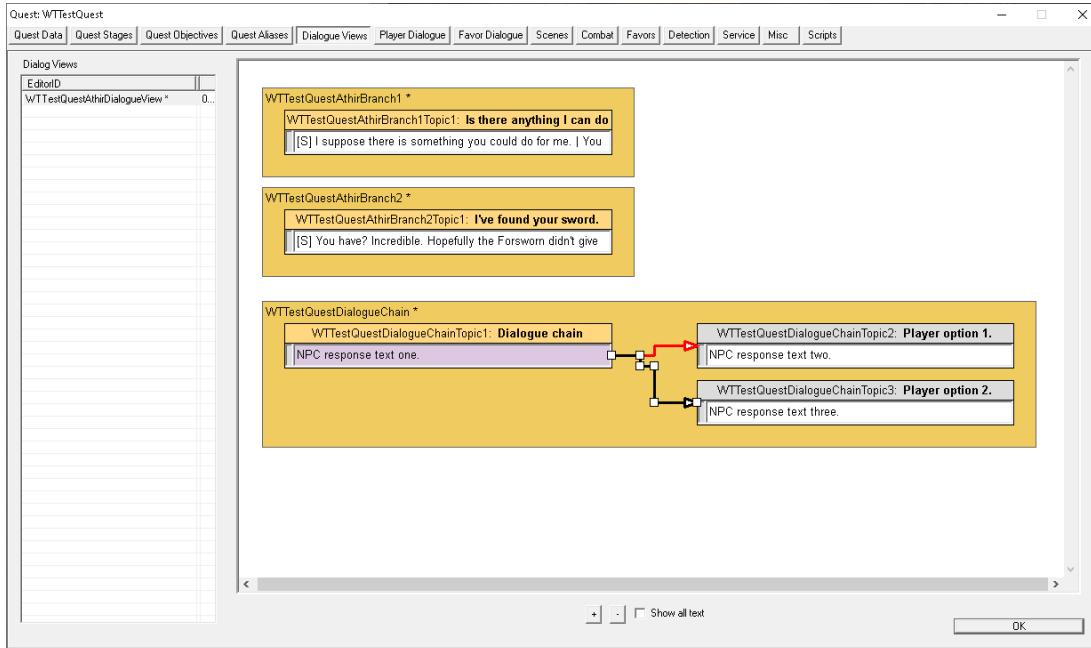


Figure 746 - Walk Away shown in Dialogue Views.

Another thing we can do is set up an invisible continue. This allows us to link multiple topics together.

An invisible continue is represented in the dialogue view as a green line between topics.

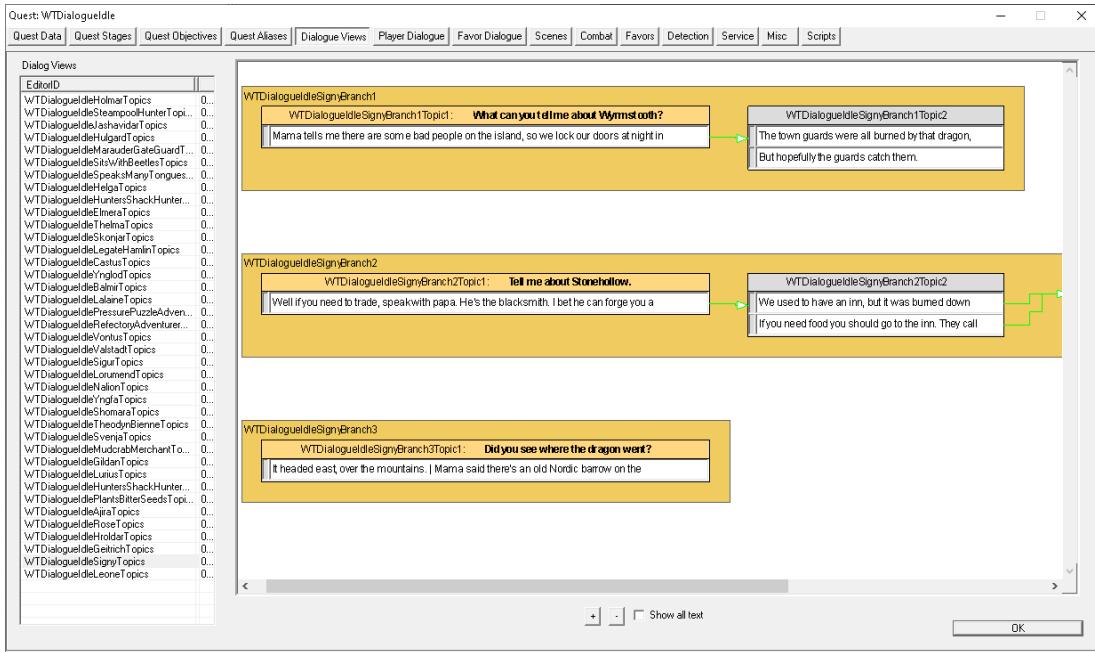


Figure 747 - Invisible Continue.

We can use invisible continues to break up a single big topic without needing to prompt the player to select a dialogue response in between. In the following example I used an invisible continue to change what an NPC says based on different conditions.

Open the first topic.

Tick ‘Invisible Continue’.

If you haven’t already linked the topics together, right-click in the dialogue list below it and select Add Link.

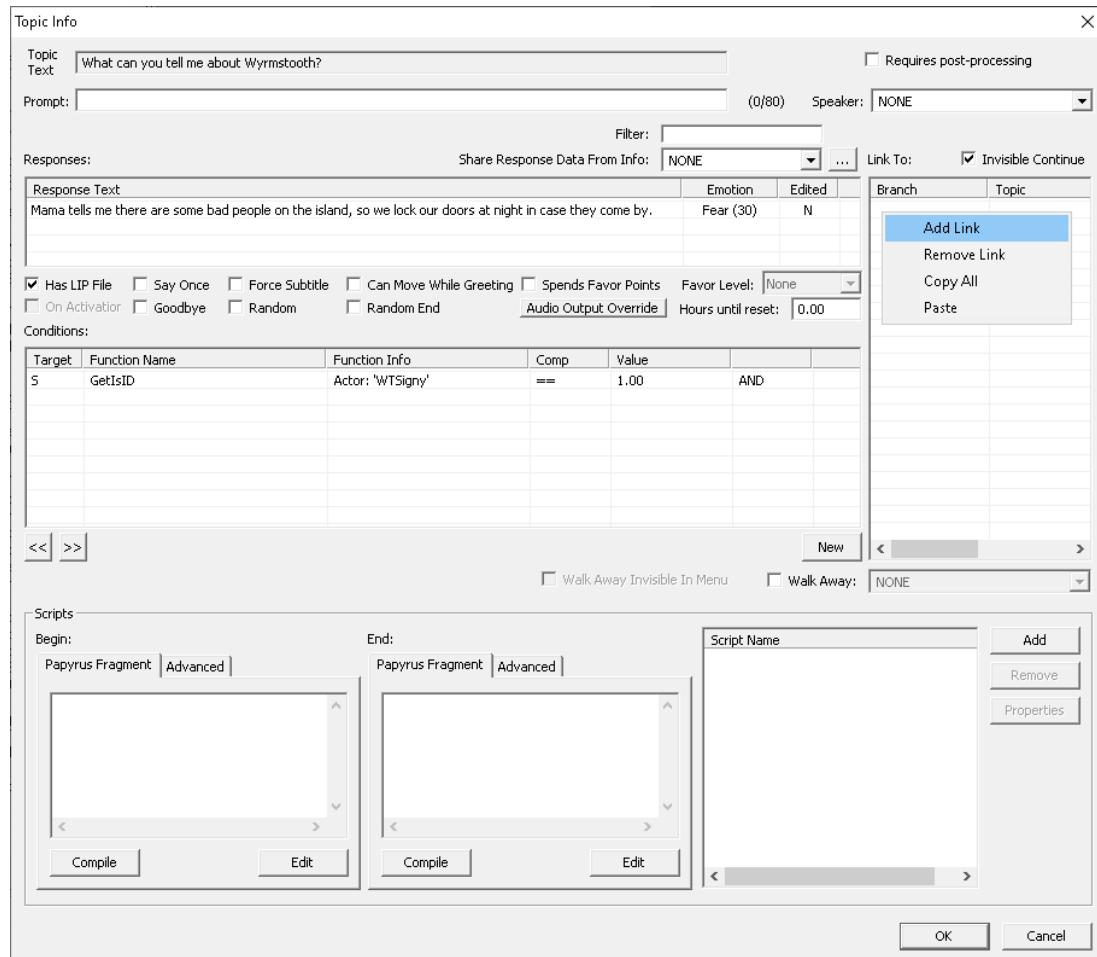


Figure 748 - Adding an invisible continue.

Select the topic you want to continue to then click OK.

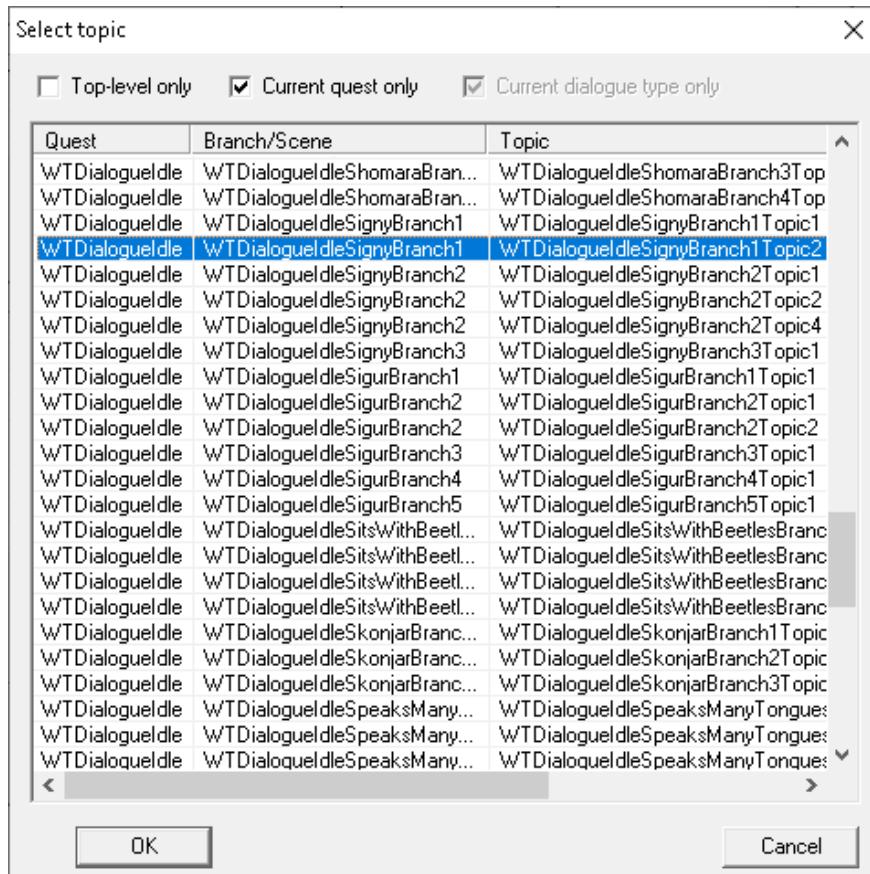


Figure 749 - Selecting the topic to continue to.

Confirm that the topic was added successfully then click OK to close out of Topic Info.

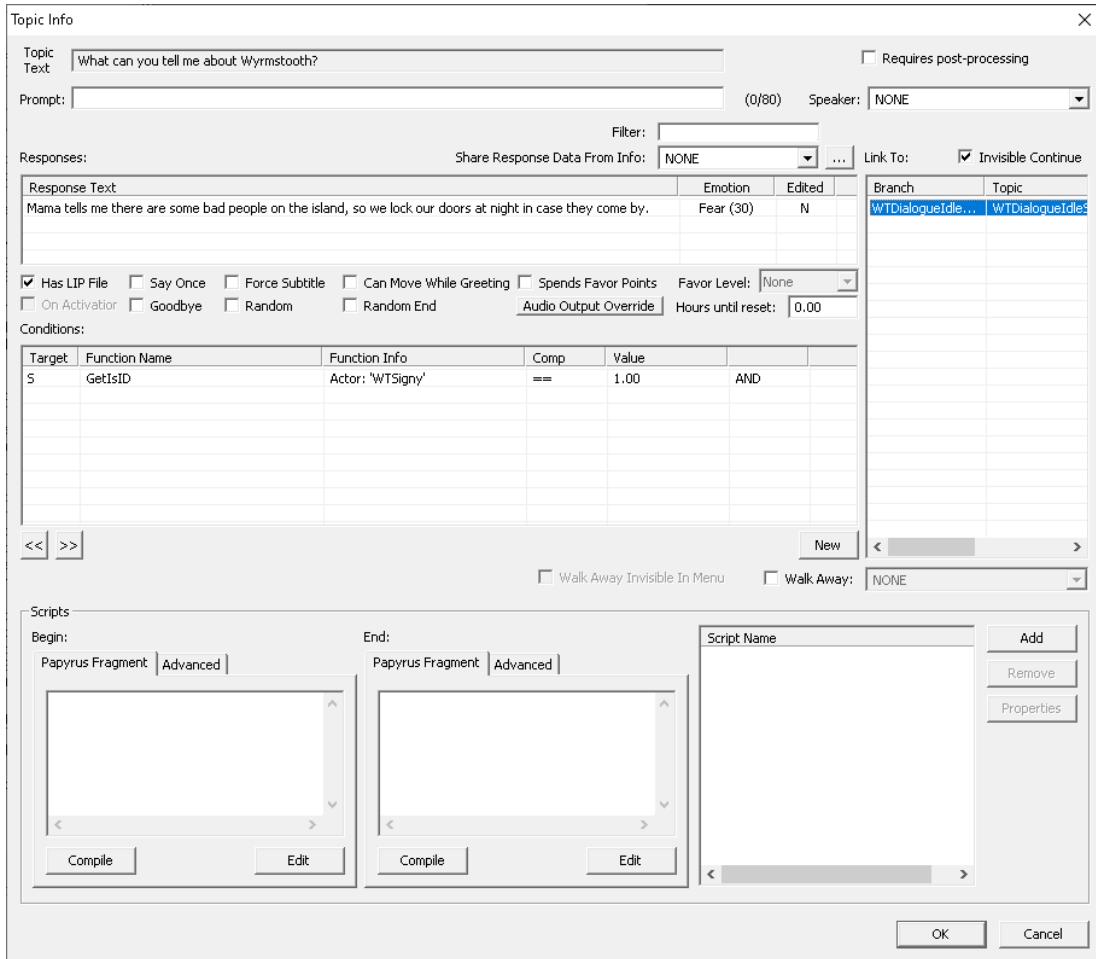


Figure 750 - Next topic linked to.

Ensure the Topic Text field is blank in the topic we've linked to.

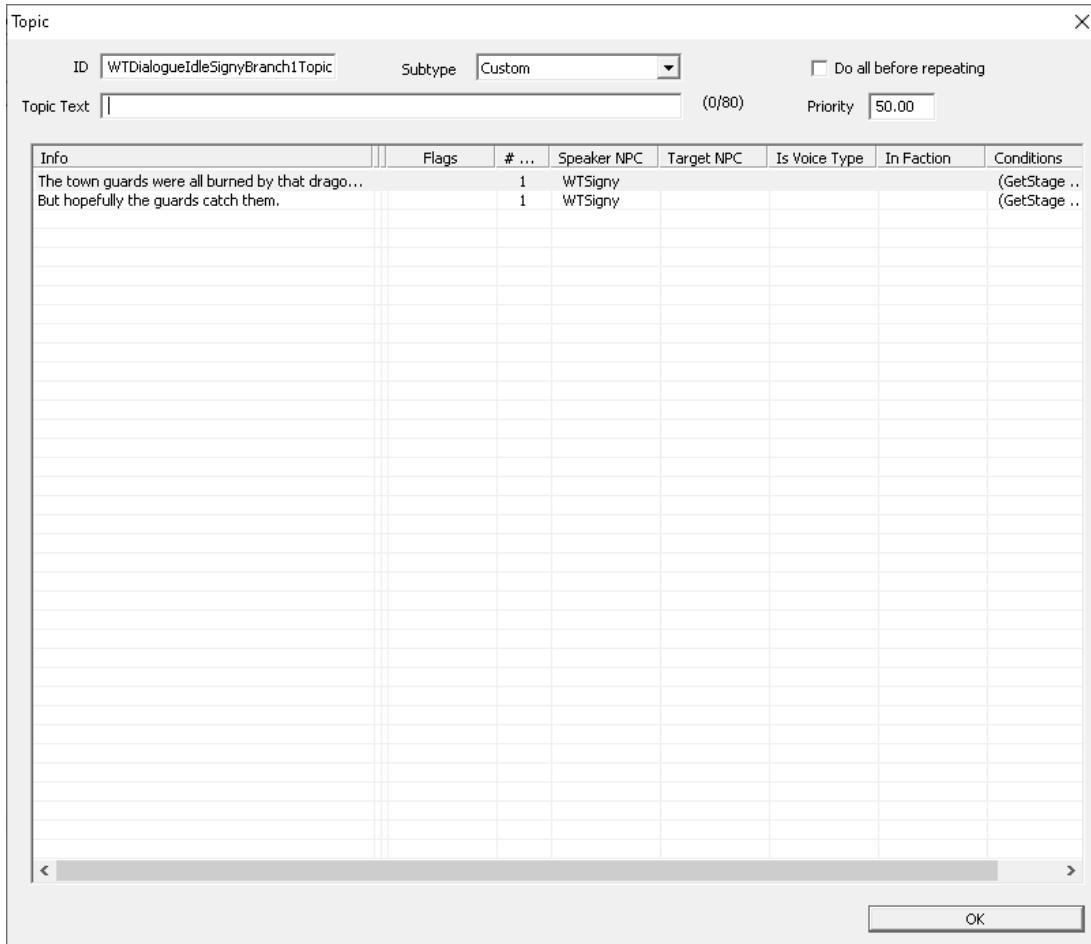


Figure 751 - Empty topic text.

So this topic contains two Topic Infos, allowing the NPC to say two different things based on our condition check.

If WTBarrowOfTheWurm hasn't been completed yet, Signy will follow on and end the topic chain with this line:

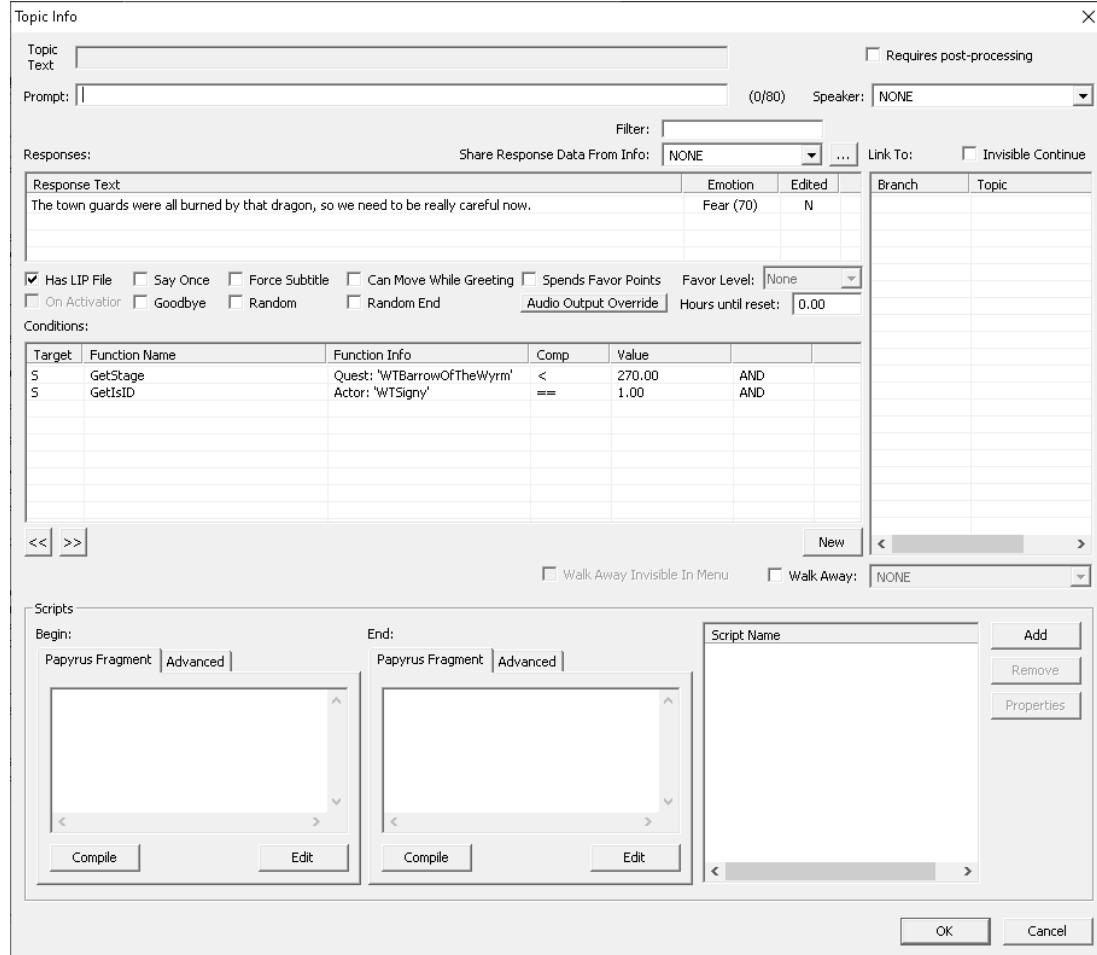


Figure 752 - First Topic Info in the linked topic.

Otherwise if WTBarrowOfTheWyrm has been completed, Signy will end the topic chain with this instead:

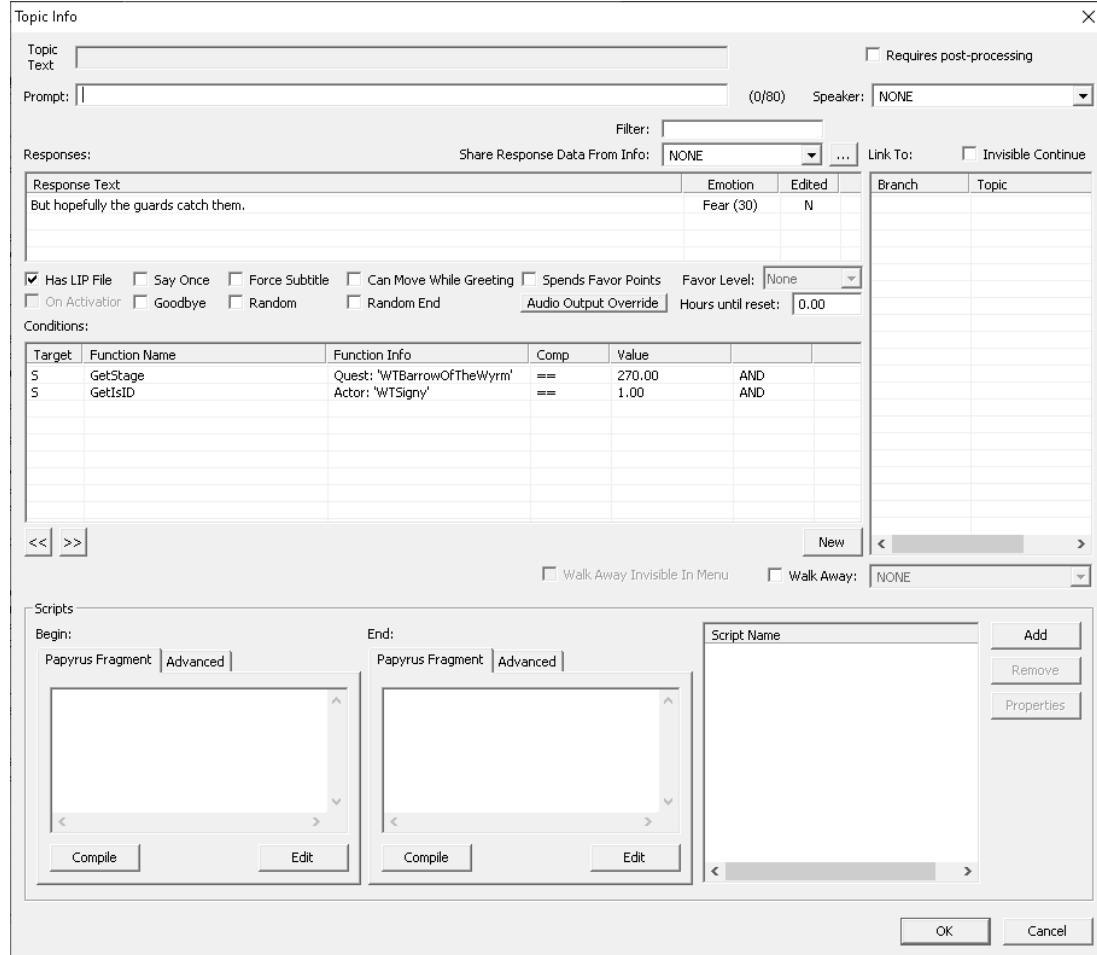


Figure 753 - Second Topic Info in the linked topic.

BLOCKING BRANCHES

If we wanted dialogue in a specific branch to be said by the NPC with priority over even the hello lines, we can set the branch up as a blocking branch.

Let's start by creating a new branch with a topic that we want the NPC to say first.

In the Dialogue Views tab, right-click in the blank area and select 'Create branch'.

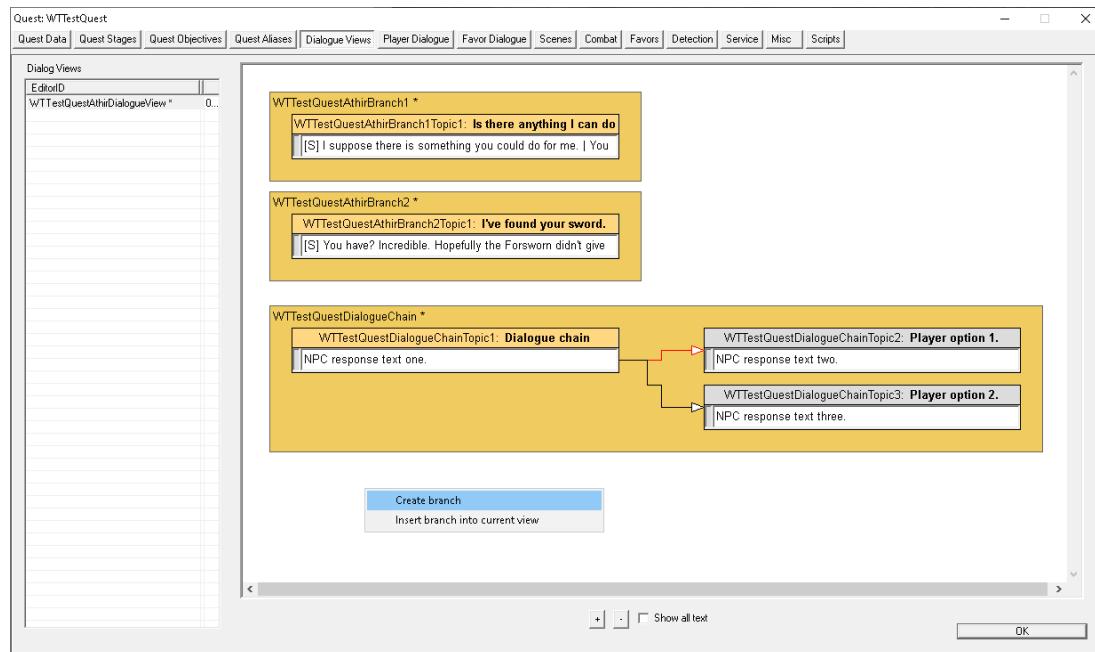


Figure 754 - Creating a new branch.

Enter in the ID of the new branch then click OK. For this example, I set it to WTTTestQuestBlockingBranch.

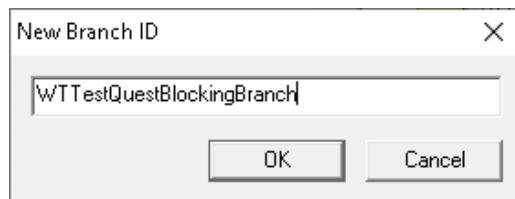


Figure 755 - Setting the ID of the new blocking branch.

Enter in the ID of the first topic in the new branch then click OK.

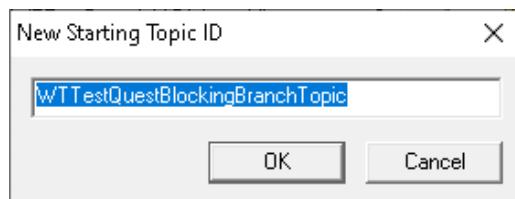


Figure 756 - Setting the ID of the new blocking branch's topic.

Double-click or right-click on the topic in the new branch and select 'Edit topic'.

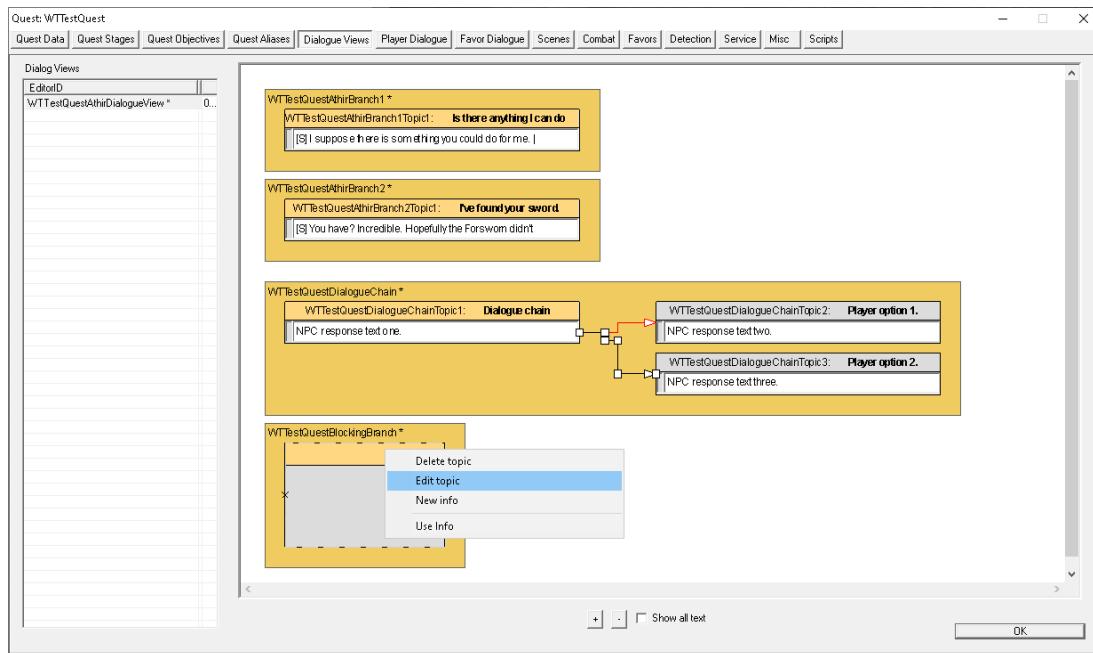


Figure 757 - Editing the topic in the new branch.

Don't add anything into the Topic Text field. Instead, just right-click in the Info list and select New.

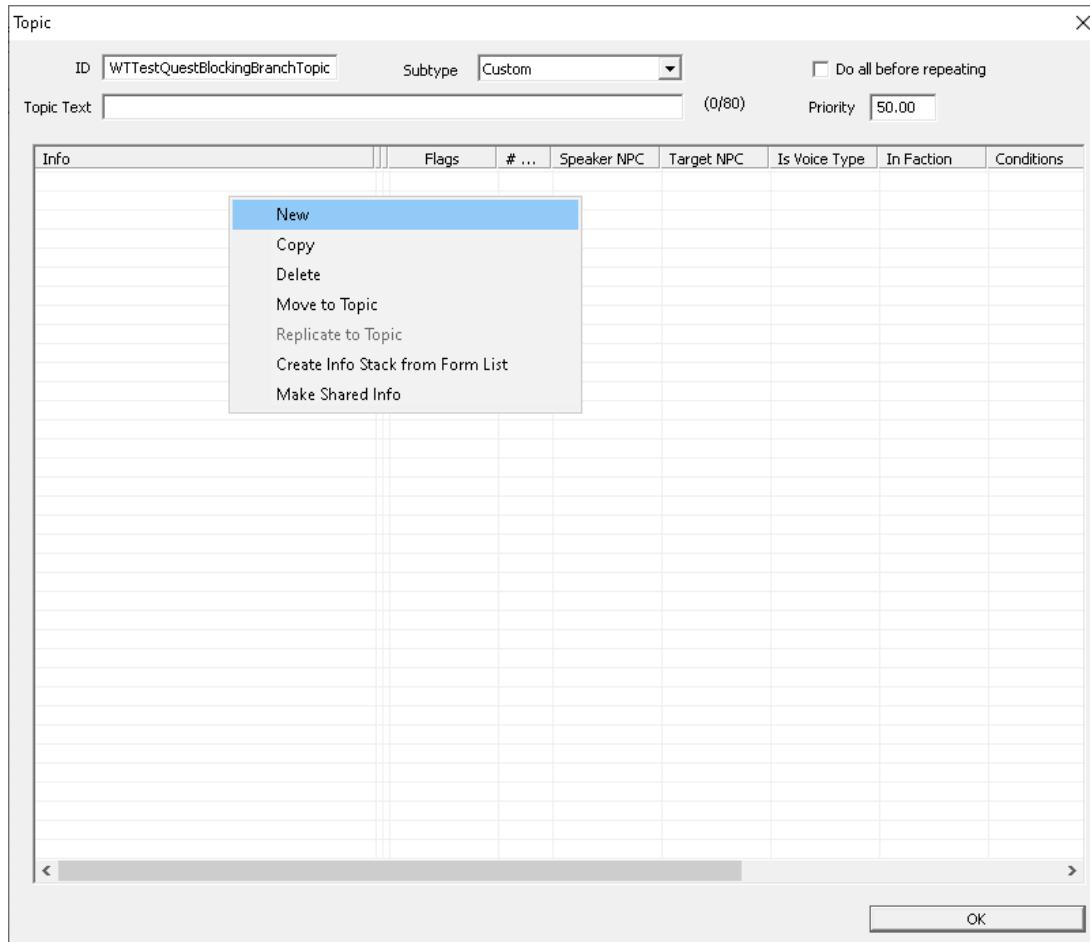


Figure 758 - Adding NPC response text.

Enter in the NPC's dialogue into the Response Text field then click OK.

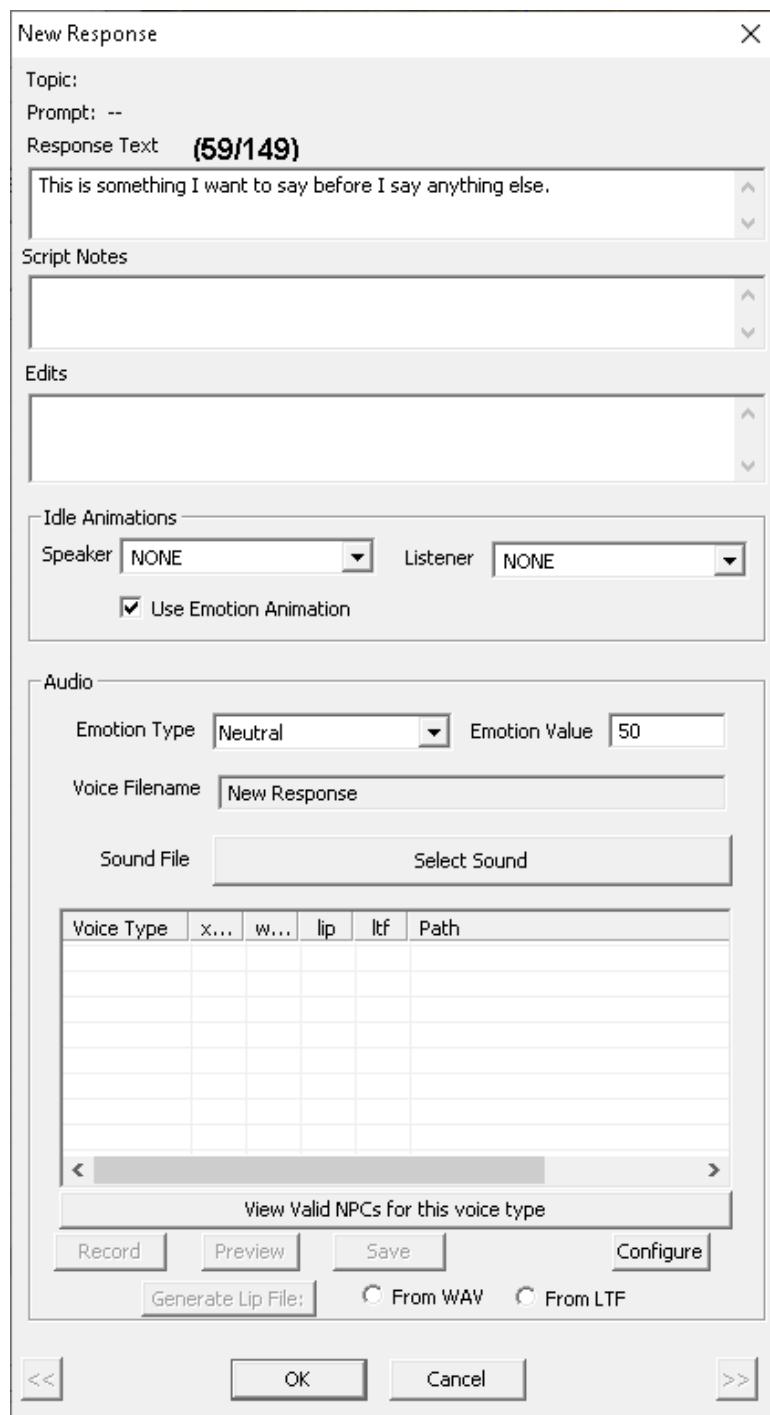


Figure 759 - Entering in the dialogue that we want the NPC.

Right-click in the Conditions list and select New.

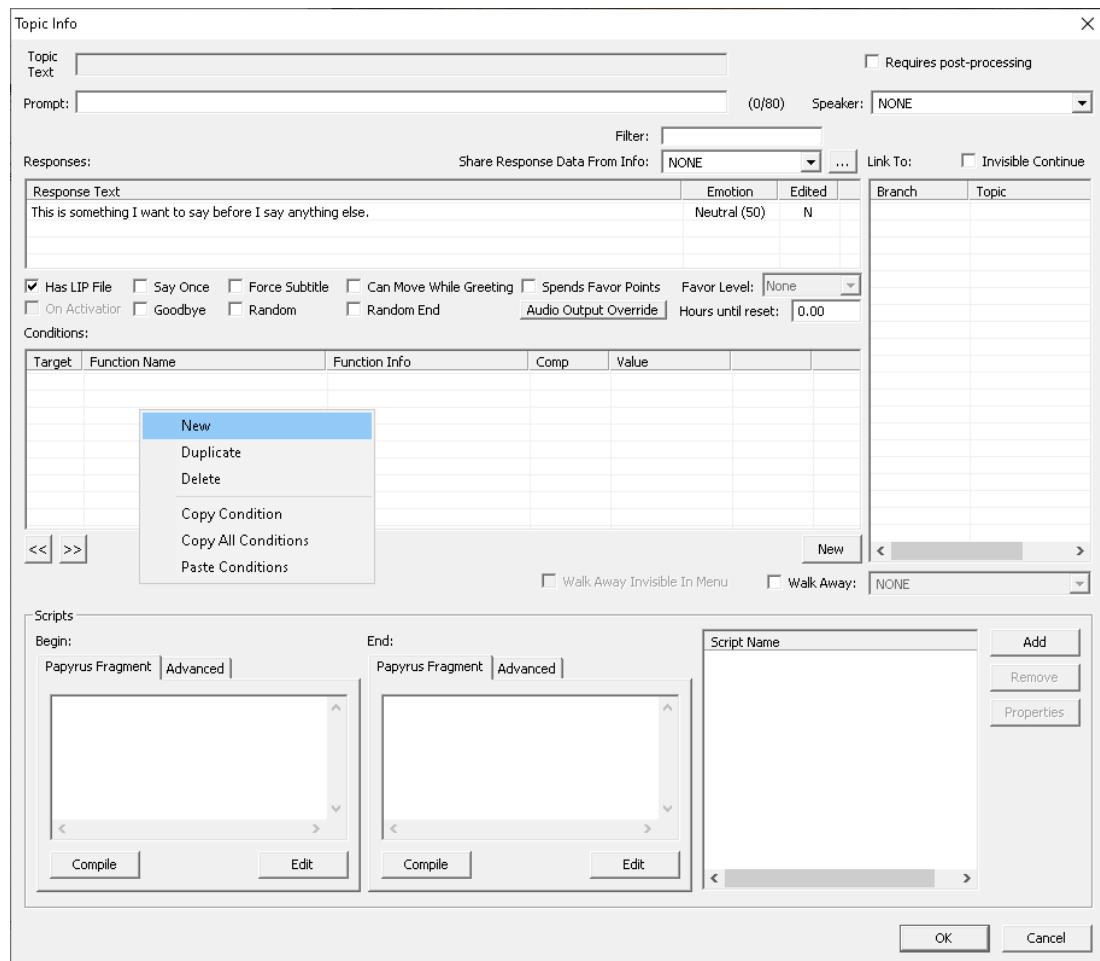


Figure 760 - Adding a new condition.

Ensure the Condition Function is set to GetIsID and set the actor. Again, for this example, that's going to be Athir.

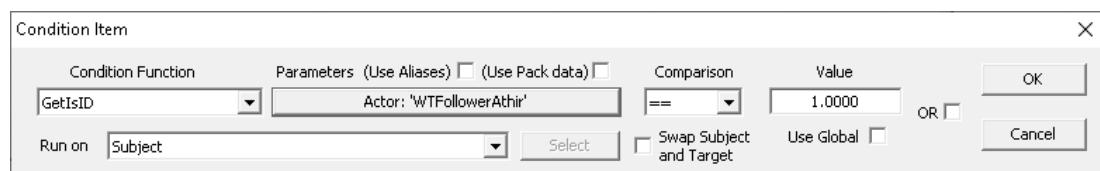


Figure 761 - Setting the GetIsID condition.

Click OK to close out of Condition Item.

Click OK to close out of Topic Info.

Click OK to close out of Topic.

Right-click in the branch and select 'Edit branch'.

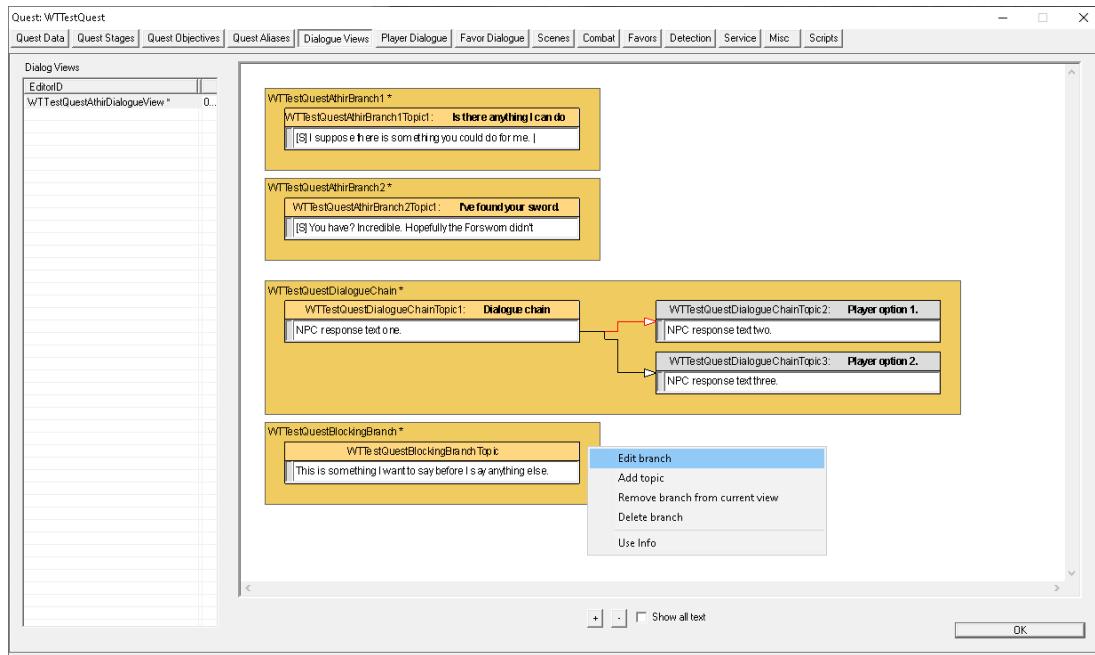


Figure 762 - Editing the branch properties.

Click on the Blocking radio button to set the branch up as a blocking branch, then click OK.

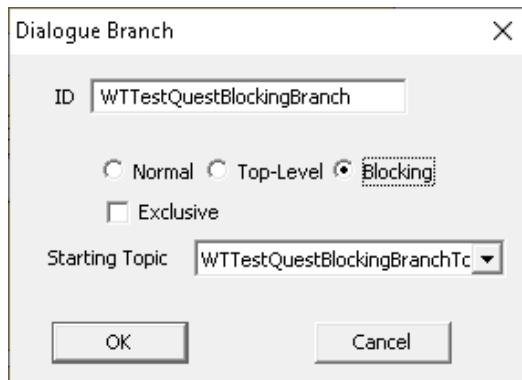


Figure 763 - Setting the branch from Top-Level to Blocking.

The colour of the branch should now change to green.

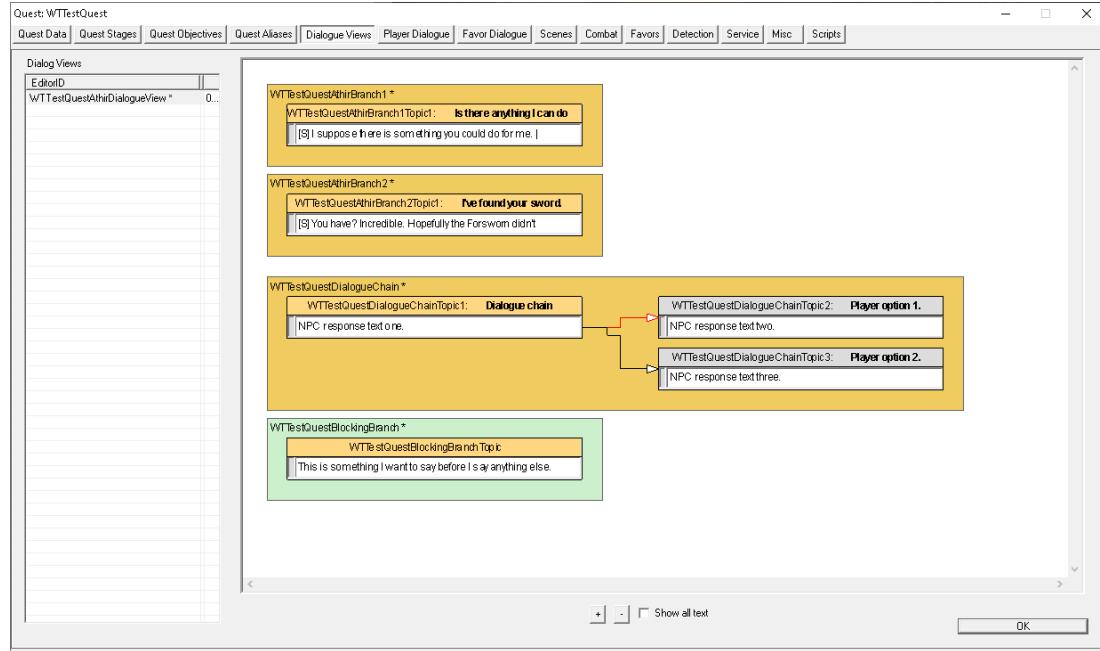


Figure 764 - The new branch set up as a blocking branch.

One caveat of using blocking branches is that they won't automatically flow on to the other branches we've defined, meaning we need to link them manually.

To make this easier, firstly move the blocking branch to the left of the other branches and topics.

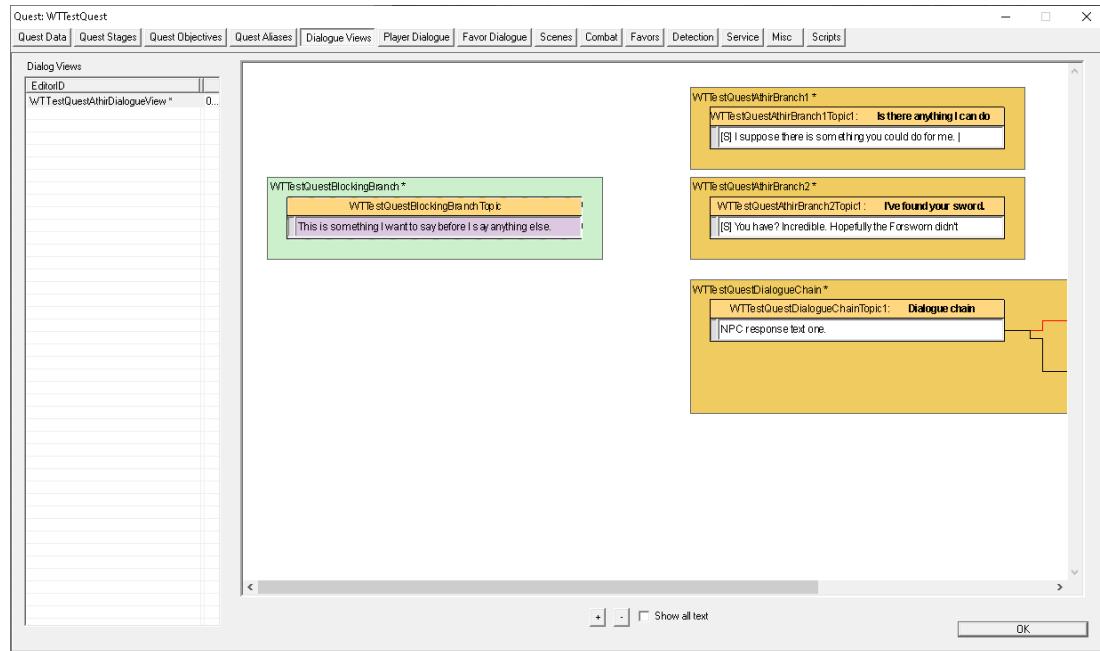


Figure 765 - Moving the blocking branch into position.

Left click on the topic response in the blocking branch's topic and drag and drop over to the other starting topics one at a time to link them all to the blocking branch's topic.

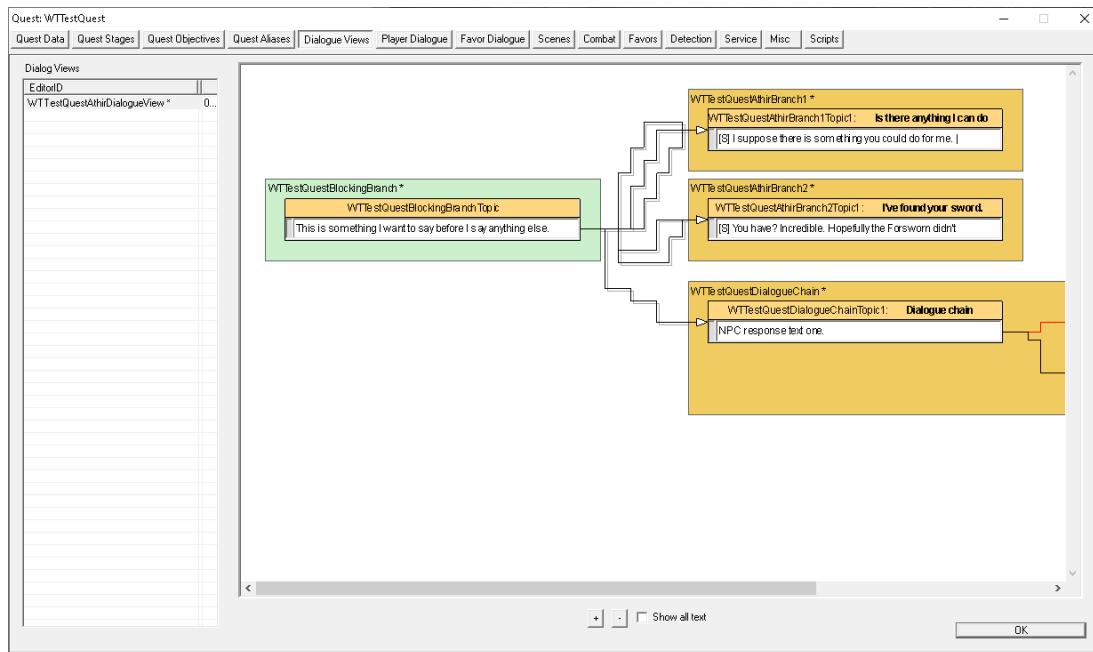


Figure 766 - The topic in the blocking branch linked to the other topics.

Note: The topic in the blocking branch does have priority over hello's, so you might want to add an extra condition to only use it in specific situations such as on a specific quest stage.

In the example below, I added a GetStage condition to ensure the topic in the blocking branch is only used when the current stage of WTTTestQuest is 0.

Alternatively, if it's something you only want an NPC to only ever say once, tick the Say Once checkbox.

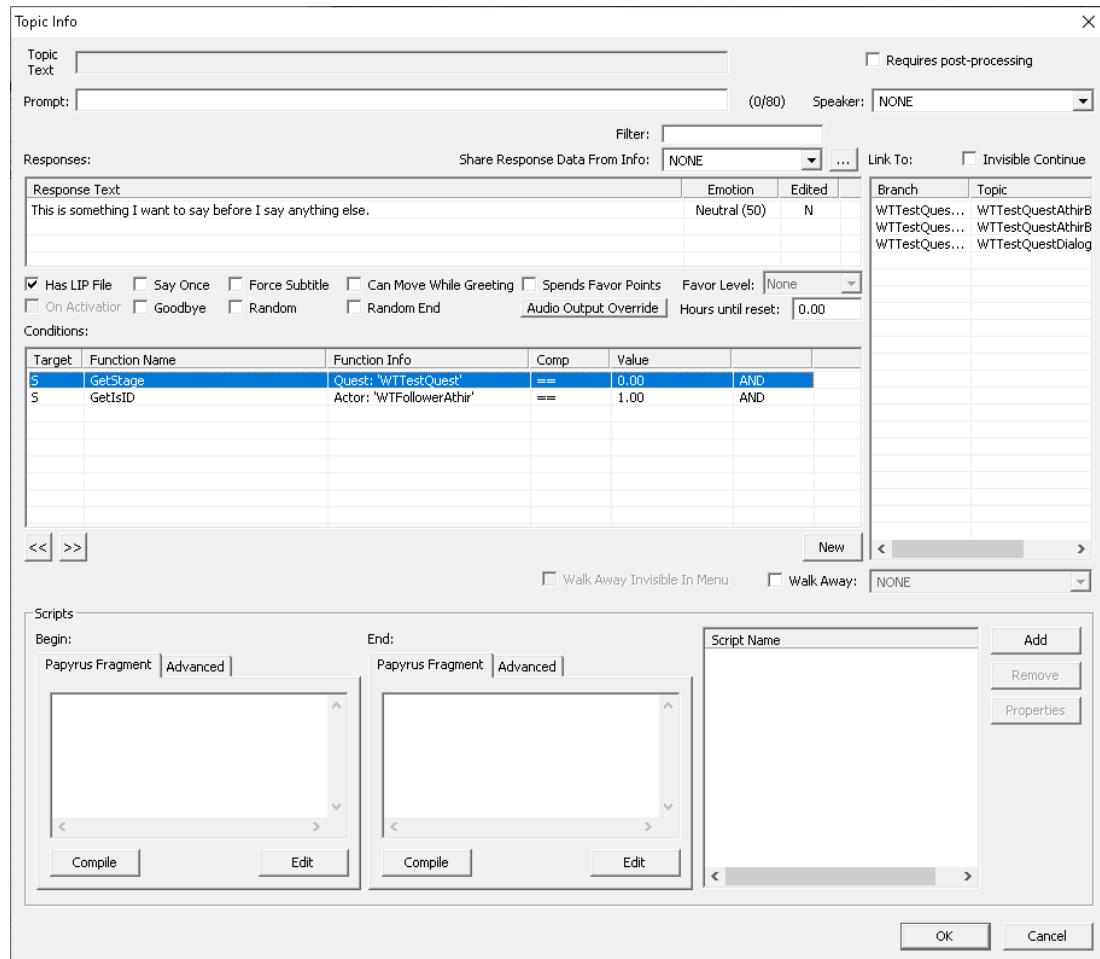


Figure 767 - Adding a getstage condition check to the blocking branch's topic.

GENERATING AN .SEQ FILE

Before we can test out our dialogue in-game, we need to generate or update an SEQ file.

Important: Every time you add a new quest containing new dialogue, you will need to generate a new SEQ file.

Click on the Save button in the Creation Kit to save these changes before continuing.

Open xEdit.

Ensure your mod is selected then click OK to begin loading.

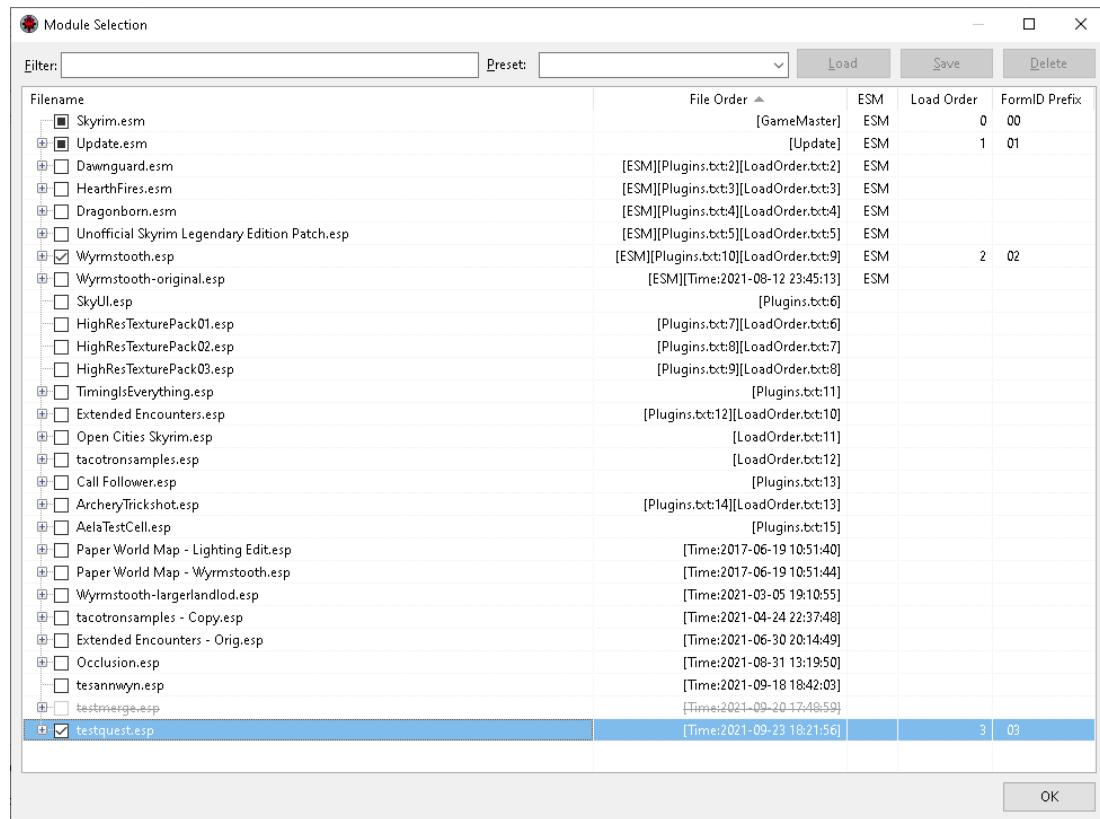


Figure 768 - Loading the mod in TES5Edit.

Right-click on your mod and select Other > Create SEQ File.

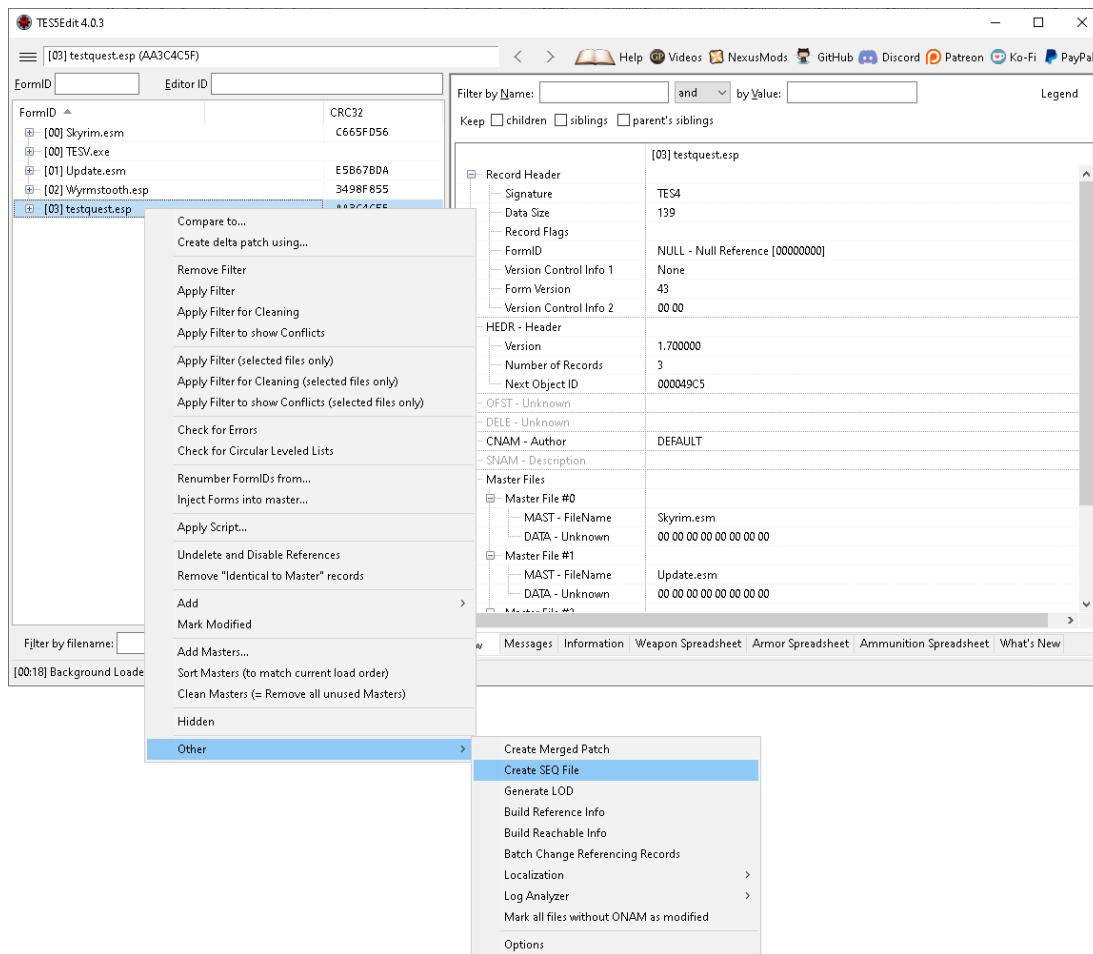


Figure 769 - Creating a new SEQ file.

This process should complete pretty quickly. Once it's done, you should see '[Create SEQ file done]' appear in the bottom left.

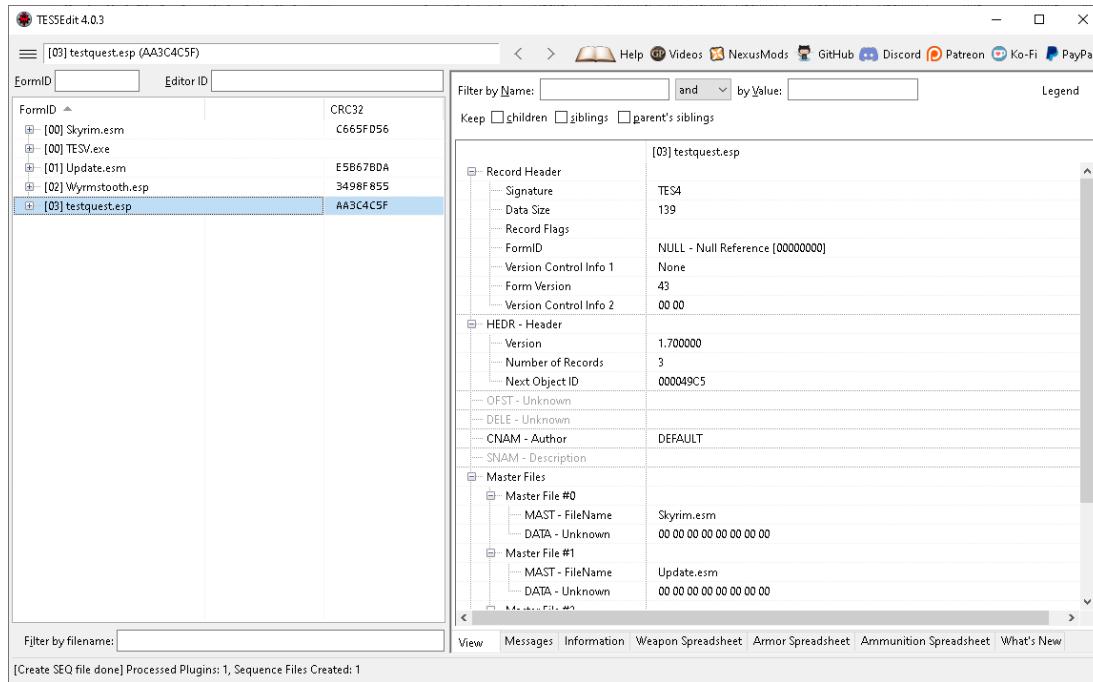


Figure 770 - SEQ file created.

To confirm the SEQ file has indeed been created, navigate to the Skyrim\Data\SEQ or Skyrim Special Edition Data\SEQ folder.

You should see an .seq file here with the same name as your .esp file.

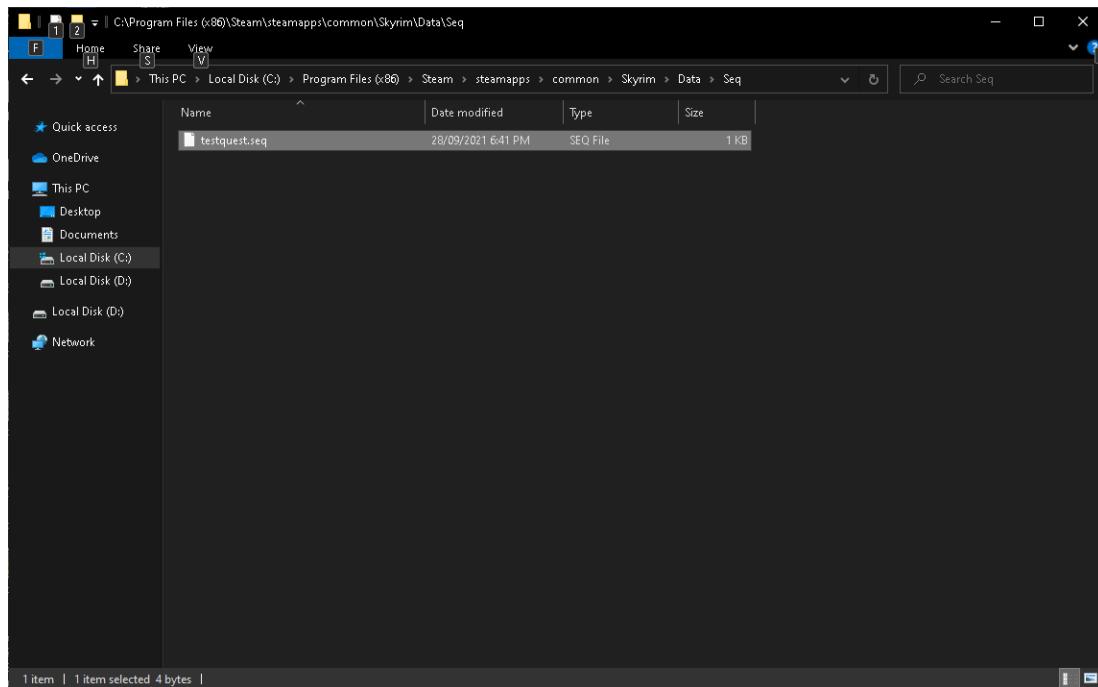


Figure 771 - SEQ file exists.

In the next section, I'll be covering the process of recording voice acting for your dialogue and generating .lip files for NPC lip-synching.

ADDING VOICE ACTING AND LIP SYNCHING

Now that we have some dialogue in our test quest, we'll need to add voice acting and lip synching.

Important: Voice acting needs to be recorded at 44.1 KHz and saved in the .wav format so we can process them later with Unfuzer.

Before passing your dialogue over to a voice actor, you will need to set up a spreadsheet containing a list of all lines spoken by an NPC. This spreadsheet should include notes for the voice actor to follow to help guide their performance.

The screenshot shows an Excel spreadsheet titled "Thelma_Dialogue.xlsx". The spreadsheet contains a list of dialogue entries for NPC Thelma, organized into columns for ID, Dialogue, and Comment. The ID column lists entries such as THLM_022 through THLM_042. The Dialogue column contains the actual lines of dialogue, and the Comment column contains notes for the voice actor, such as "(Player is naked)" and "(dialogue in quotes should sound sarcastic)". The spreadsheet is in "Normal" view with a green header bar.

ID	Dialogue	Comment
THLM_022	By the gods, cover yourself up you pustulent churl!	
THLM_023	If you need to steal about completely naked, do so elsewhere! This is a house of commerce!	(Player is naked)
THLM_024	Disgusting! Your parents should be ashamed of you.	
THLM_025	If you can't behave yourself, then get out!	
THLM_026	Watch it, you clodhopper!	(Player runs into Thelma)
THLM_027	Are you drunk? Get out of here!	
THLM_028	Didn't your mother teach you not to run around indoors? What have I done to deserve this mistreatment...	
THLM_029	Put that down, now!	
THLM_030	What are you doing? Making another mess, hmm?	(Player is messing around)
THLM_031	Let me say this slowly so that you can understand: stop...making...a...mess!	
THLM_032	Stop touching that!	
THLM_033	I'm supposed to be the cleaner! But somebody has to step up and maintain order when Lunrus is absent.	
THLM_034	You'd struggle to gather at least half a wit among the miners. Hmpf	
THLM_035	Is there anything you need help with?	
THLM_036	Yes, in fact there is something you can help me with, if you care to hear me out.	
THLM_037	A few moments ago an ignorant lout walked into the tradehouse and started pestering me, asking me all sorts of bothersome questions.	
THLM_038	Tell me about Wyrmsooth, 'What do you do here', it's an unbearable burden on these tired old ears.	(dialogue in quotes should sound sarcastic)
THLM_039	So, I want you to do something about it: remove this individual from the tradehouse by what ever means necessary.	
THLM_040	(Go on. Write it in your journal. Good! Well? What are you waiting for? I've given you a quest, now get to it!	
THLM_041	I'll... be going.	
THLM_042	Congratulations. You're halfway to becoming a true hero of the people.	Tell me about Wyrmsooth
THLM_043	If it wasn't for the coin, I'd have taken the first ship back to Windhelm. That's the kind of place Wyrmsooth is.	
THLM_044	Get out of here!	

Figure 772 - Sample dialogue spreadsheet.

Once you've received the voice acting from your voice actor, browse to the `Skyrim\Data\Sound\voice` or `Skyrim Special Edition\Data\Sound\voice` folder.

Create a new folder named after your mod. For this test quest example, that'll be 'testquest.esp'.

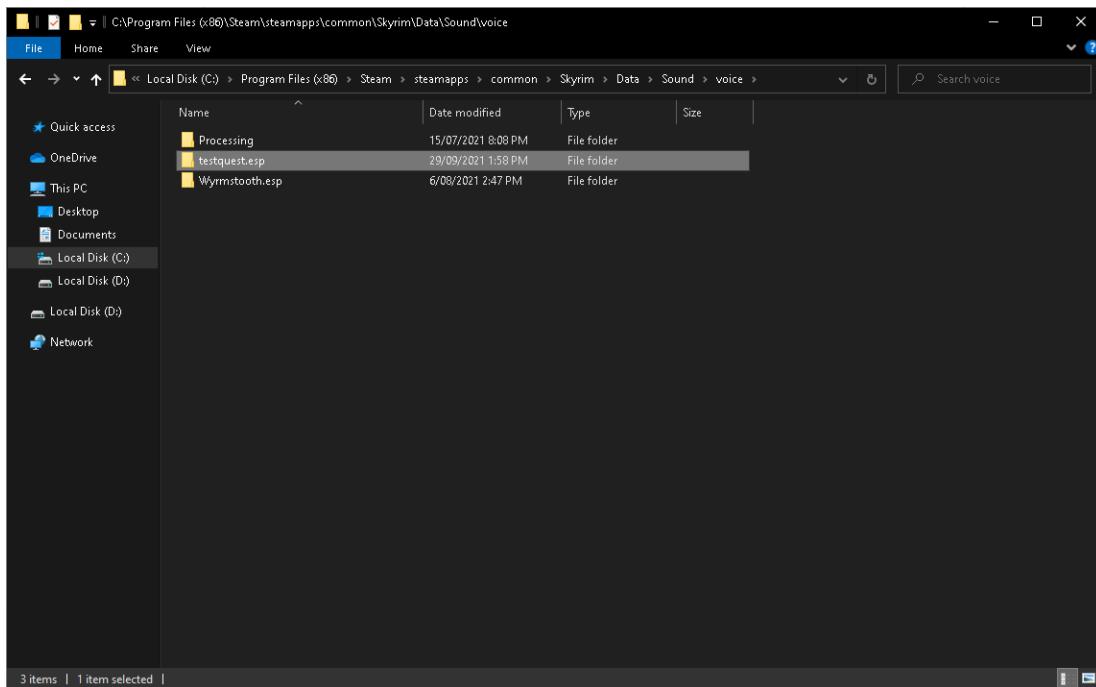


Figure 773 - Voice Type folders.

Go into that folder and create a new folder named after the voice type assigned to the NPC.

For example, if the NPC uses the MaleNord voice type, name that folder MaleNord.

In my example, Athir is from the Wyrmstooth mod and uses the WTVoiceType voice type, so I created a folder called 'WTVoiceType'.

Note: Ideally, you should be using a custom voice type for NPCs that have their own voice acting. Players may find it a bit jarring if an NPC's voice actor keeps changing.

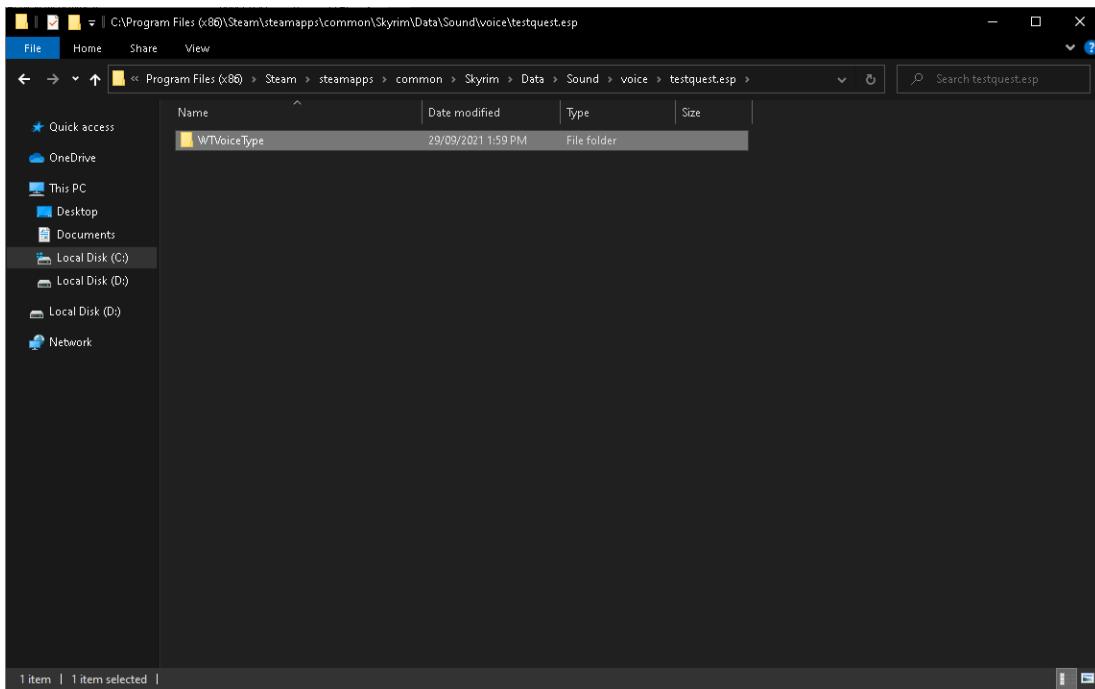


Figure 774 - Creating a voice type folder.

Copy your voice acting files into that folder.

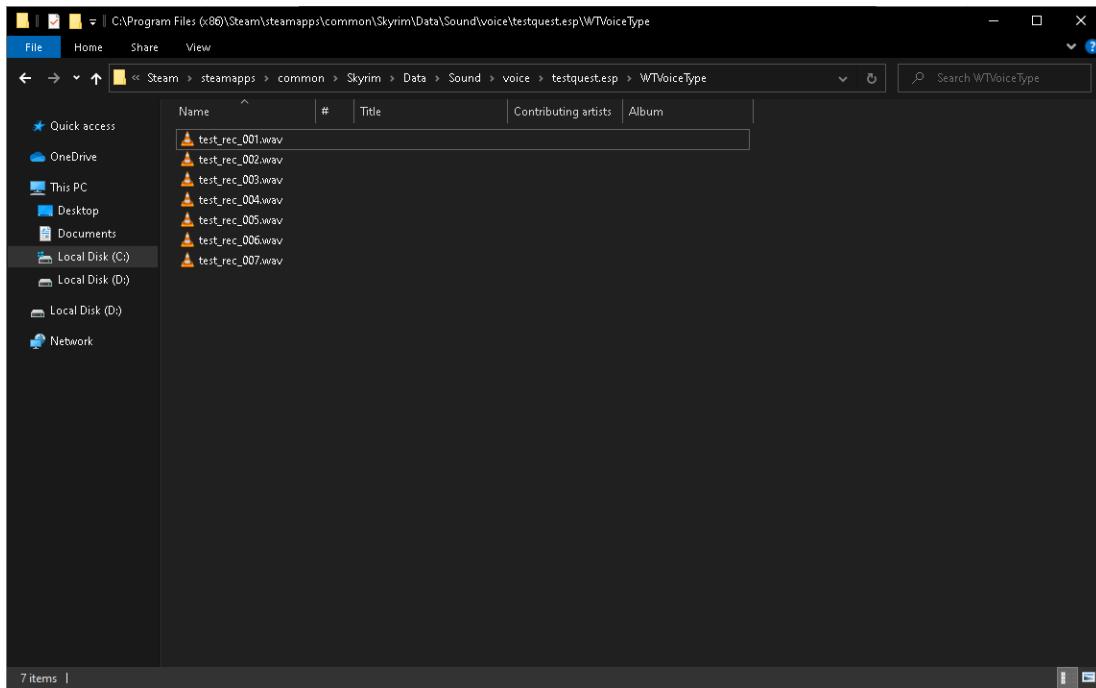


Figure 775 - Folder containing voice acting files.

Next, we need to rename these .wav files to match the internal naming scheme.

In the Quest properties, go to the Dialogue Views tab and double-click on the response text below a topic to open it.

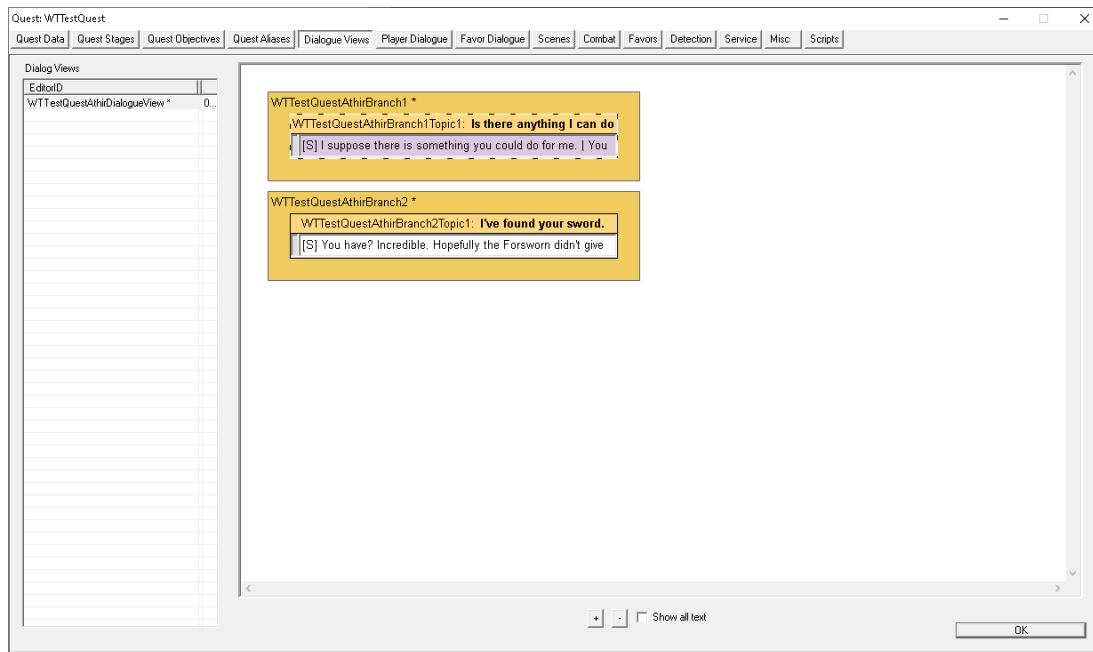


Figure 776 - Dialogue Views tab.

Double-click on the first response to open its properties.

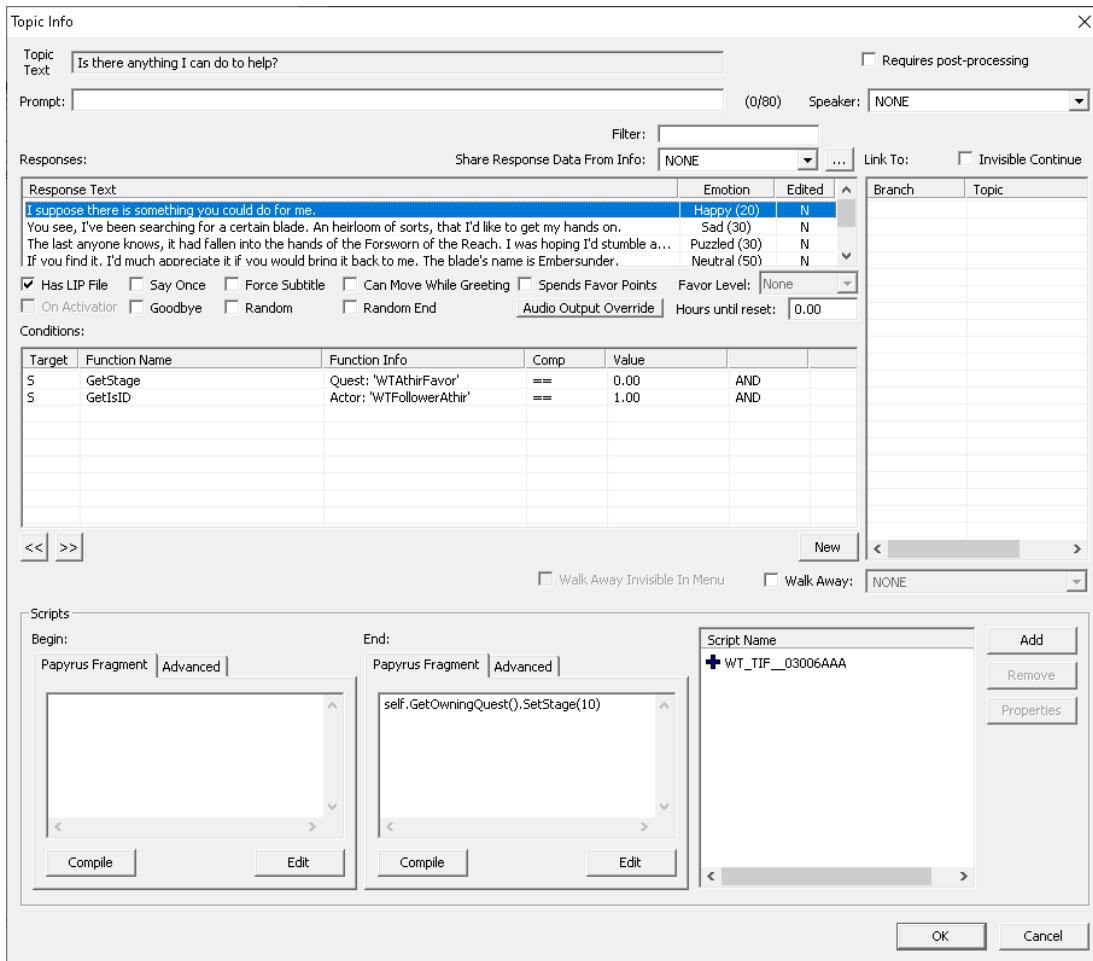


Figure 777 - Topic Info.

The file name for this line appears in the Voice Filename field. Select this string and press CTRL + C to copy it.

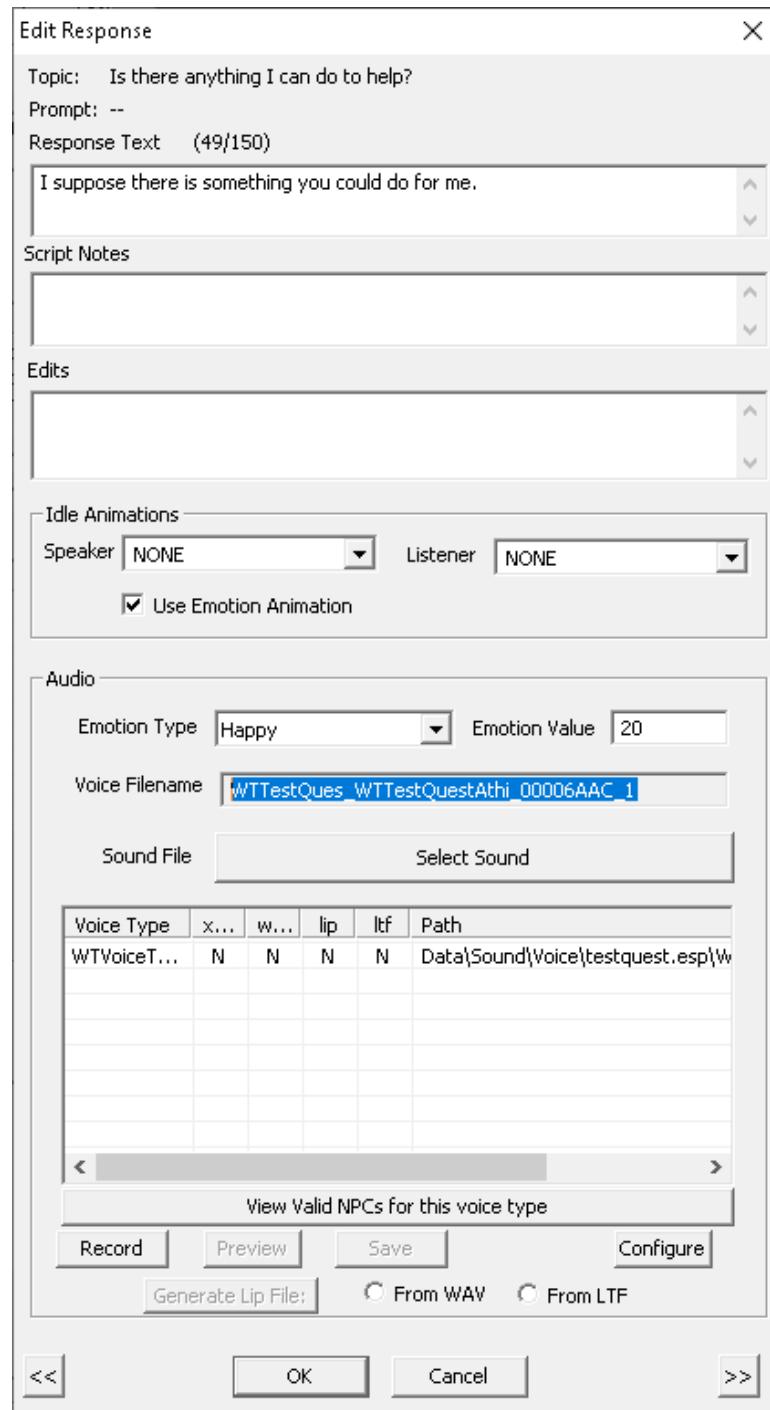


Figure 778 - Voice Filename.

Go to the folder containing the voice files and rename the corresponding .wav file for this line.

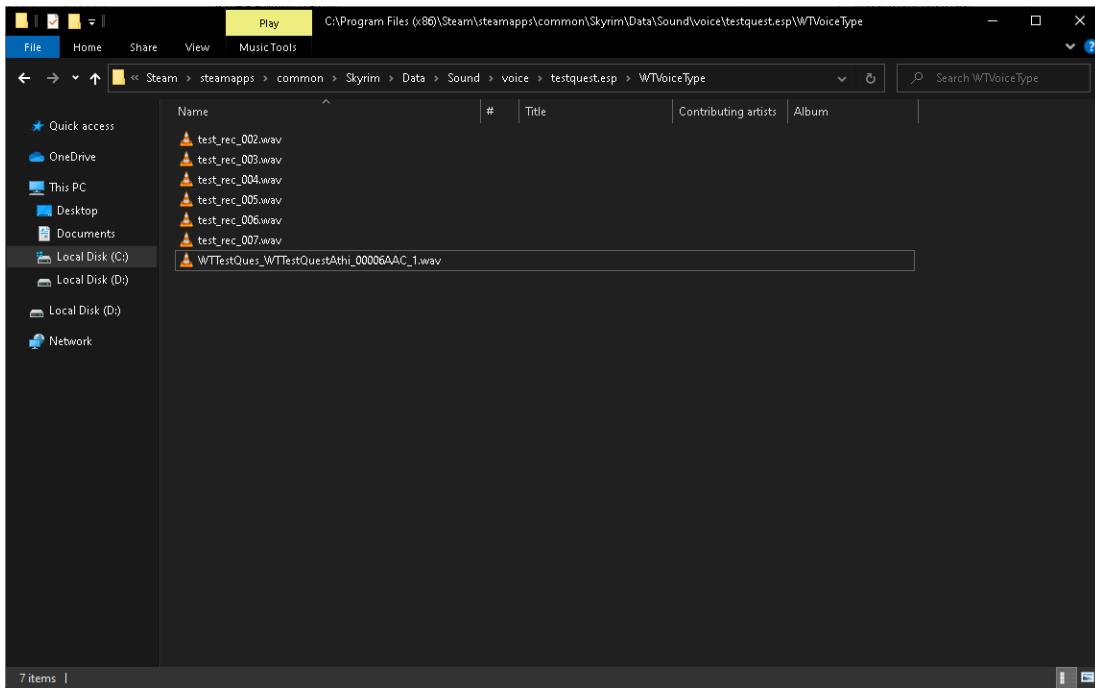


Figure 779 - Renamed the .wav file.

Repeat these steps to rename the rest of the .wav files for this topic.

The filenames are sequential, so if the Voice Filename for the first line is WTTTestQues_WTTTestQuestAthi_00006AAC_1, the second line will be WTTTestQues_WTTTestQuestAthi_00006AAC_2 then WTTTestQues_WTTTestQuestAthi_00006AAC_3 and so on.

Repeat these steps to rename the .wav files for the responses from the other topic too.

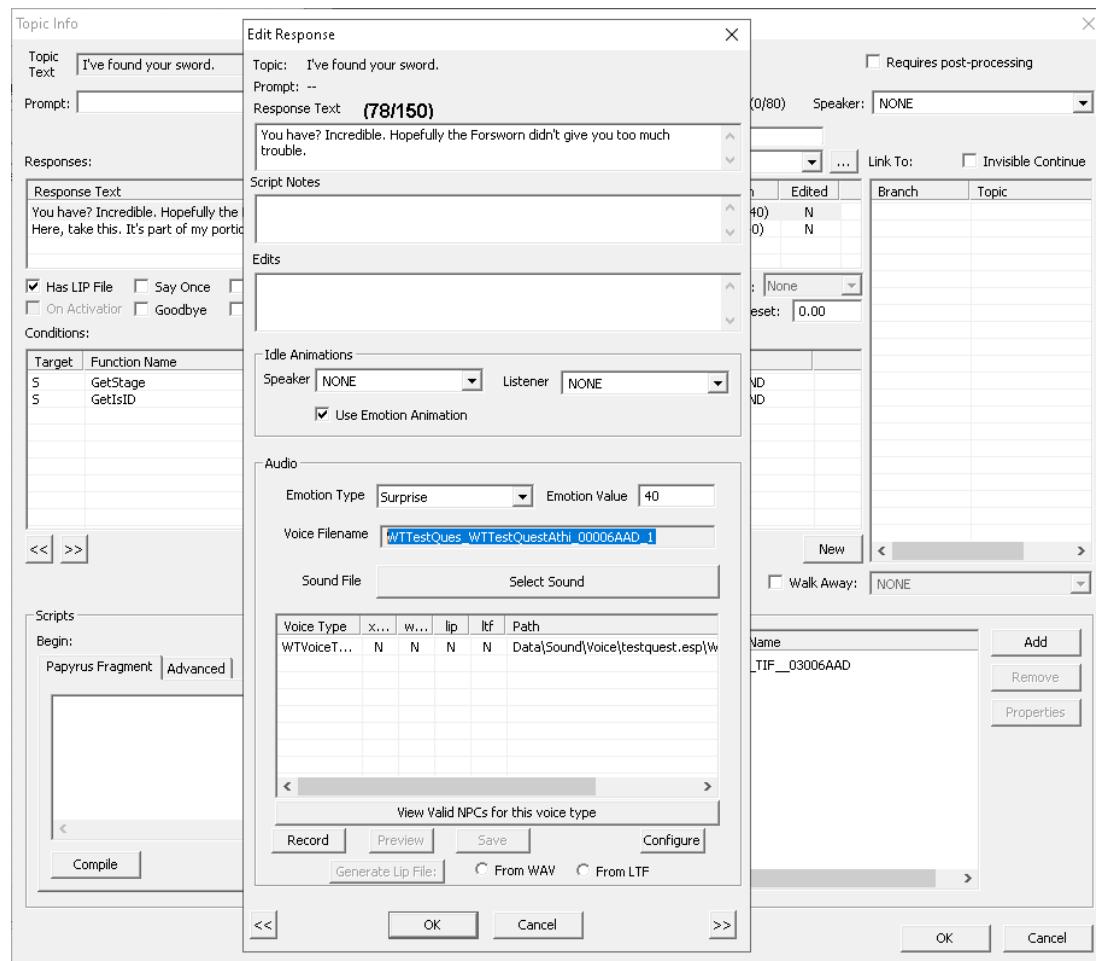


Figure 780 - Getting the Voice Filename for the other topic.

All the .wav files should now have been renamed like they have in the following screenshot:

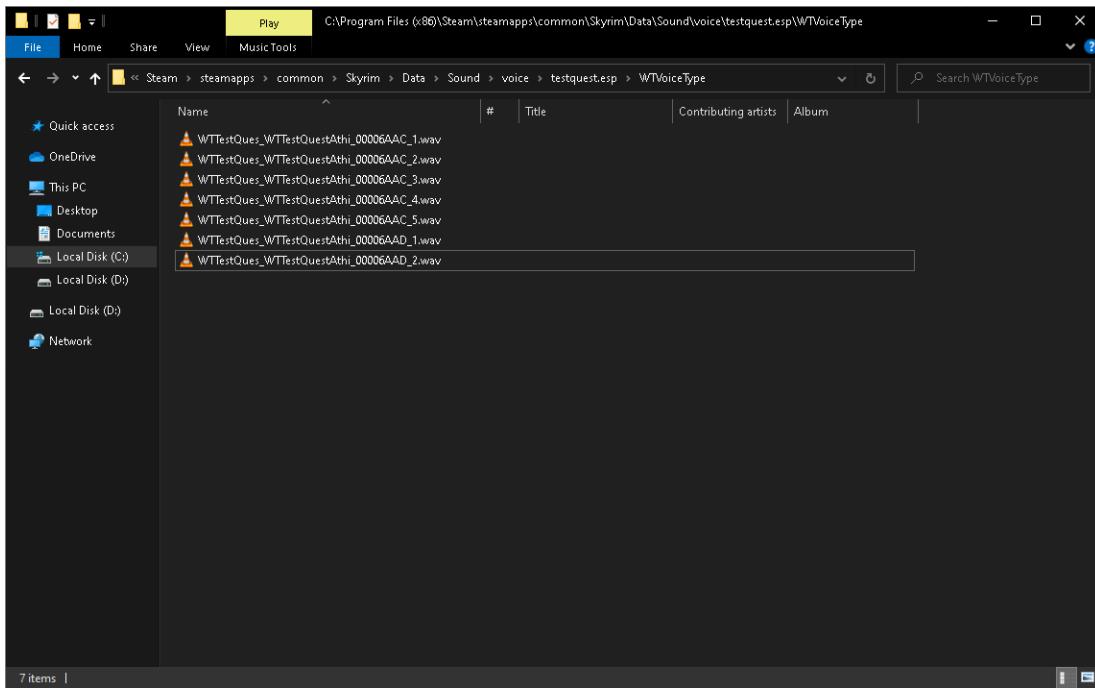


Figure 781 - .wav files renamed.

Next we need to generate the lip sync data. The lip sync data we generate will be stored in .lip files.

When the process is complete, we should have a .lip file for each .wav file in the voice type folder.

Go back to the first line of the first topic.

The ‘Generate Lip File:’ button will be greyed out until you select the .wav file in the list above. Once selected, the ‘Generate Lip File:’ button should un-grey itself if the corresponding .wav file can be found in the voice type folder.

Click on the ‘Generate Lip File:’ button to generate the .lip file.

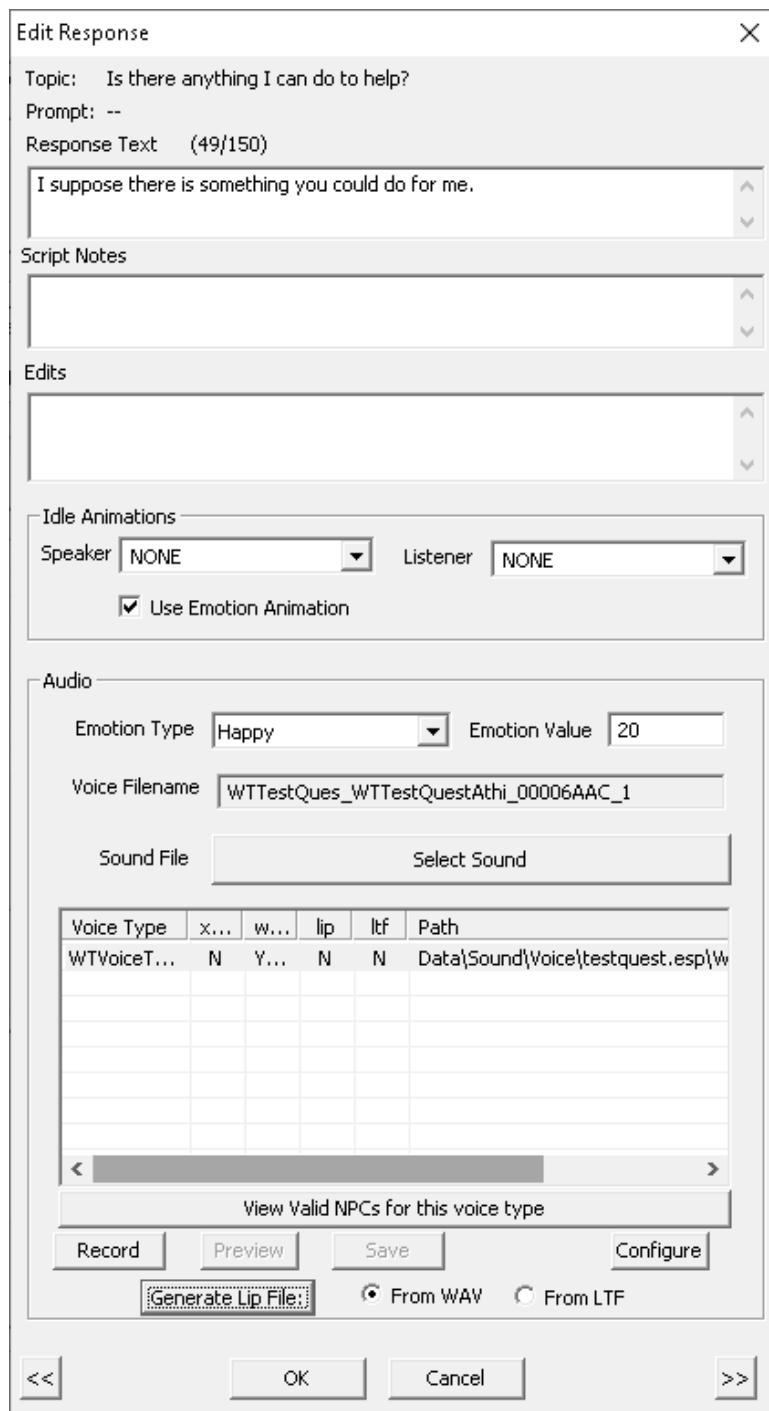


Figure 782 - Generating a .lip file.

A .lip file should appear in the voice type folder.

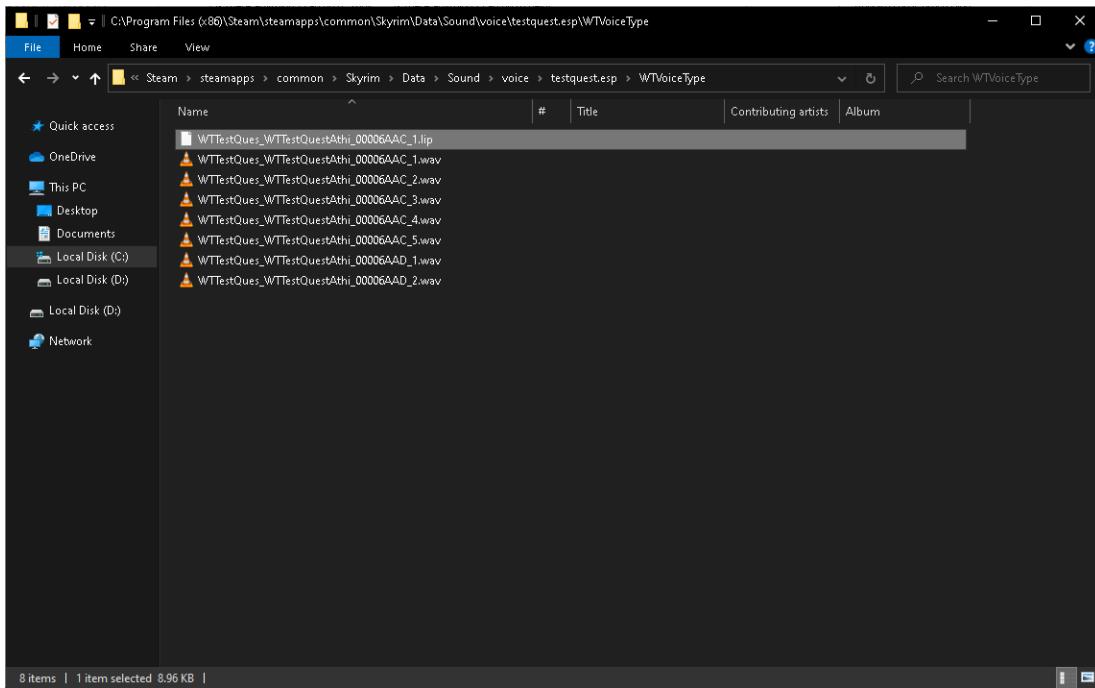


Figure 783 - .lip file created successfully.

Repeat these steps to generate a .lip file for the other responses in this topic.

Repeat these steps to generate .lip files for the responses in the other topic too.

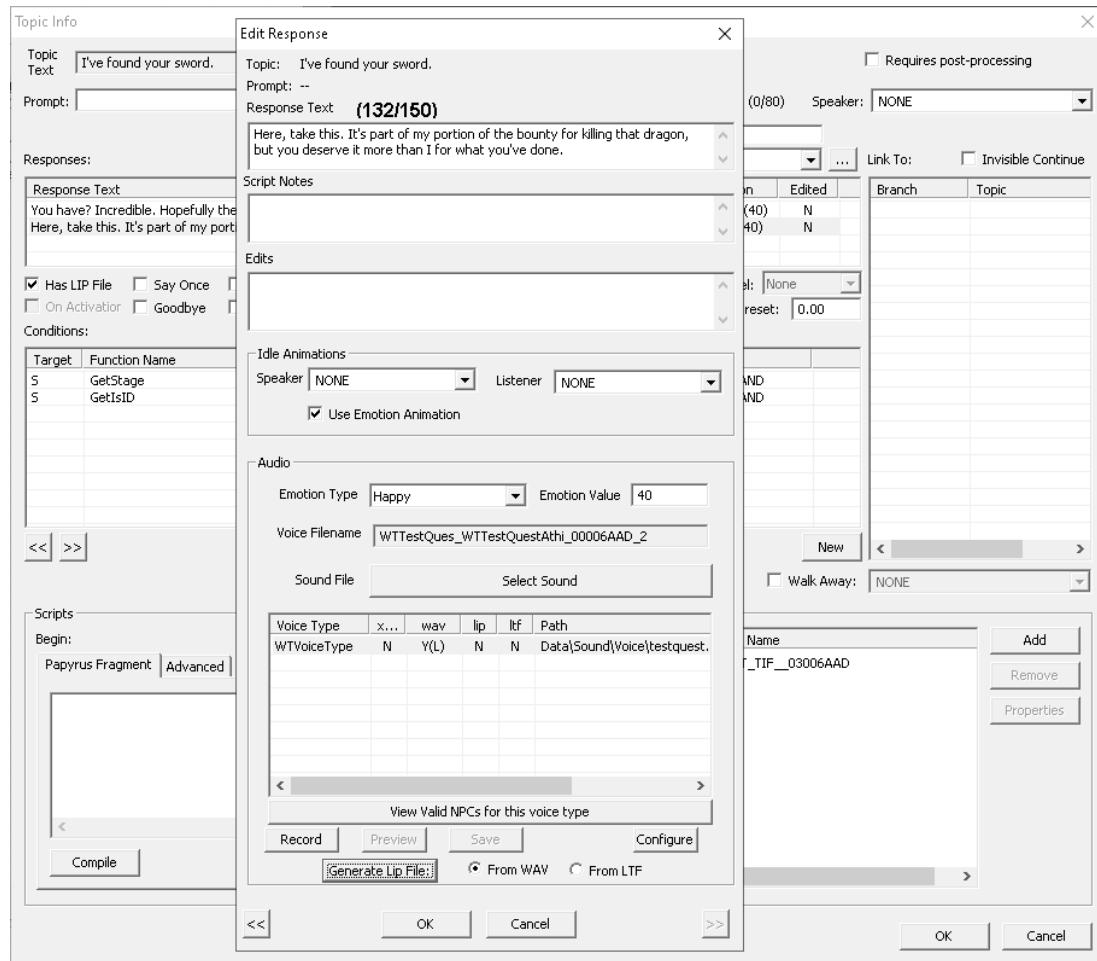


Figure 784 - Generating .lip files for the responses in the second topic.

We should now have a .lip file for each .wav file in the voice type folder.

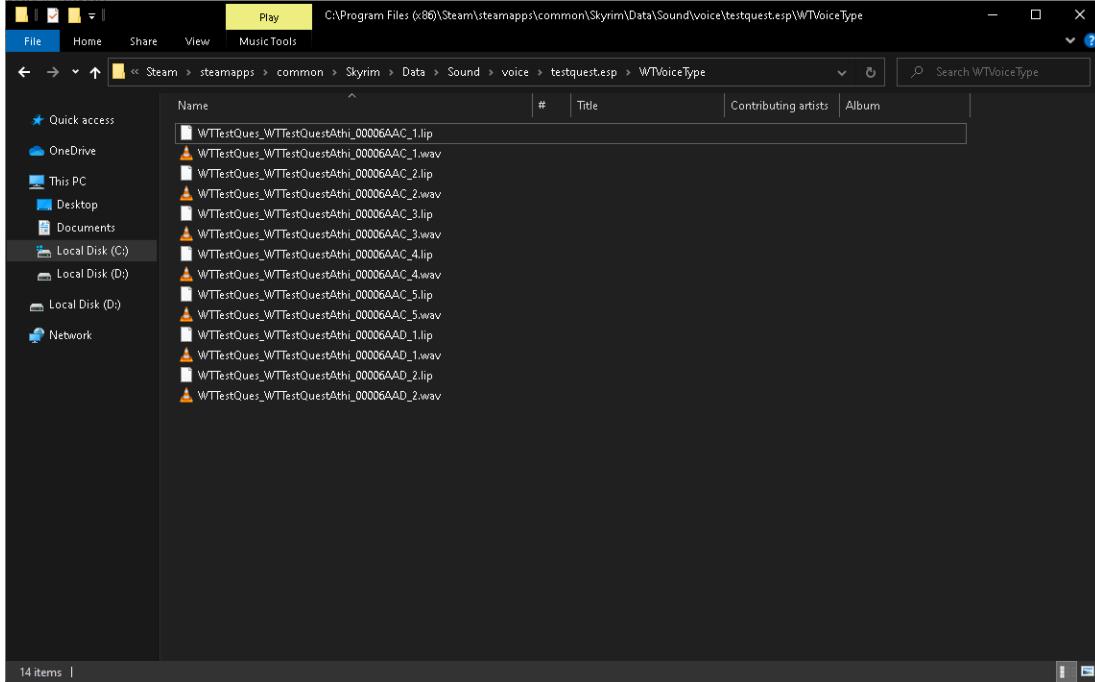


Figure 785 - A .lip file generated for each .wav file.

Lastly, we need to pack the pairs of .lip and .wav files together in a .fuz file. We'll need the [Unfuzer](#) utility which can be downloaded from Nexusmods.

After extracting Unfuzer to a new folder, launch it by running Unfuzer.exe.

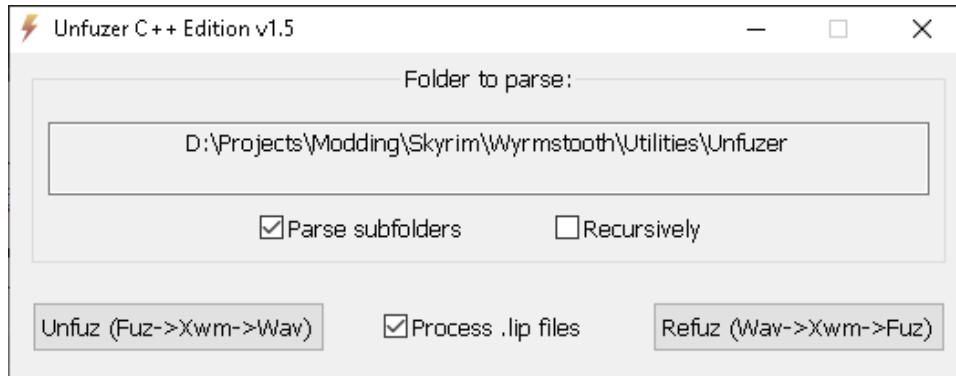


Figure 786 - Unfuzer.

The first thing we need to do is change the path to point to our voice type directory.

Click on the path beneath ‘Folder to parse’.

Select the folder containing the voice type folder. In our example, that’s ‘C:\Program Files (x86)\Steam\steamapps\common\Skyrim\Data\Sound\voice\testquest.esp\WTVoiceType’.

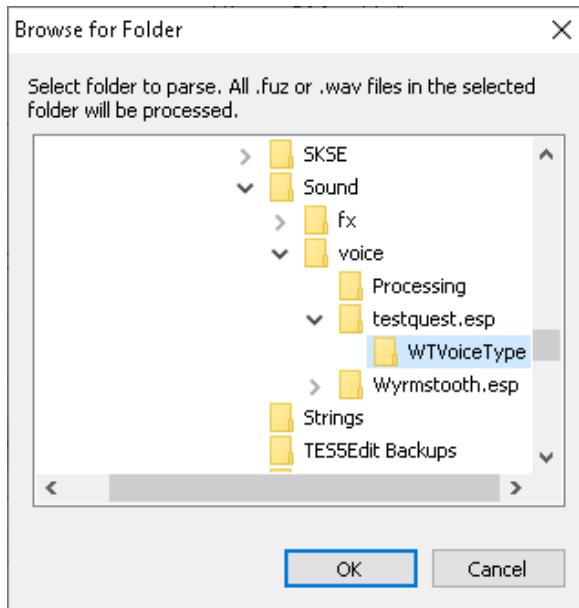


Figure 787 - Selecting the voice type folder.

Click OK.

There are no subfolders so we can untick ‘Parse subfolders’.

Ensure ‘Process .lip files’ is ticked.

Click ‘Refuz (Wav->Xwm->Fuz)’ to begin the process.

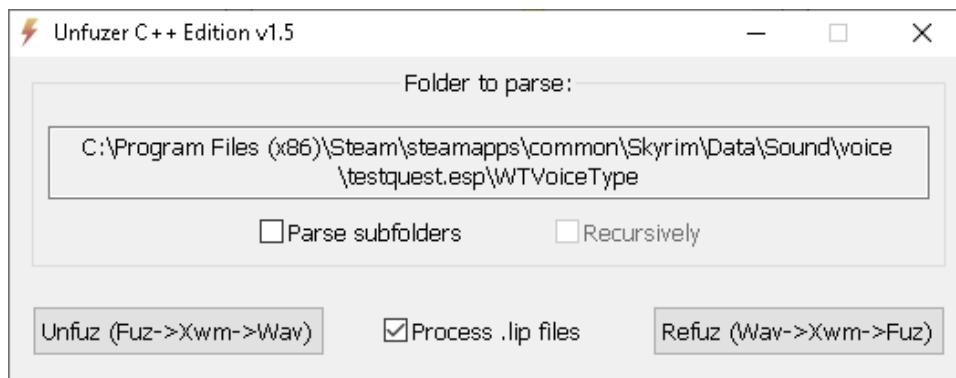


Figure 788 - Ready to fuzz.

Go back to the voice type folder. The .wav and .lip files should now be gone and in their place you should see a .fuz file for each pair of .wav and .lip files.

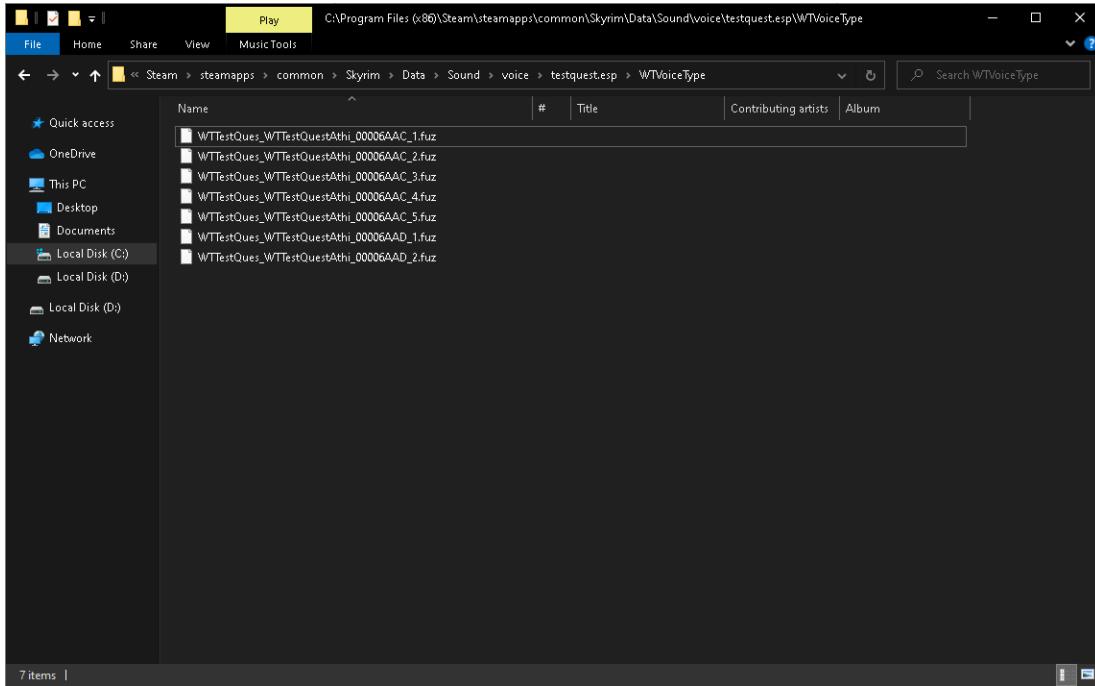


Figure 789 - The .wav and .fuz files fused together.

The last thing left to do is to check out our dialogue in-game to confirm it works.



Figure 790 - The NPC saying their response to the topic selected by the player.

ADDING HELLO, COMBAT, CRIME AND OTHER INCIDENTAL DIALOGUE

This section will cover adding incidental dialogue such as hellos, combat barks, trespass, theft, and pickpocket alerts, and so on. Most of these topics are set up in basically the same manner, so I'll cover a few examples which should give you enough of an idea.

Important: Incidental dialogue should go into a quest with 'Start Game Enabled' ticked.

To start things off, let's create a new quest. See the section [The basics of setting up quests](#) for instructions on how to do this. For this example, I named the quest DialogueGeneric with a 'WT' prefix.

Note: To prevent reanimated NPCs from using these lines, you can add a 'IsCommandedActor == 0' condition to the quest as per the screenshot below:

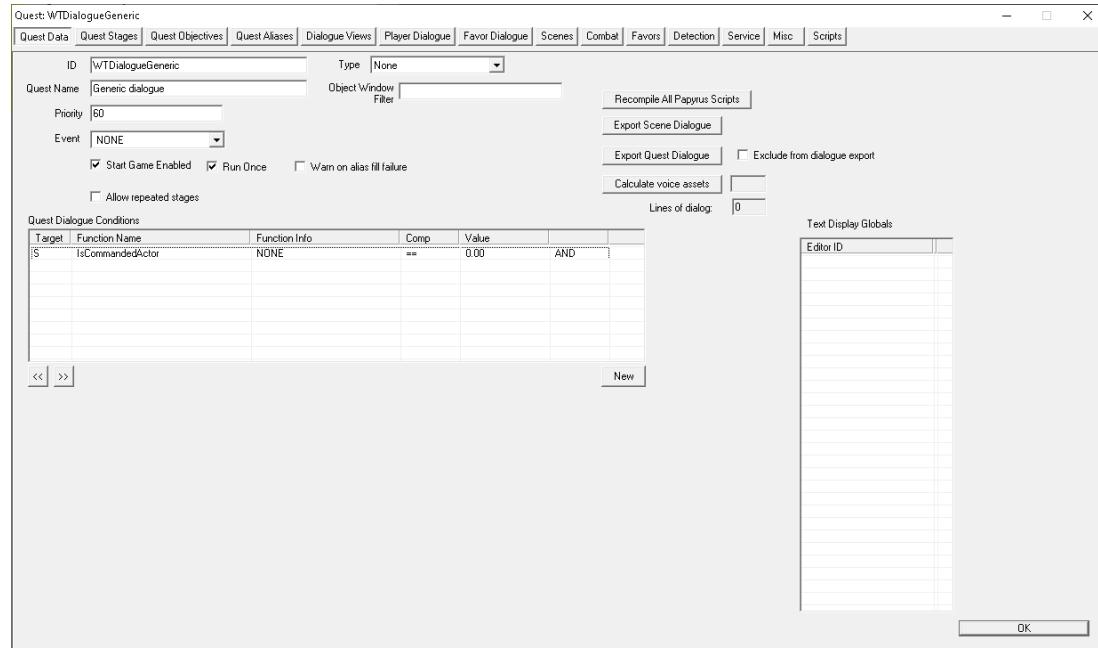


Figure 791 - Quest Data tab.

To add that condition, right-click in the Quest Dialogue Conditions list and select New.

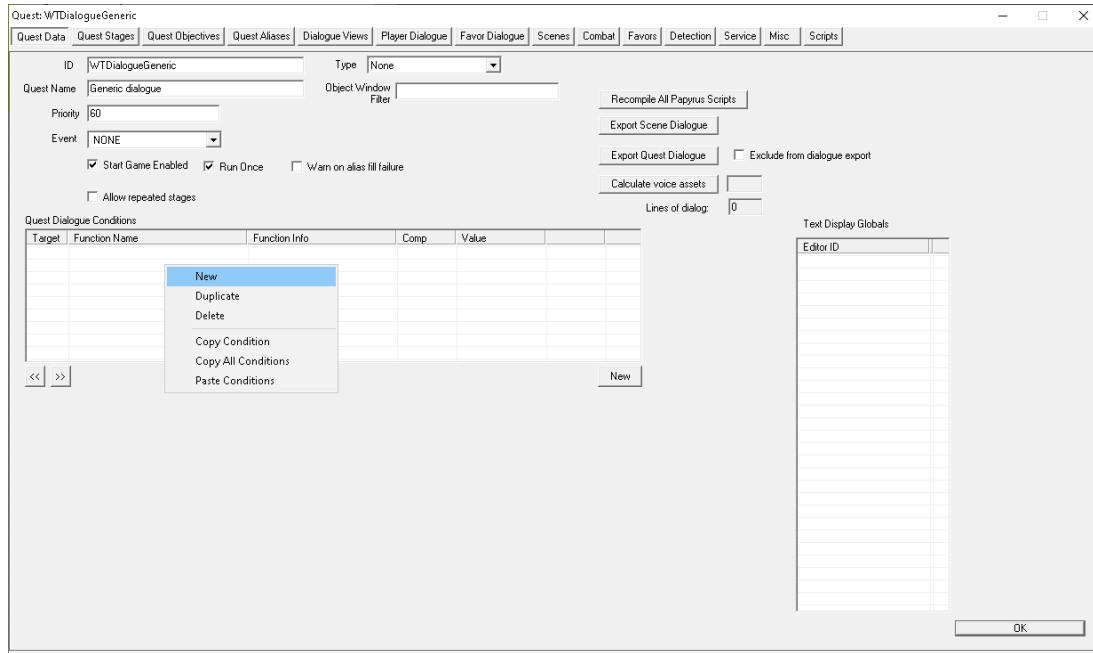


Figure 792 - Adding a Quest Dialogue Condition.

Set the Condition Function drop-down to 'IsCommandedActor' and set the Value field to 0.

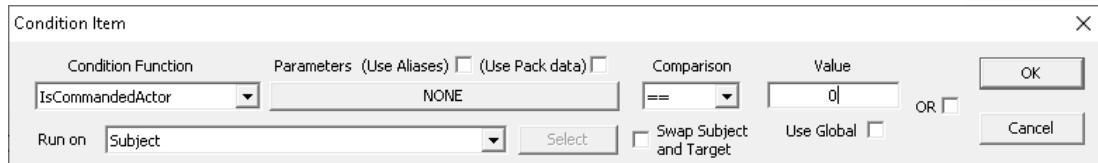


Figure 793 - IsCommandedActor condition function.

Click OK to close out of the Condition Item properties.

Let's start by adding some hellos. These lines are spoken by an NPC automatically when close to the player.

Go to the Misc tab, right-click in the Topics list and select New Topic.

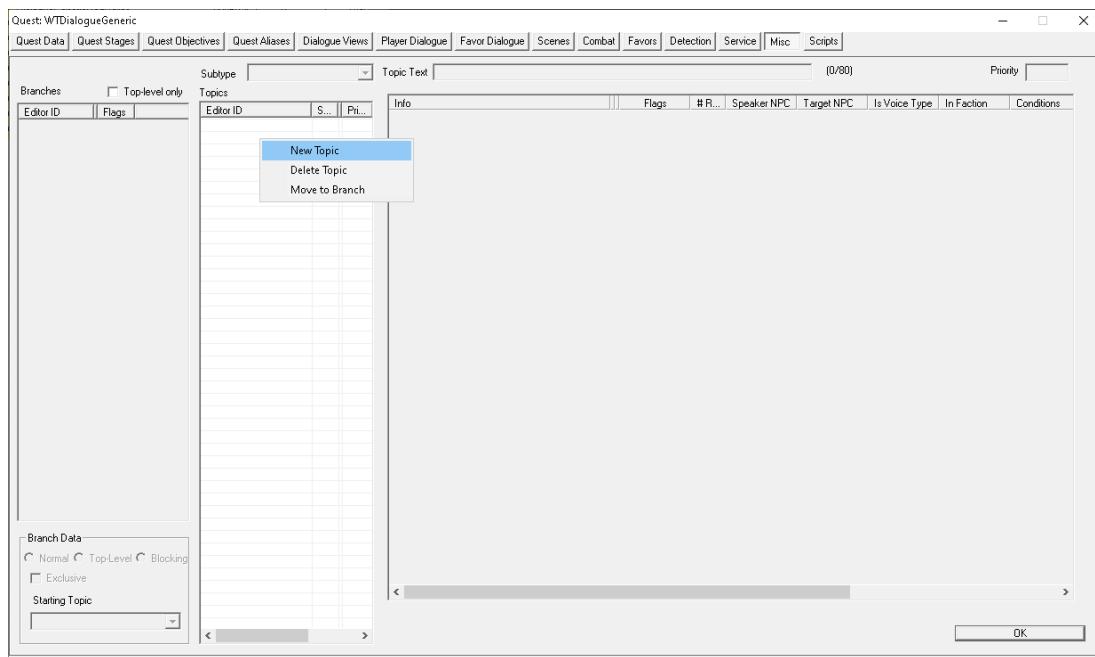


Figure 794 - Adding a new topic.

Select 'Hello' then click OK.

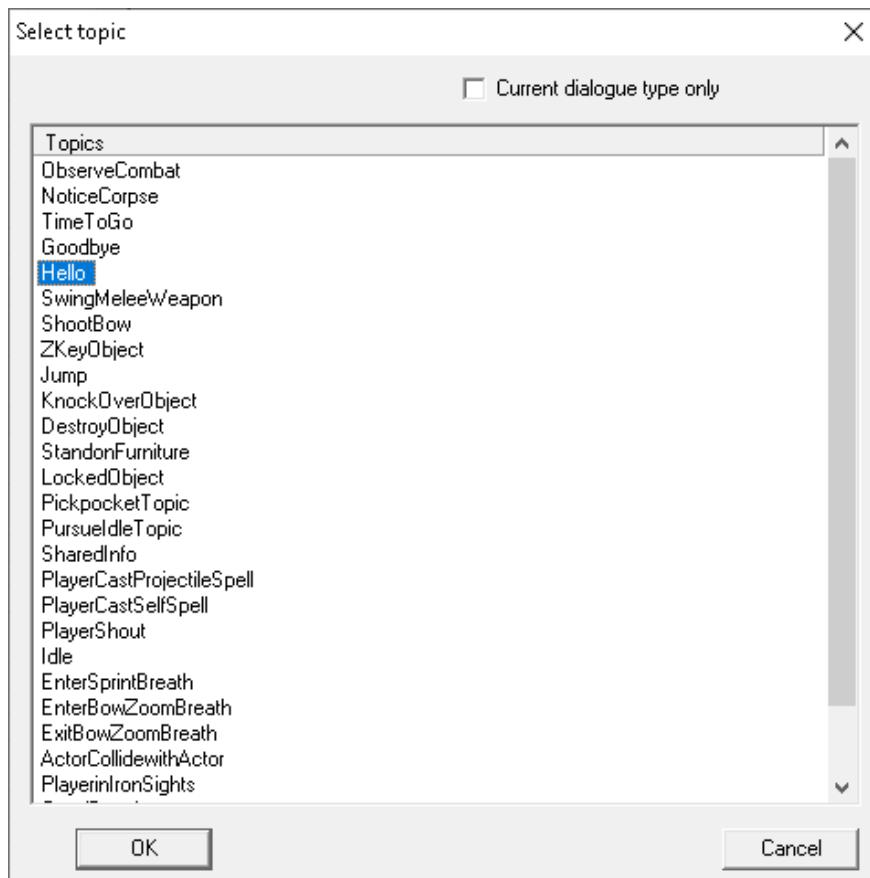


Figure 795 - List of topic types.

Next, you'll be prompted to enter in an ID for the topic.

For this example, I just called it WTDialougeGenericHello.

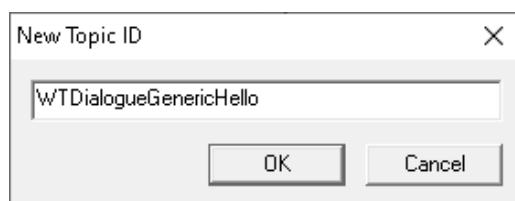


Figure 796 - Topic ID.

Click on the Hello topic in the Topics list to highlight it.

Now right-click in the section to the right of the Topics list and select New to add a new hello.

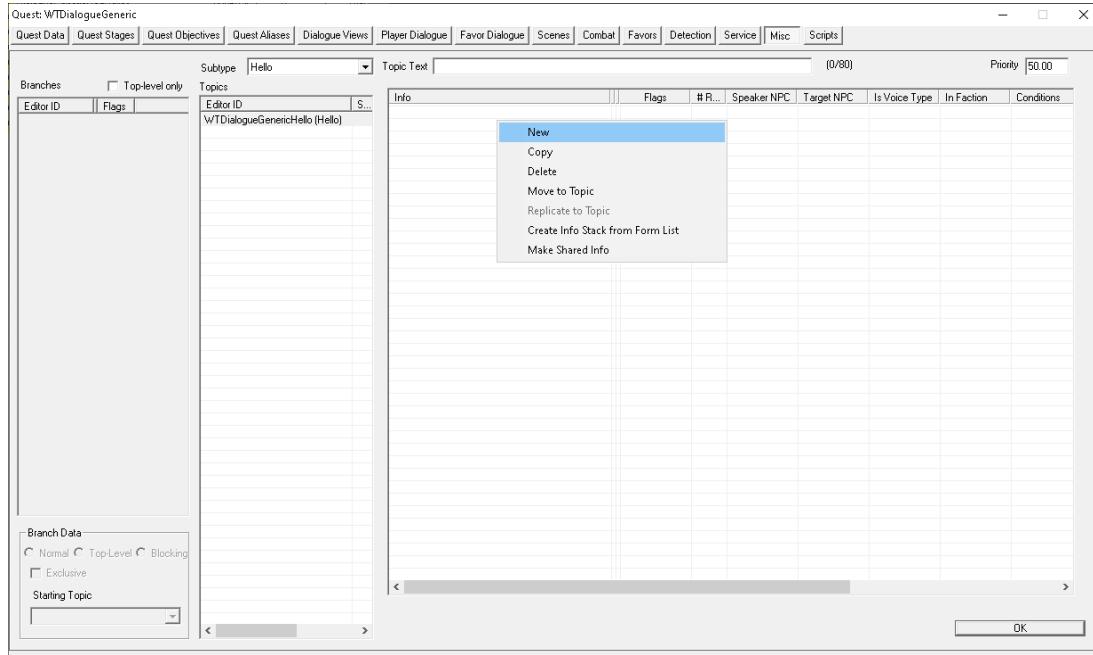


Figure 797 - Adding a new hello.

Enter in the line of dialogue in the Response Text field.

Set the Emotion Type and Emotion Value if necessary then click OK.

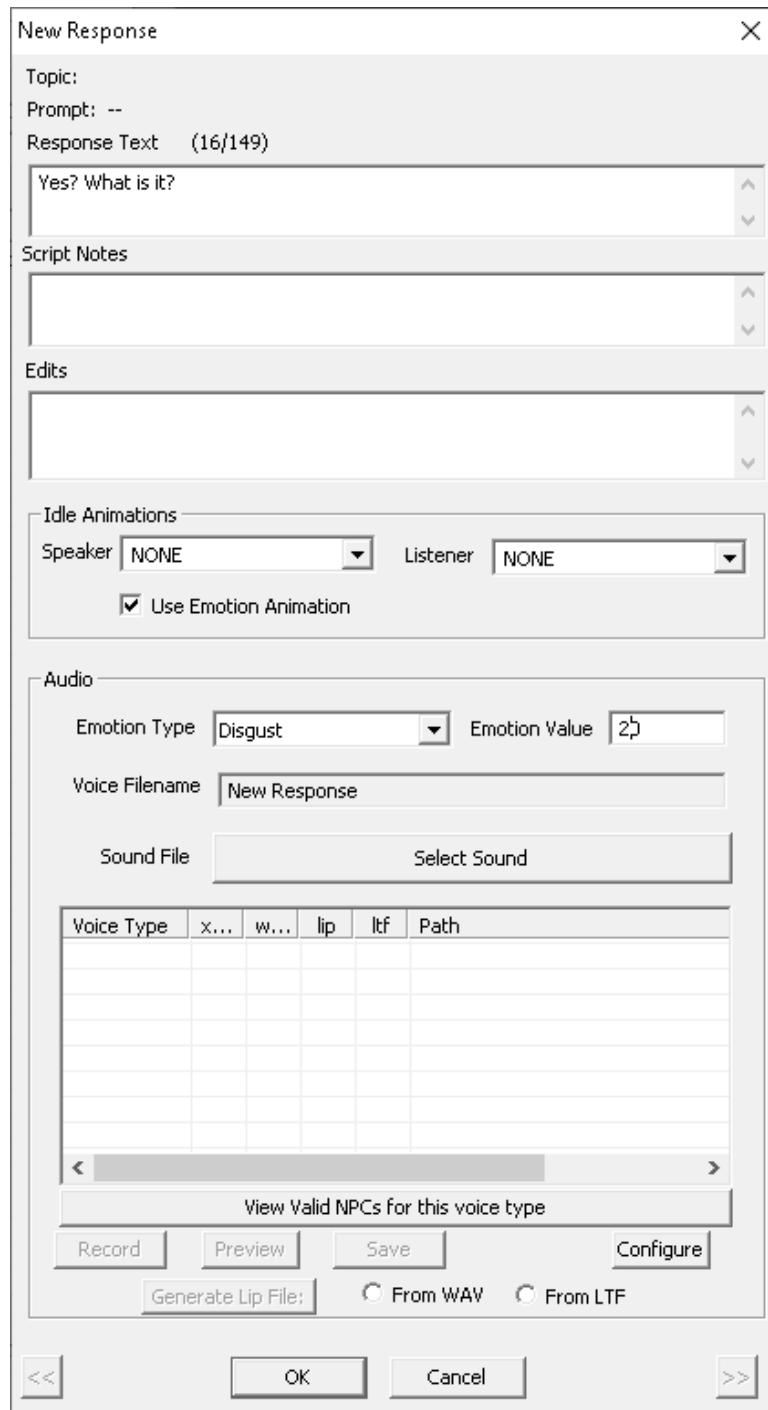


Figure 798 - Adding response text.

Now we need to add a condition to determine which NPCs will say this line.

Right-click in the Conditions list and select New.

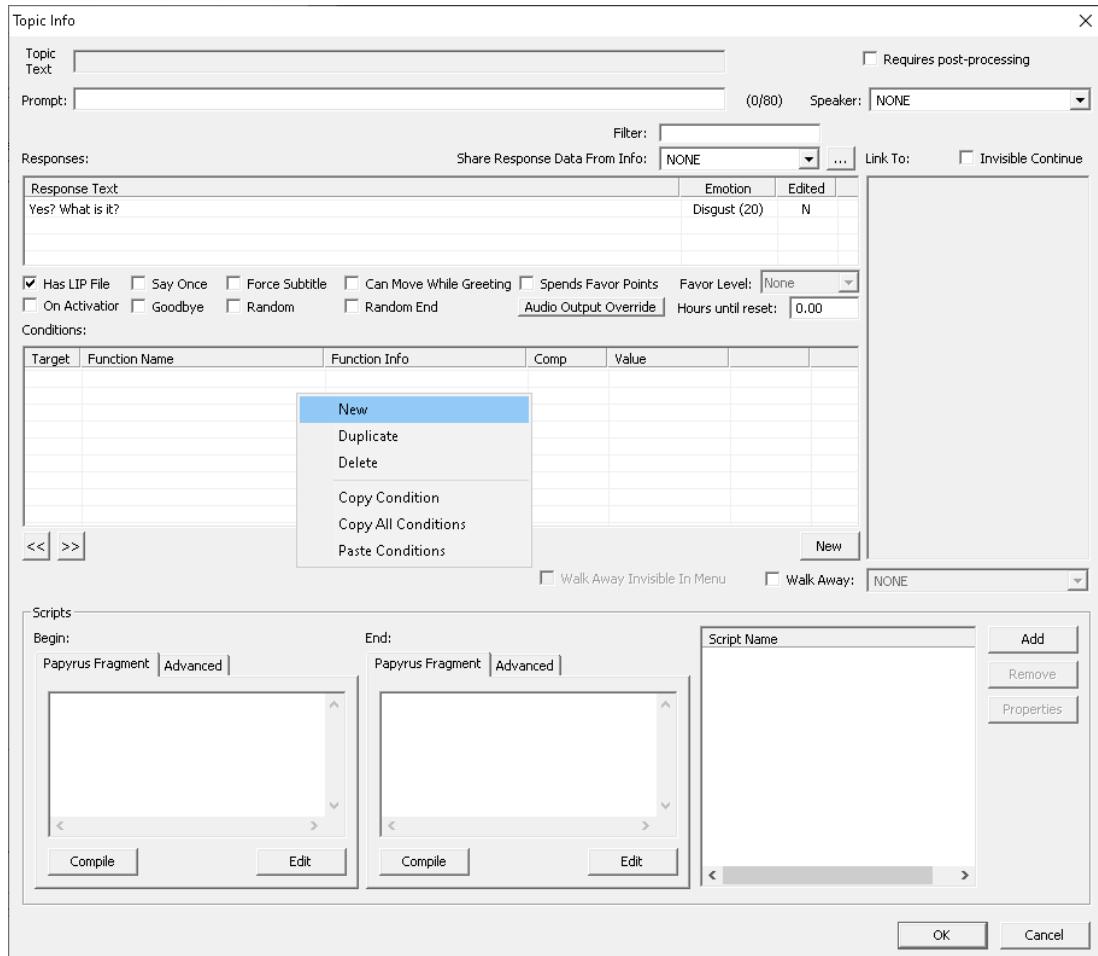


Figure 799 - Adding a condition to determine which NPC(s) will say this line.

One way to do that is by using the GetIsID condition to limit the line to a specific NPC. This is fine if you don't want to share lines between NPCs, if for example, your NPCs are all voiced by different voice actors.

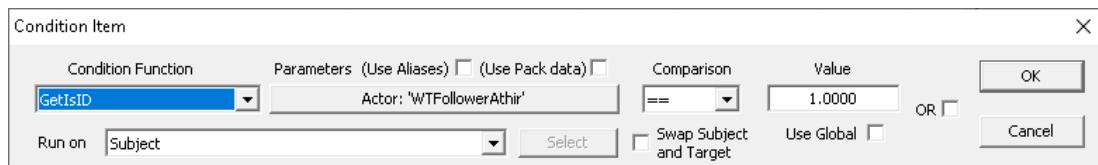


Figure 800 - Adding a GetIsID condition.

Another option is to limit the scope by voice type. This is done in the base game where a single voice actor is responsible for voice acting multiple NPCs.

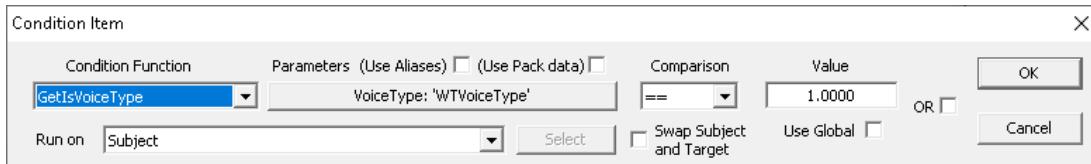


Figure 801 - Adding a *GetIsVoiceType* condition.

You can use a combination of conditions as well. The GetInFaction condition can be used to limit a line of dialogue only to members of a specific faction. For example, if you only want guards to say a specific line of dialogue, you can set the GetInFaction to the 'IsGuardFaction' faction which all guard NPCs are members of.

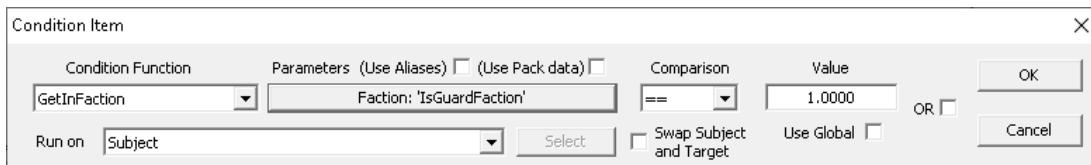


Figure 802 - Adding a *GetInFaction* condition.

For a full list of [Condition Functions](#), see the article on the Creation Kit wiki.

Once your condition has been added, tick the 'Random' tickbox. By default, NPCs will cycle through all their hellos sequentially. By ticking random, NPCs will choose a hello at random instead.

Click OK to close out of Topic Info.

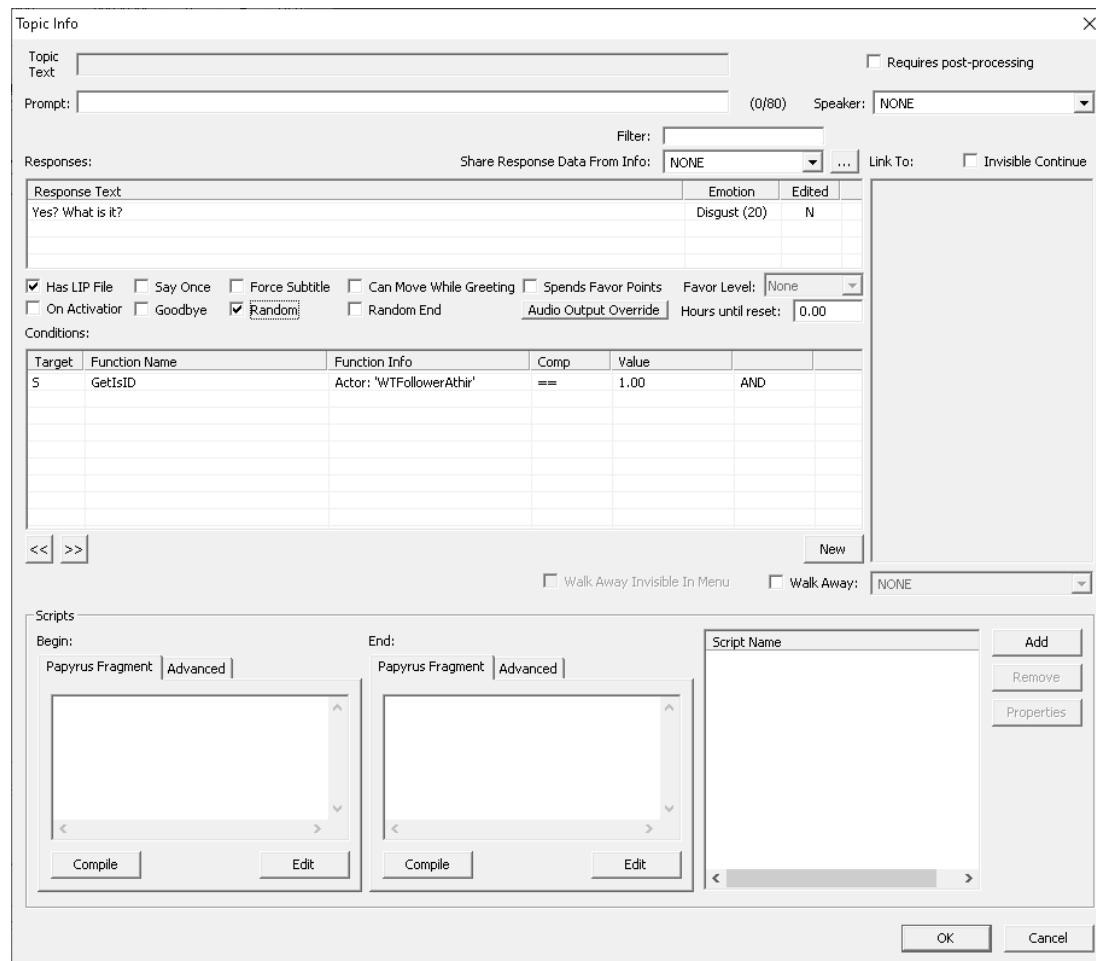


Figure 803 - Response condition added.

You should generally aim to have at least a dozen alternative lines for each NPC or voice type for the sake of variety.

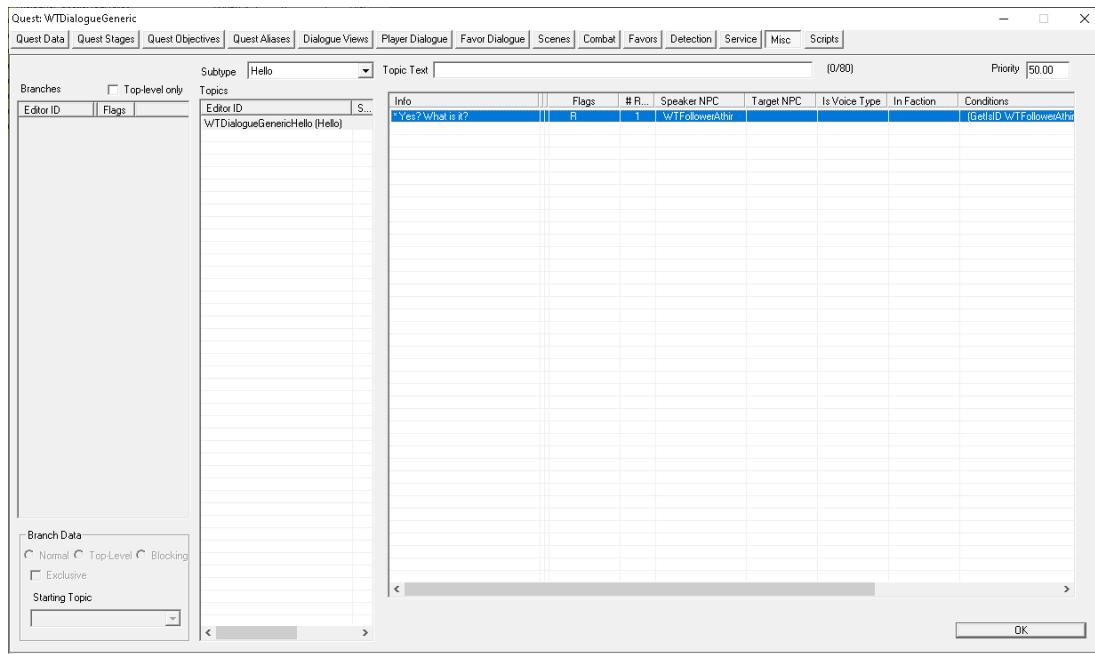


Figure 804 - The first hello added.

The following screenshot is from the WTDIALOGUEidle quest in Wyrmstooth showing you some of the hellos set up for NPCs.

As you can see some of the hellos are configured to only occur at certain quest stages using the GetStage condition function. Some hellos are only configured to occur at certain times of the day using the GetCurrentTime condition function.

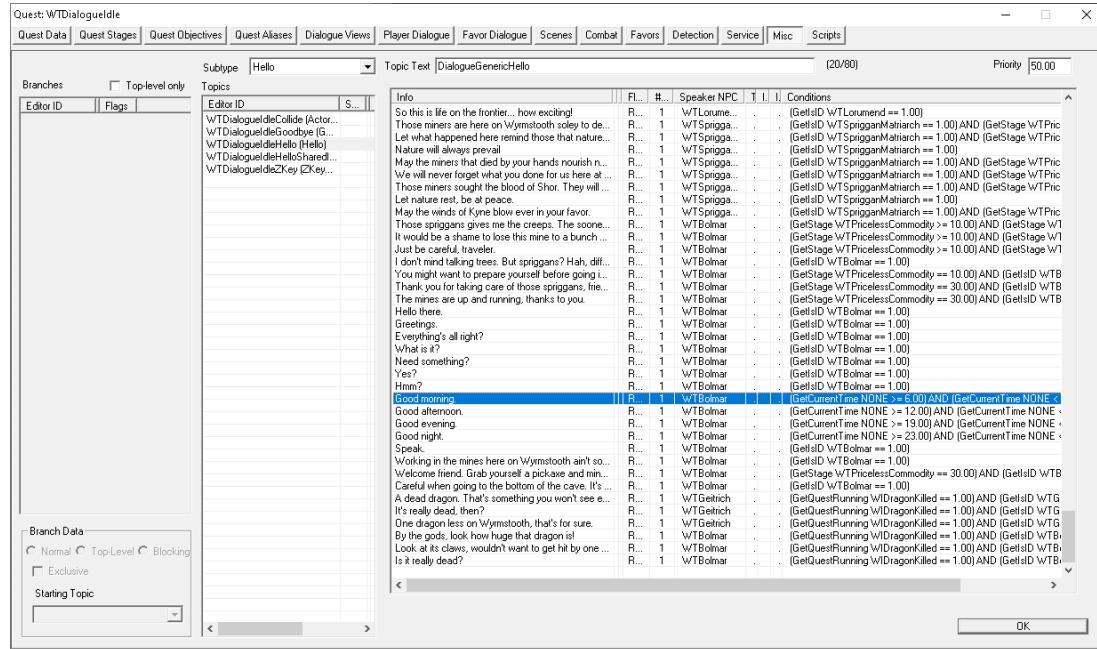


Figure 805 - List of hellos from Wyrmstooth.

Goodbyes are set up in the same way as hellos.

Right-click in the Topics list and select New Topic.

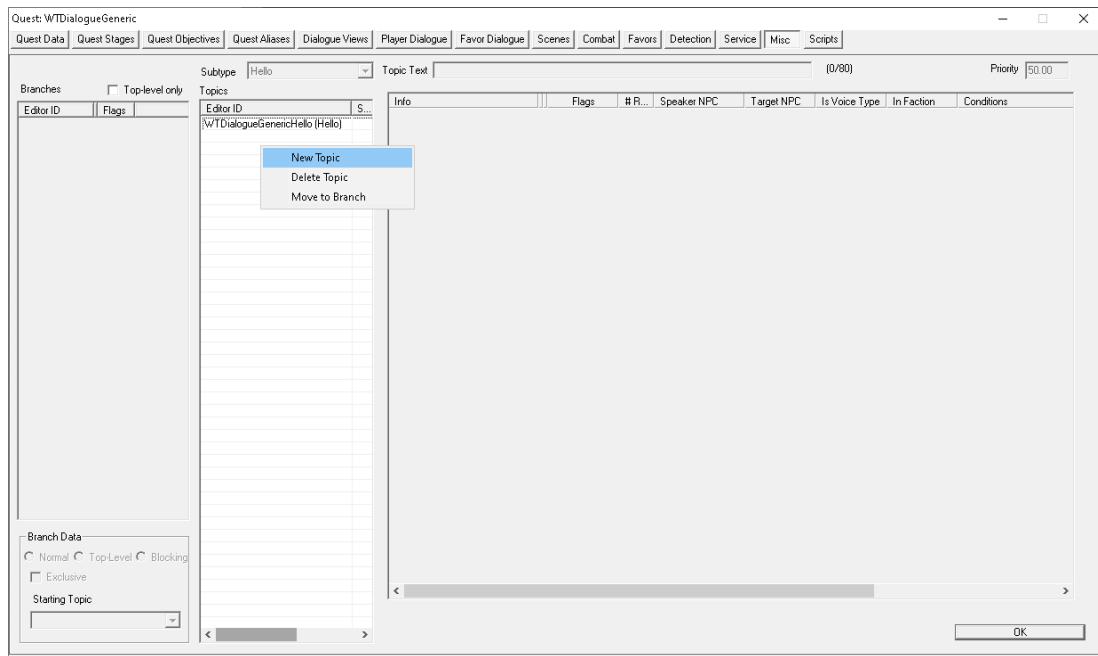


Figure 806 - Adding a goodbye topic.

This time, set the topic type to Goodbye then click OK.

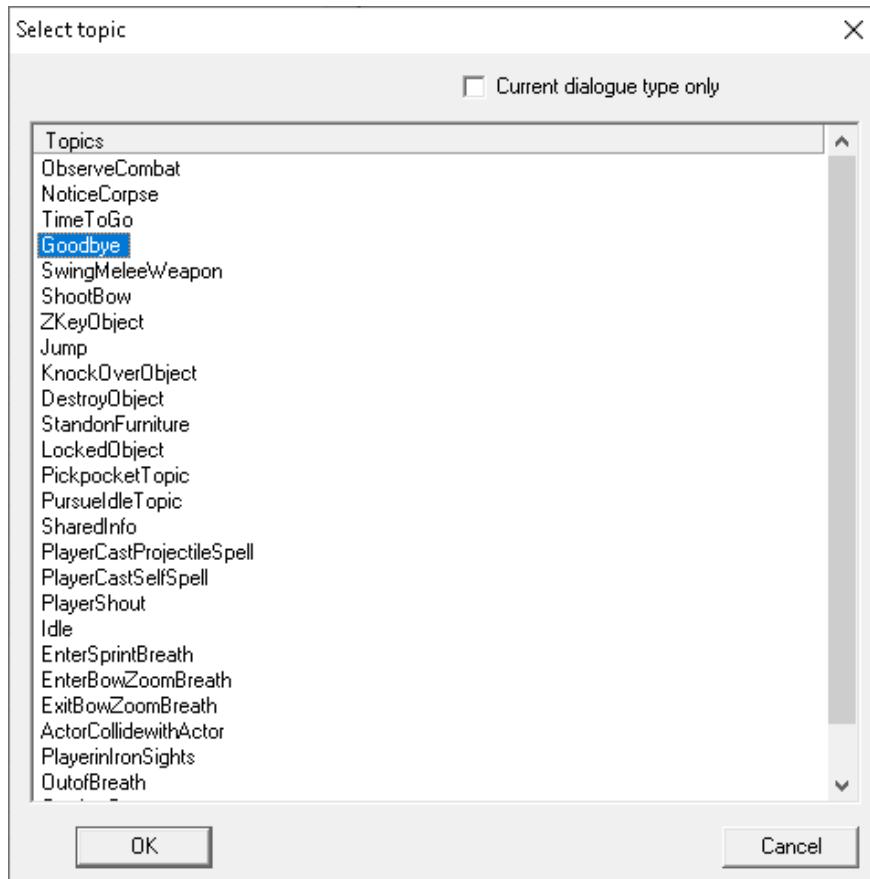


Figure 807 - Selecting the Goodbye topic type.

Enter in the Topic ID. For this example I just called it WTDIALOGUEGENERICGOODBYE.

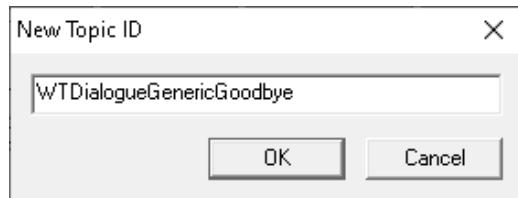


Figure 808 - Setting the topic ID.

To add a goodbye, click on the goodbye topic in the Topics list then right-click in the response list to the right and select New.

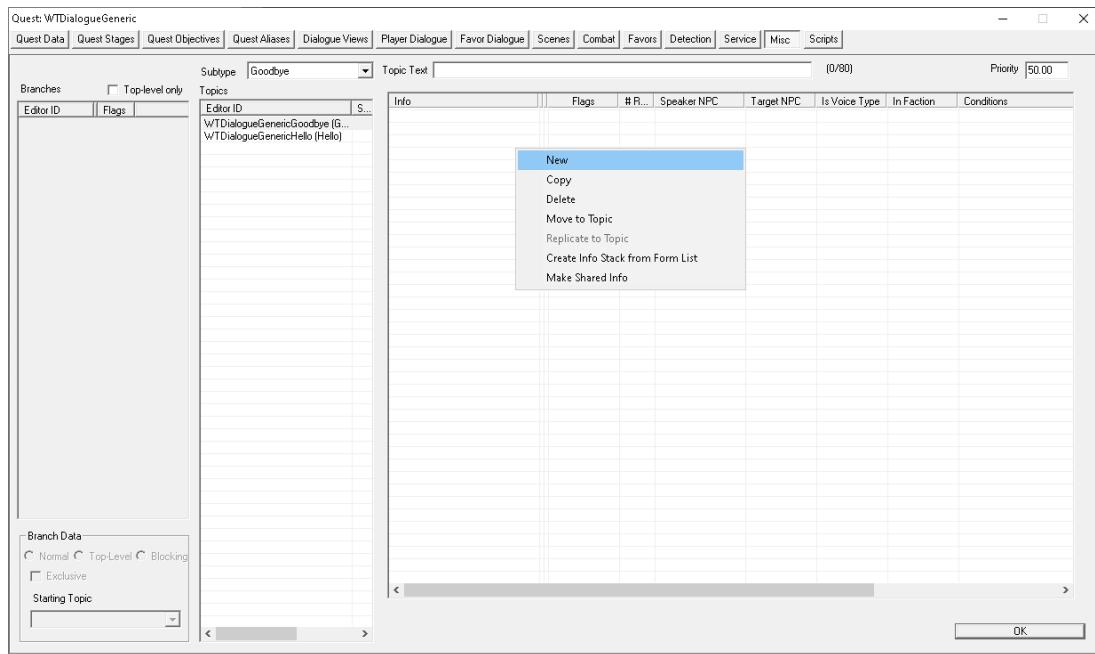


Figure 809 - Adding a new goodbye.

Goodbyes are set up in the same manner as hellos.

Note: You don't need to tick the 'Goodbye' tickbox for Goodbye dialogue.

Goodbyes are triggered when the player manually exits out of dialogue with an NPC.

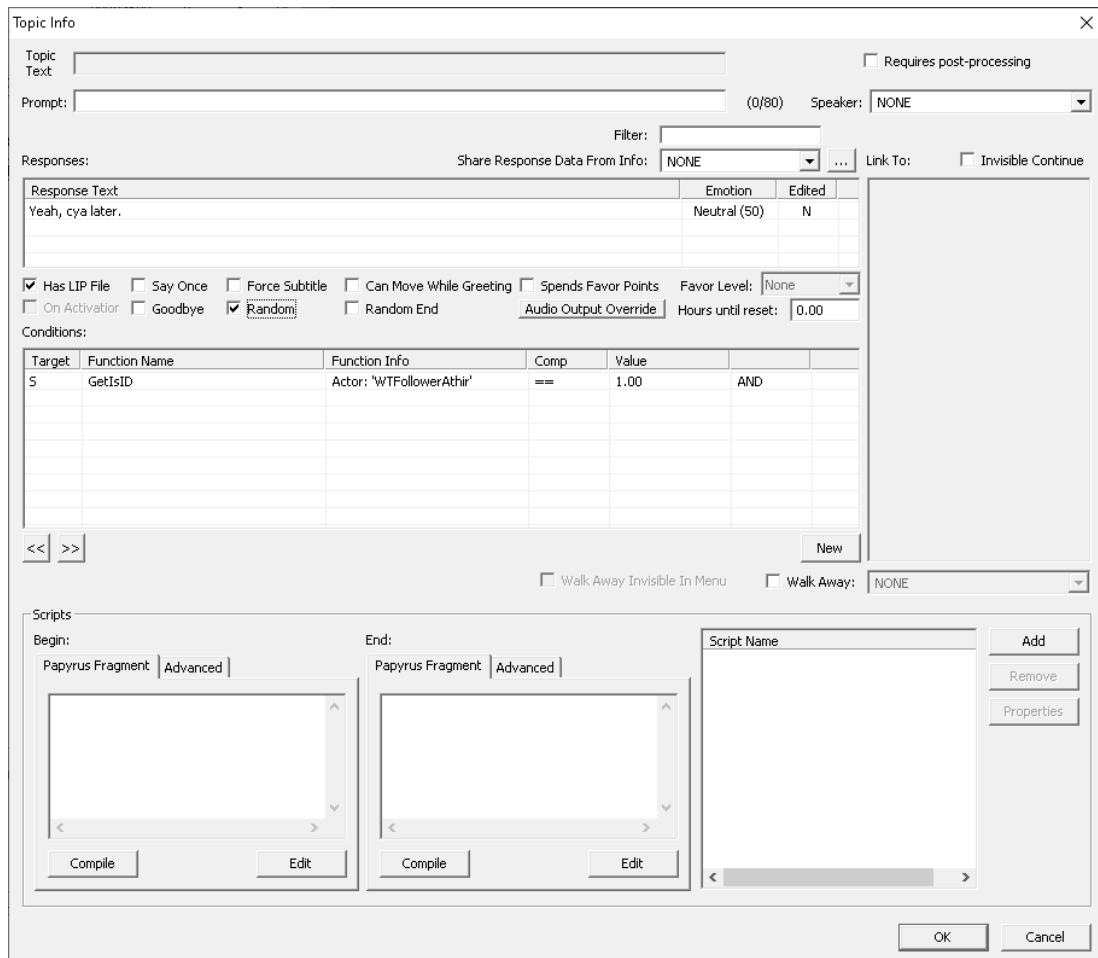


Figure 810 - A sample goodbye.

For more information on each topic type see the [Misc](#) tab article on the Creation Kit wiki.

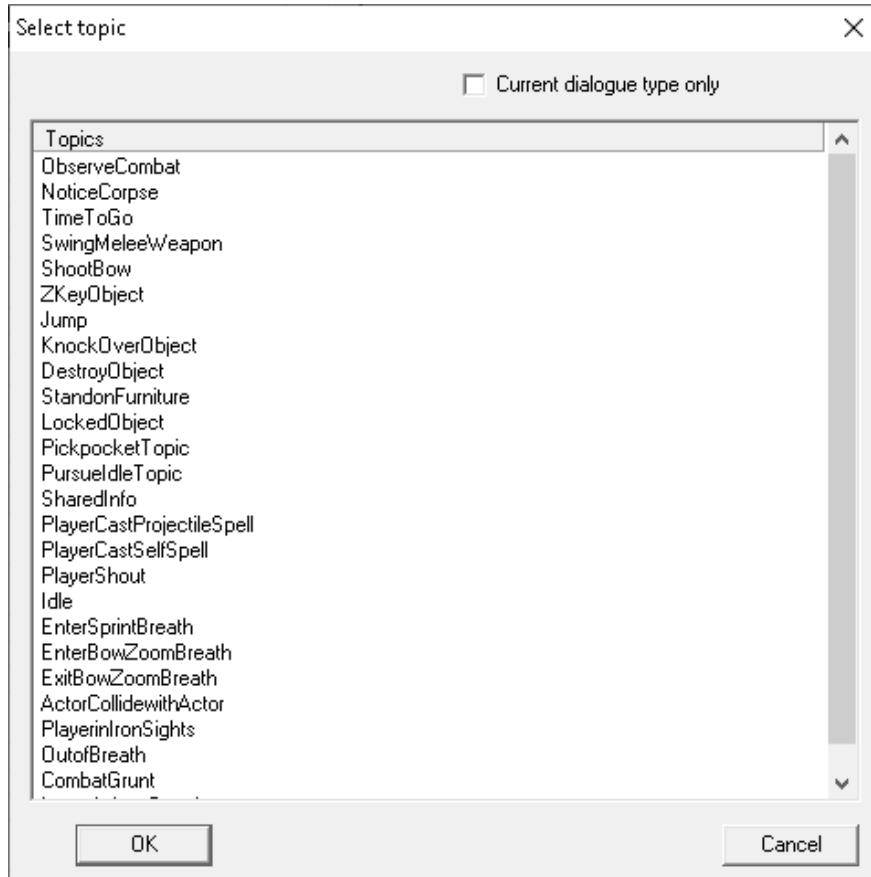


Figure 811 - List of topic types for the Misc tab.

TimeToGo is a trespass warning that doesn't lead to the guards getting called.

EnterSprintBreath, EnterBowZoomBreath, ExitBowZoomBreath, LeaveWaterBreath, OutOfBreath, PlayerCastProjectileSpell, PlayerCastSelfSpell pertain to the player and don't need to be set up for NPCs.

Jump, DestroyObject, StandOnFurniture and PlayerIronSights aren't used.

Next, let's look at setting up topics under the Combat tab.

Topics defined here pertain to dialogue spoken by NPCs during combat, or upon witnessing the player commit a crime.

Let's start by adding an attack line. This line may be used when an NPC makes an attack with a melee weapon.

Right-click in the Topics list and select New Topic.

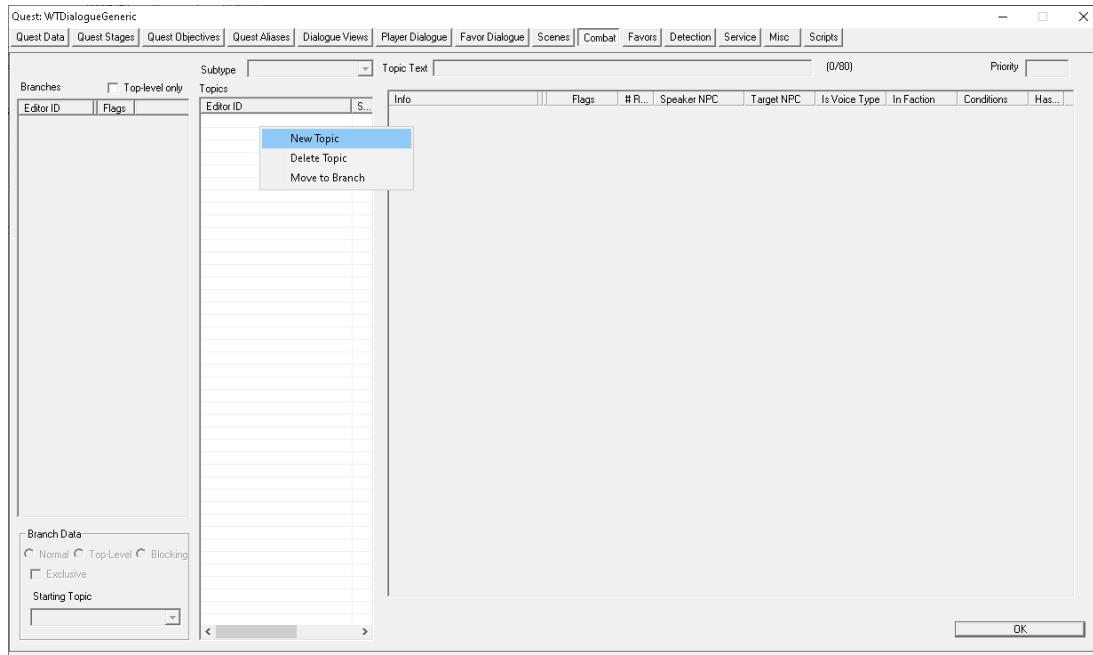


Figure 812 - Adding a new topic to the Combat tab.

Select Attack in the topic types list then click OK.

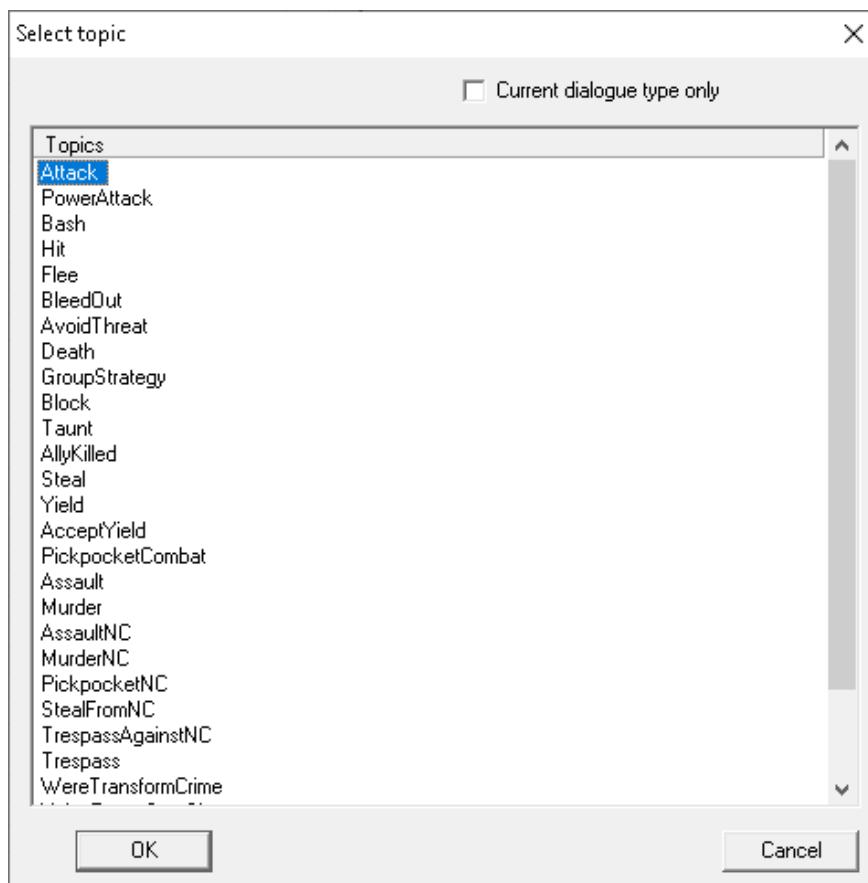


Figure 813 - List of combat topic types.

Set the Topic ID and click OK. For this example, I just set it to WTDIALOGUEGENERICATTACK.

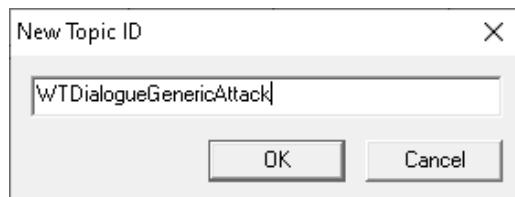


Figure 814 - Setting the topic ID for the attack topic.

Click on the attack topic to highlight it, then right-click in the section to the right and select New.

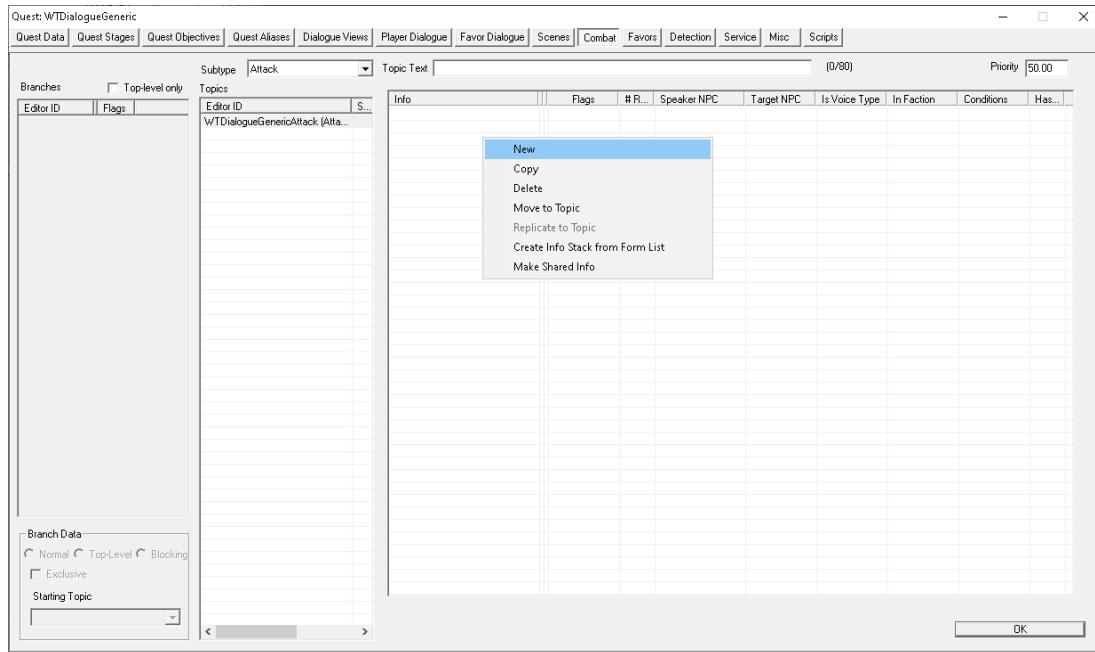


Figure 815 - Adding a new attack line.

Enter in the line of dialogue in the Response Text field. Generally you're going to want to set the Emotion Type to Anger for attack lines.

Click OK once you're done.

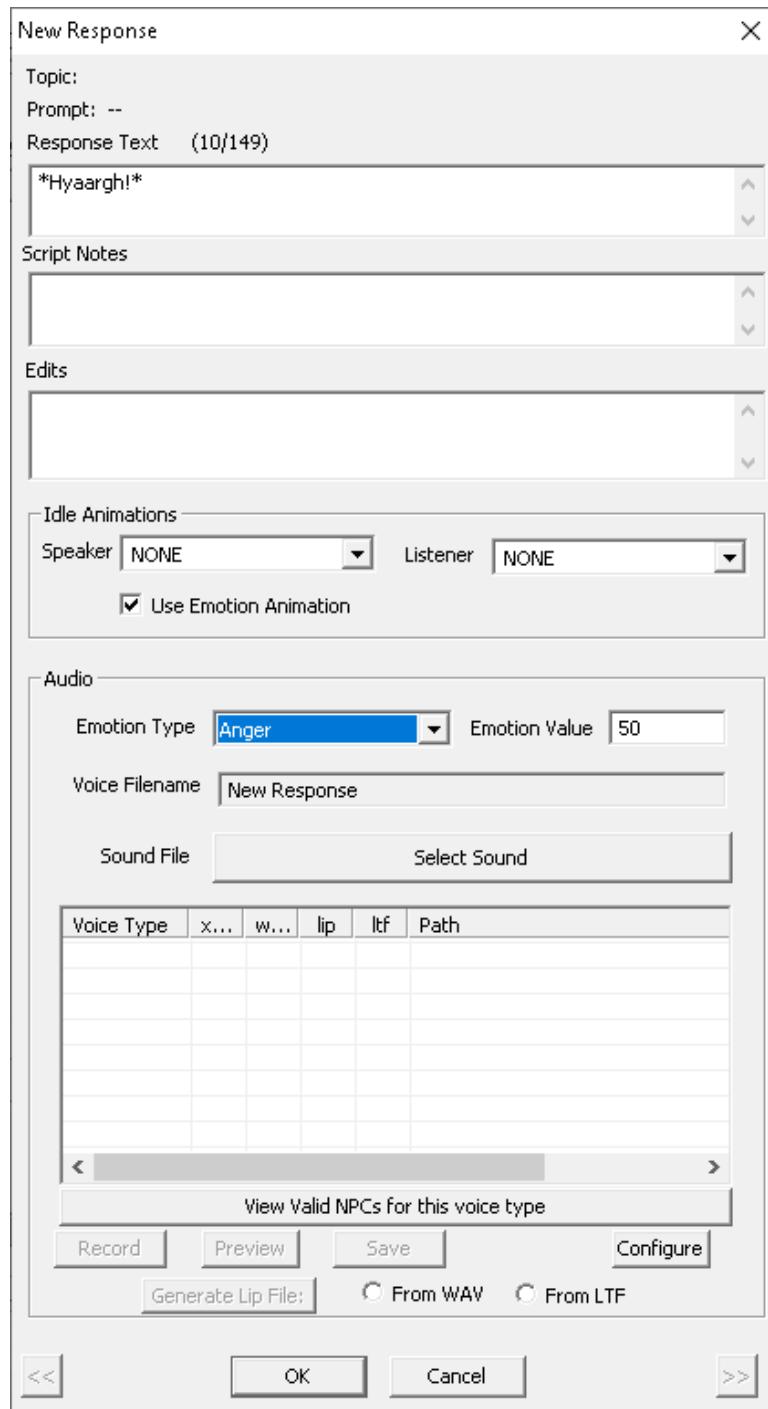


Figure 816 - Setting up an attack line.

Add a condition to determine which NPC(s) should use this line then click OK to close out of Topic Info.

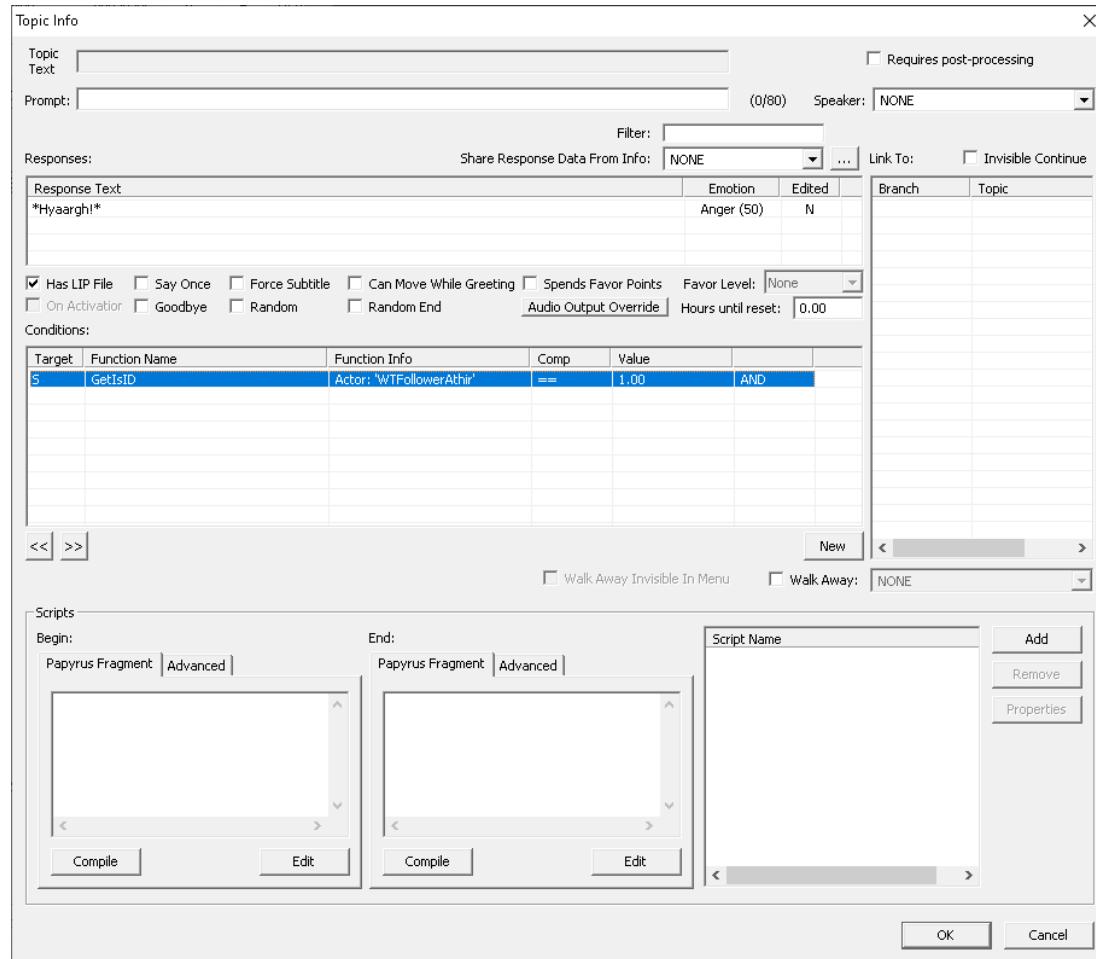


Figure 817 - Added a condition to the attack line.

You're going to want to have at least half a dozen or more different attack grunts for variation.

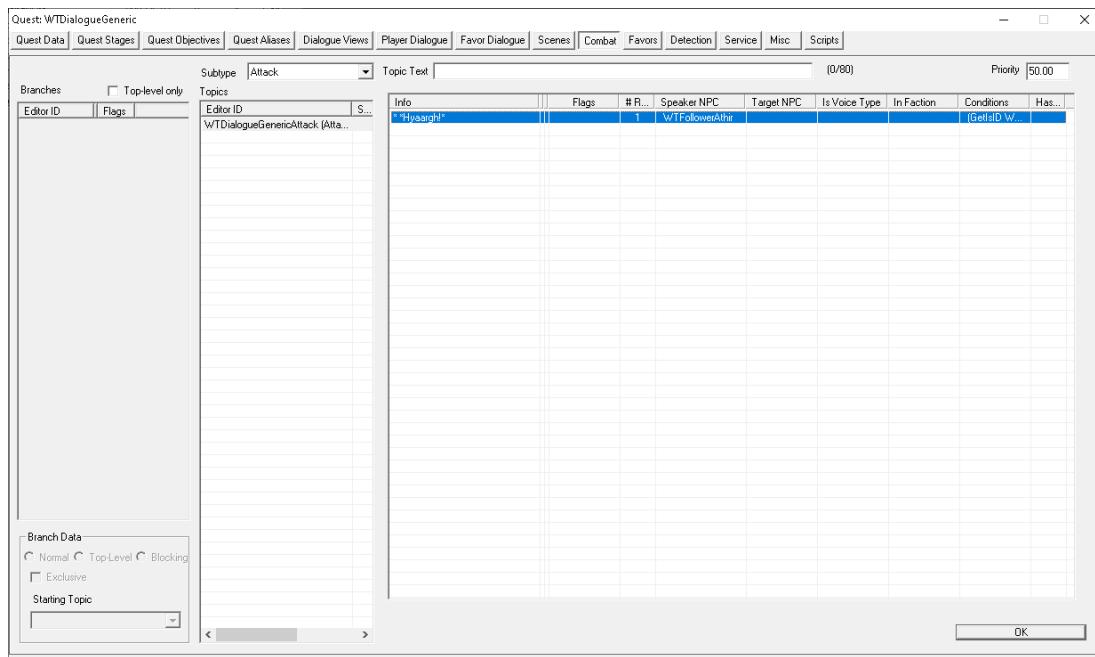


Figure 818 - The attack line has been added.

We can set up combat taunts in the same manner. These are phrases randomly spoken while an NPC is engaged in dialogue.

Right-click in the Topics list and select New Topic.

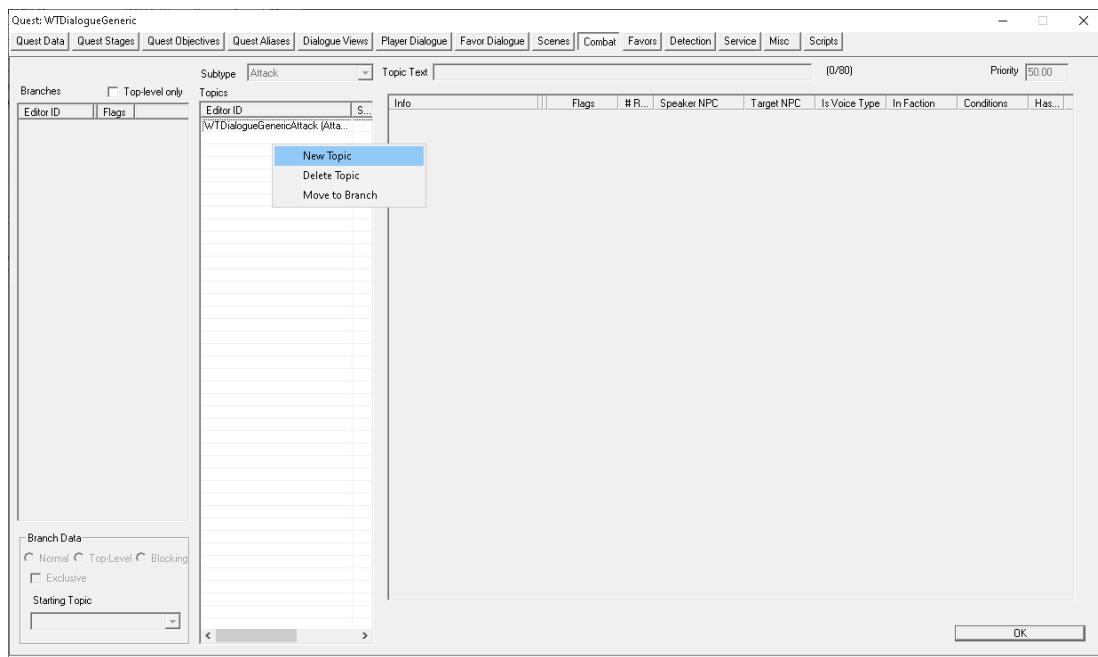


Figure 819 - Adding a taunts topic.

Click on the Taunt topic type to highlight it then click OK.

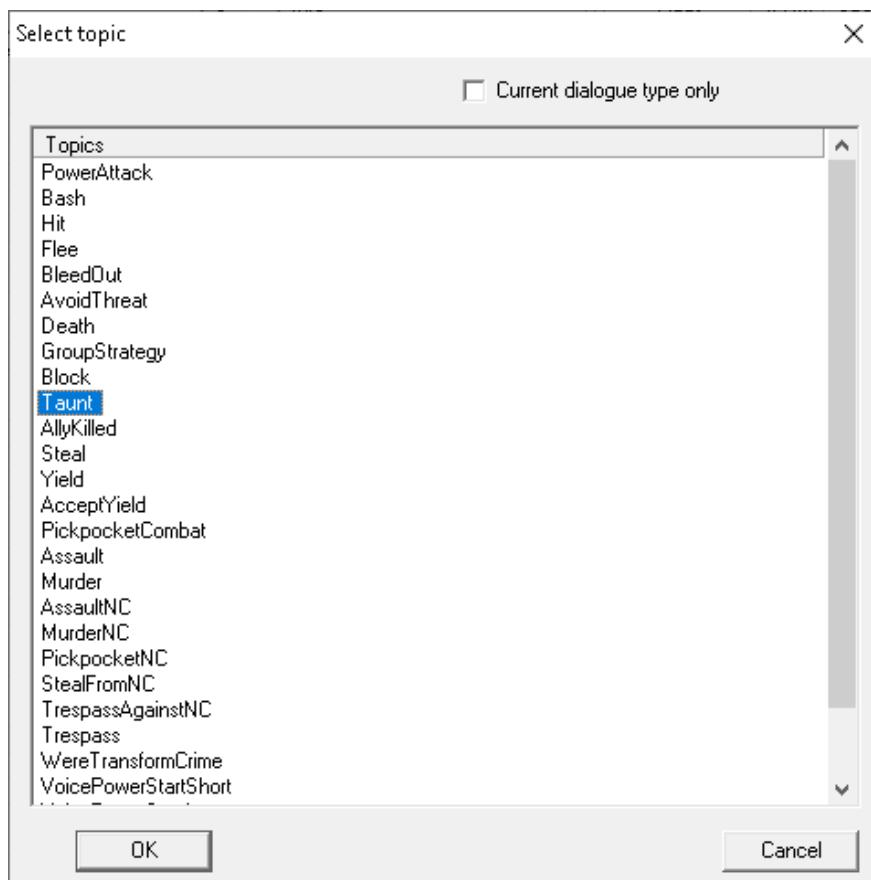


Figure 820 - Selecting the taunt topic type.

Enter in a Topic ID then click OK. For this example, I just went with WTDIALOGUEGENERICTAUNTS.

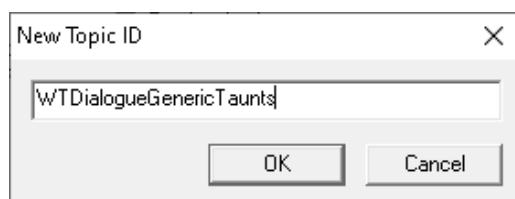


Figure 821 - Entering in a topic ID for the taunt topic.

Click on the taunts topic to highlight it then right-click in the area to the right and select New.

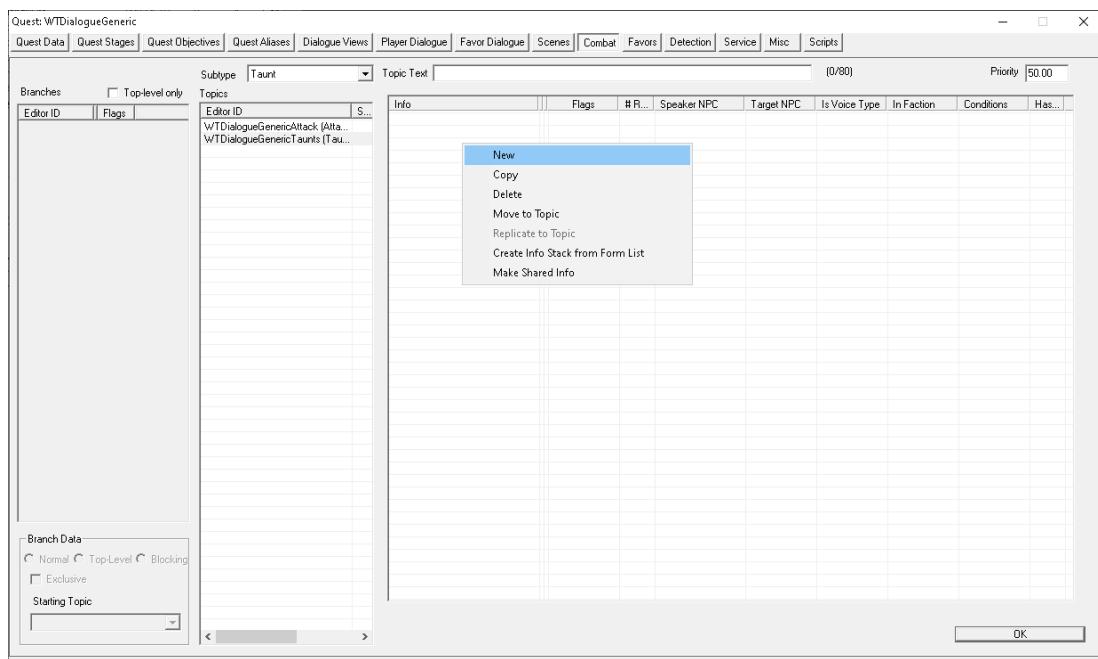


Figure 822 - Adding a new combat taunt.

Enter in the line of dialogue in the Response Text field. Again, you're probably going to want to set the Emotion Type to Anger.

Click OK to close out of New Response.

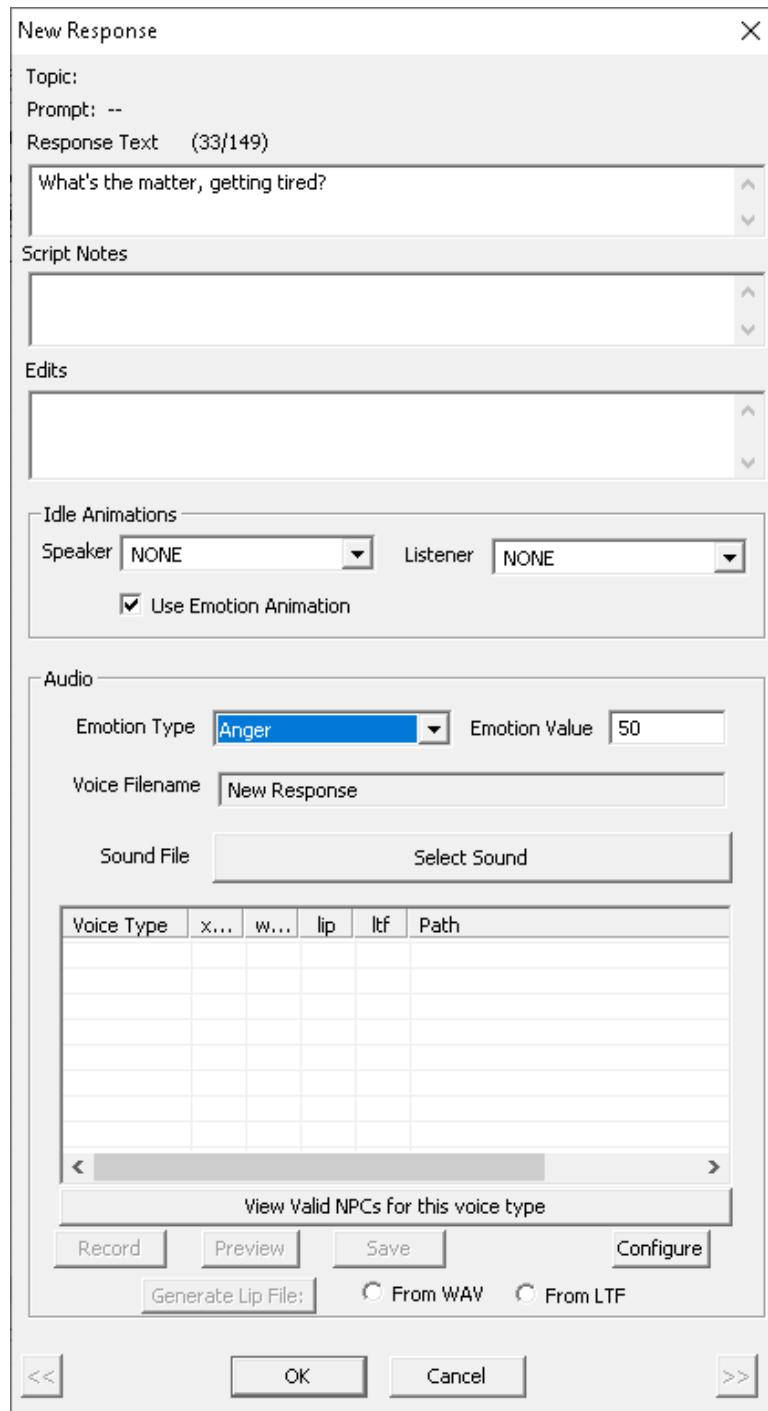


Figure 823 - Setting up a combat taunt.

Again, tick Random and add a condition to determine which NPC(s) can use this line of dialogue during combat then click OK.

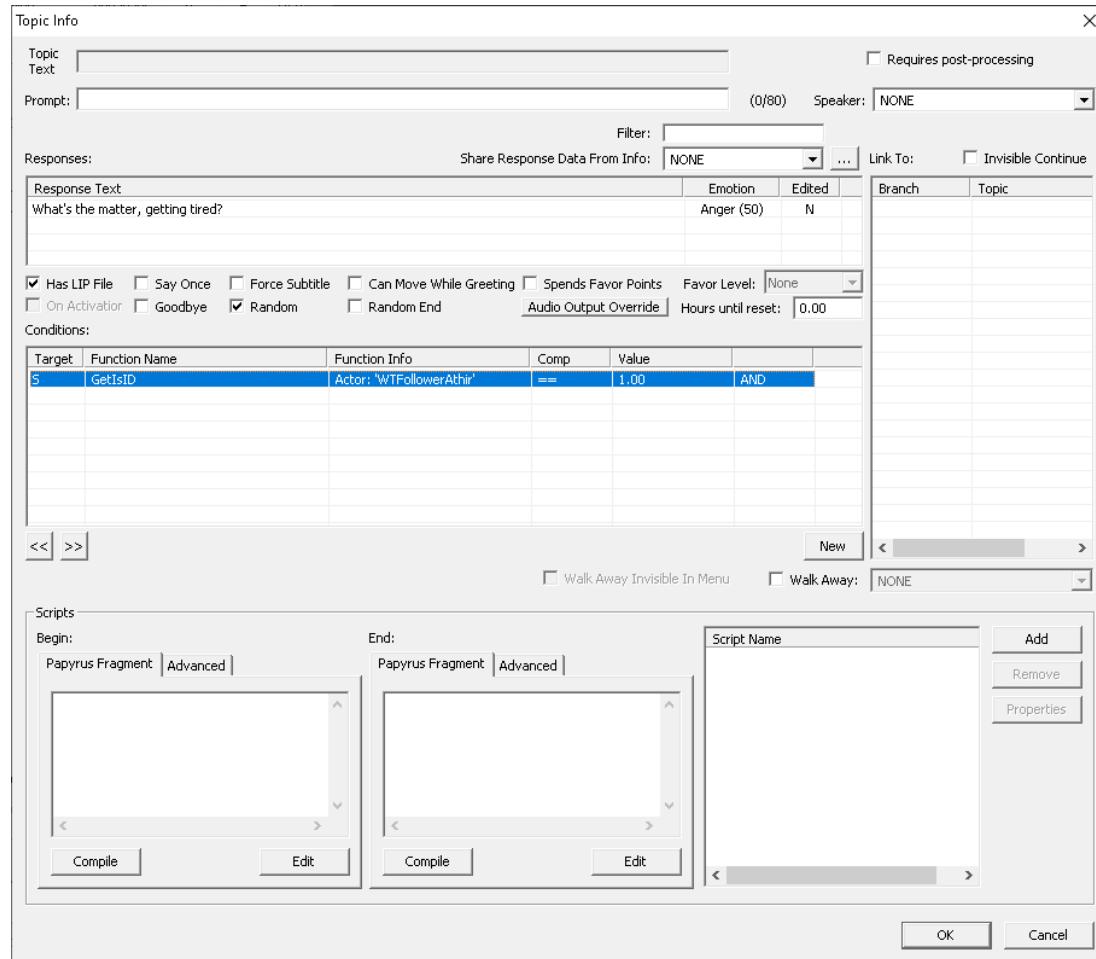


Figure 824 - Combat taunt topic info properties.

Here are some combat taunts I set up for Wyrmstooth:

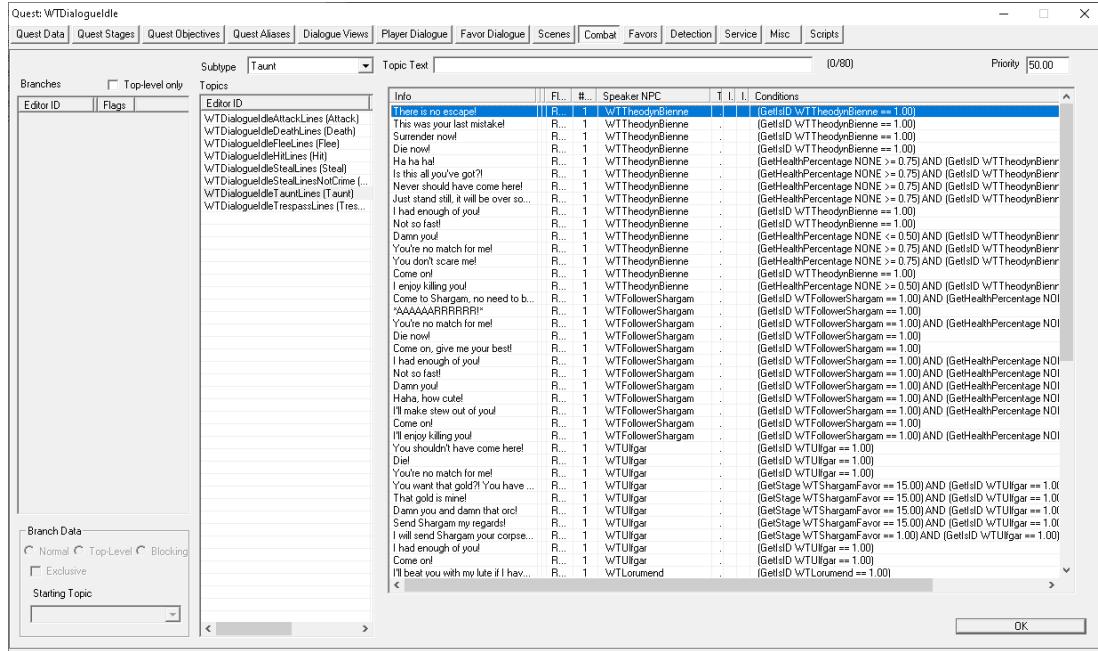


Figure 825 - Combat taunts in Wyrmstooth.

Some of the combat taunts make use of the GetHealthPercentage condition function so that optimistic lines are only spoken if the NPC is in good health, while more pessimistic lines are only spoken at low health when the NPC is closer to death.

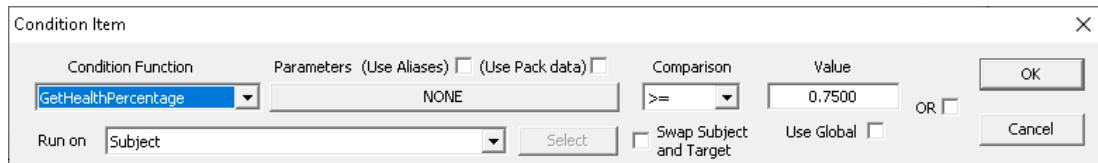


Figure 826 - GetHealthPercentage function checking whether the NPC has 75% of their health or more.

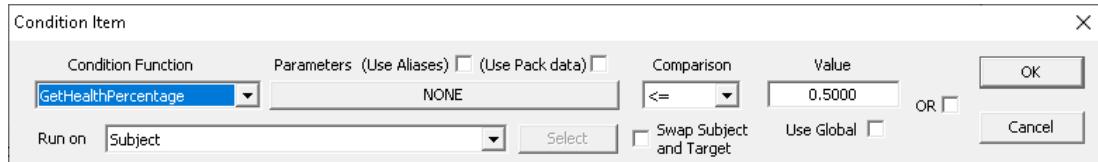


Figure 827 - GetHealthPercentage function checking whether the NPC is down to 50% health or less.

In Wyrmstooth, I also make use of the GetStage condition function to only play certain combat taunts if the NPC has learned specific information from the player.

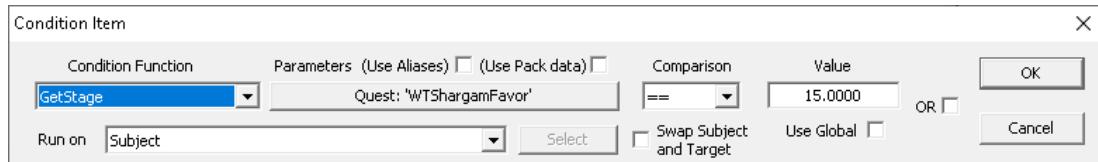


Figure 828 - GetStage condition used to play specific taunts only at a specific quest stage.

Let's look at crime lines next.

Firstly it's important to note that crime topics also have a 'No Care' variant. They're the ones that have 'NC' appended to the end.

No Care lines are used when the player is spotted committing a crime, but the NPC witnessing the player is not in or is not friendly with the faction that the crime was committed against.

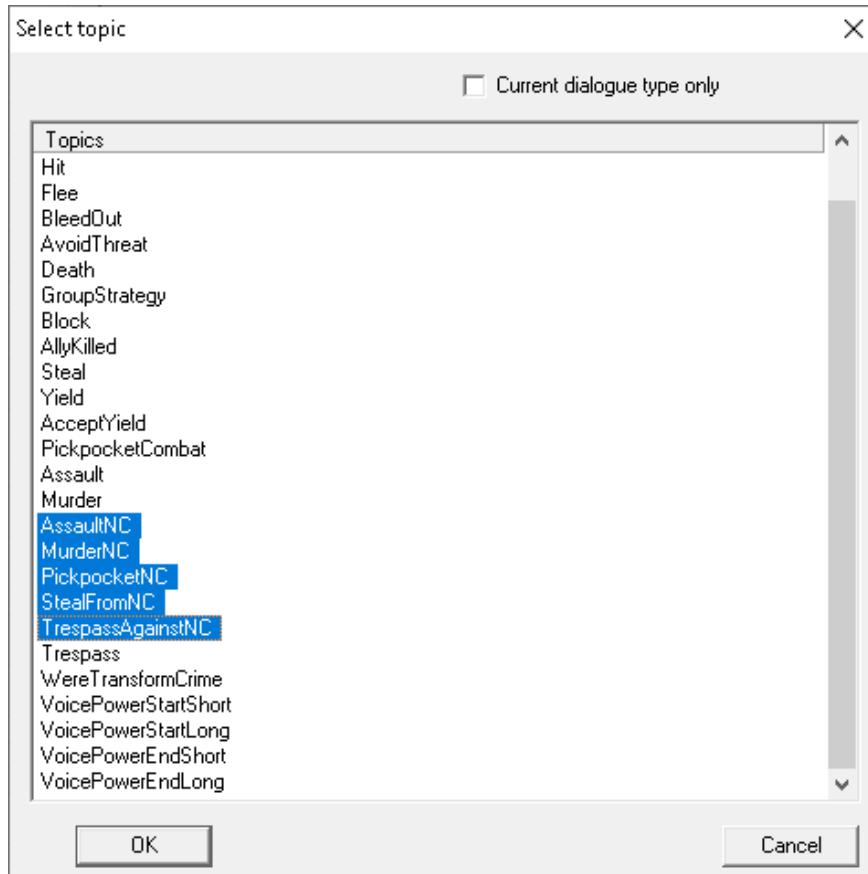


Figure 829 - No Care topics highlighted.

Crime topics are set up in the same manner as the Attack and Taunt topics, but Trespass is a bit different.

TrespassAgainstNC is the No Care version that does not result in guards getting called.

Trespass, on the other hand, escalates in several warning levels, with the last level resulting in guards getting called.

Let's take a look at setting up the Trespass topic.

Right-click in the Topics list and select New.

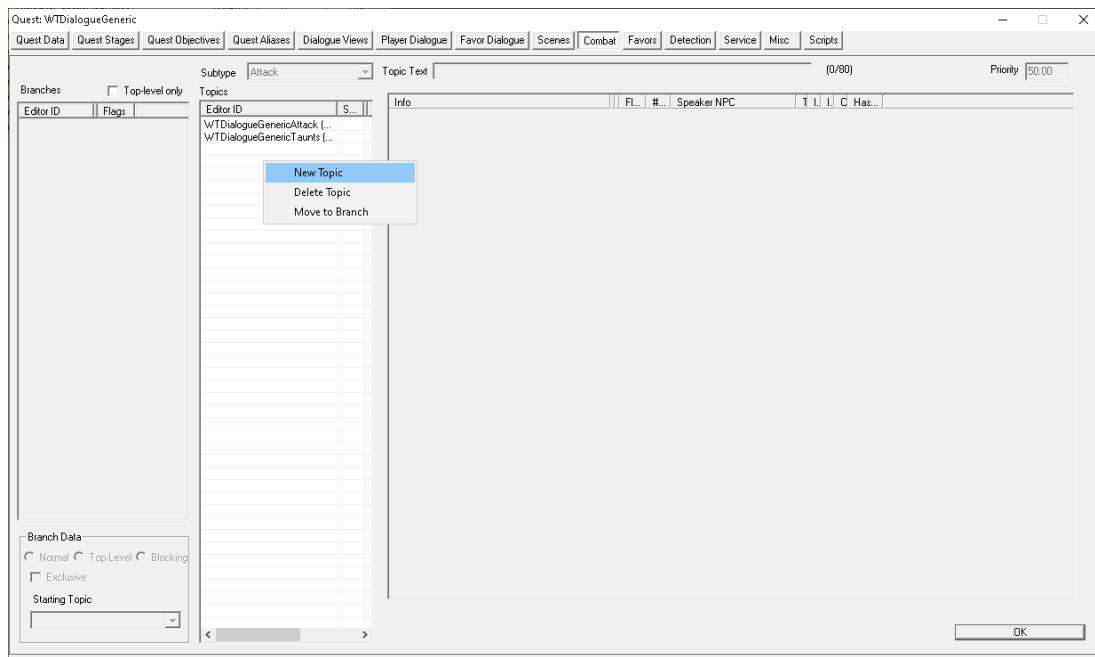


Figure 830 - Setting up the Trespass topic.

Click on Trespass to highlight it then click OK.

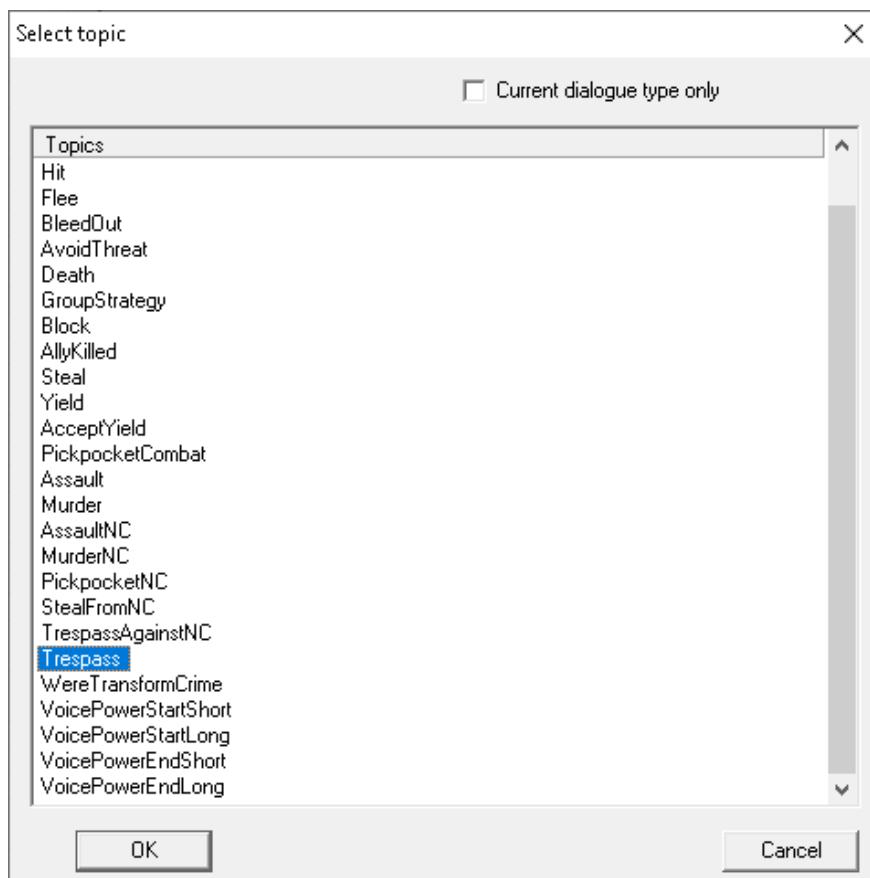


Figure 831 - Selecting the Trespass topic.

Enter in the topic ID then click OK. For this example I just went with WTDialogueGenericTrespass.

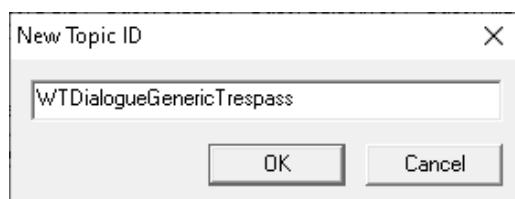


Figure 832 - Setting the topic ID.

Click on the Trespass topic to highlight it then right-click in the area to the right and select New.

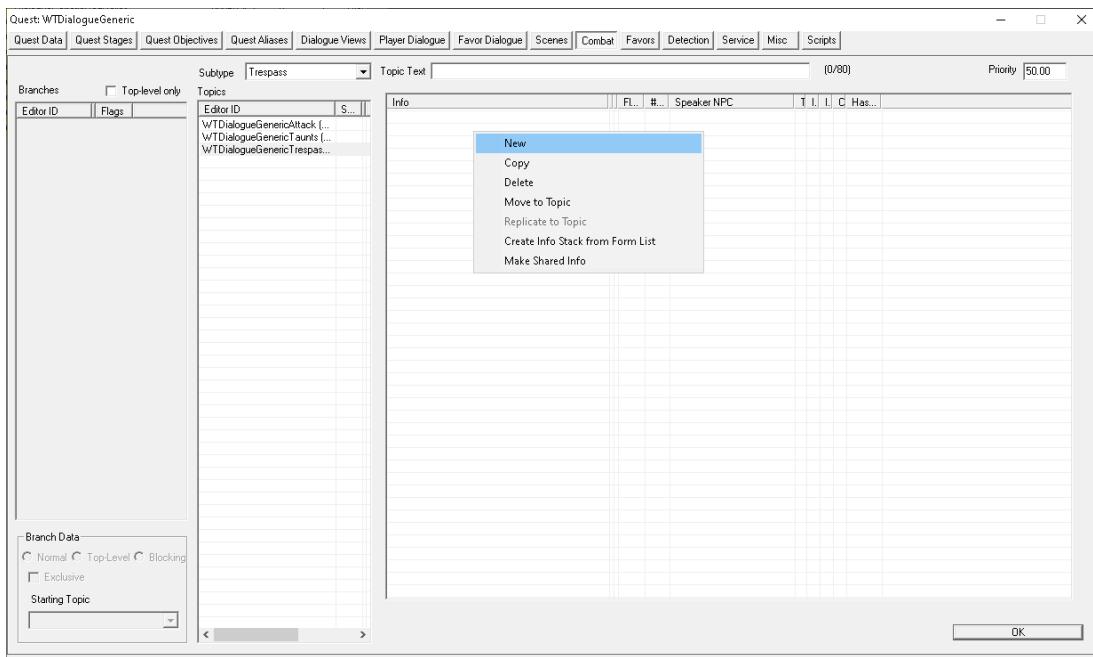


Figure 833 - Adding a trespass response.

Warning level 0 is the first line that's spoken by an NPC when the player is caught trespassing. It should go something along the lines of "You're not supposed to be here".

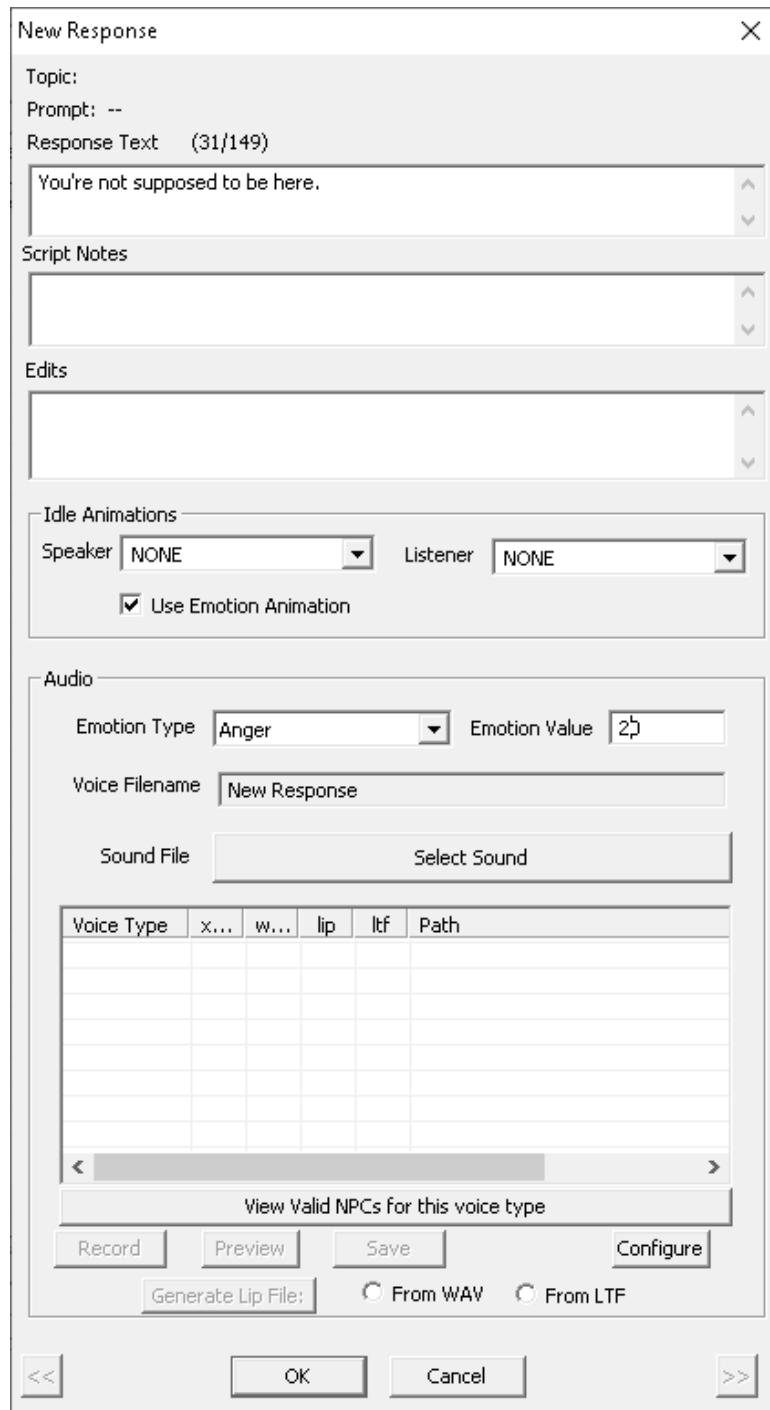


Figure 834 - Response Text for a level 0 trespass warning.

Generally we won't need to tick the Random button unless you're setting up multiple responses per warning level.

We'll need to add two additional conditions for each trespass warning.

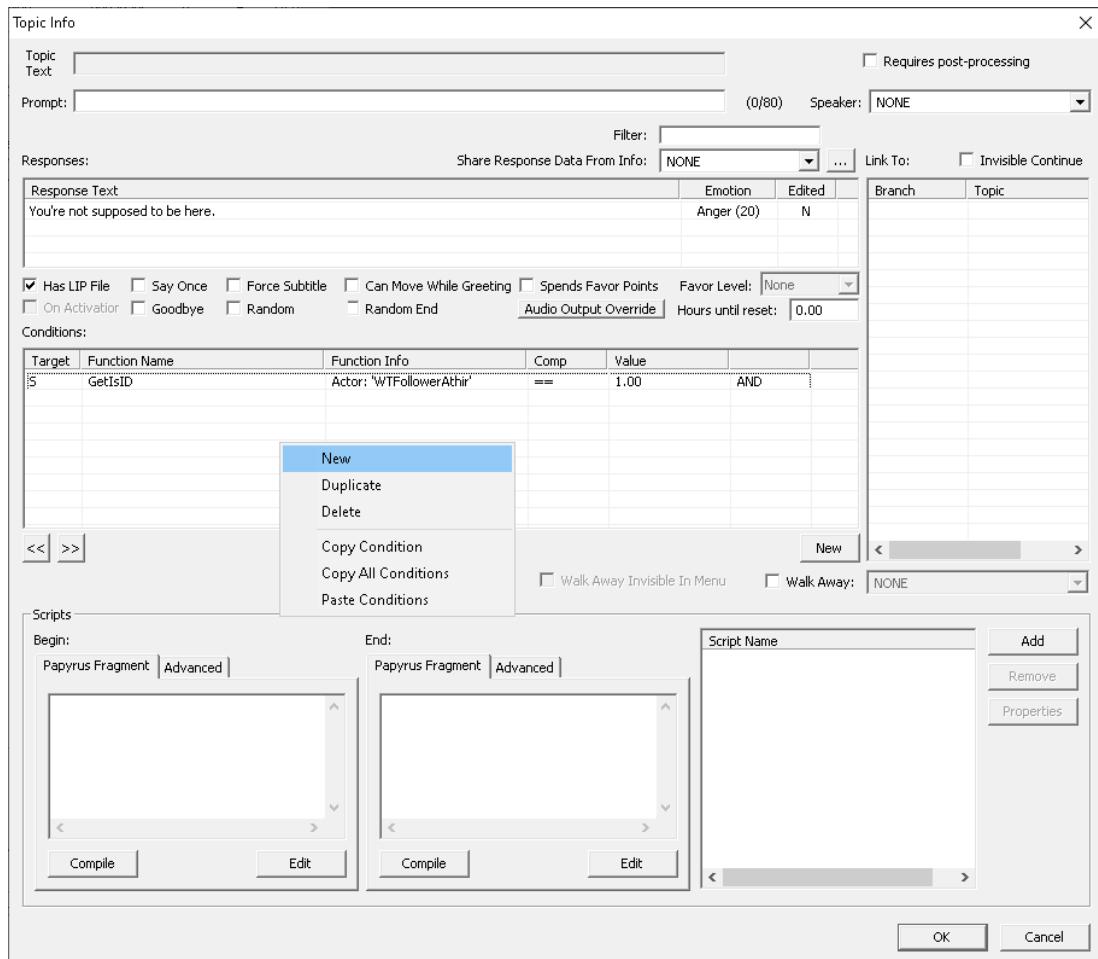


Figure 835 - Adding the first trespass condition.

The first condition to check whether the player is trespassing. We can use the IsTrespassing condition function to do this.

Set the Value field to 1 then click OK.

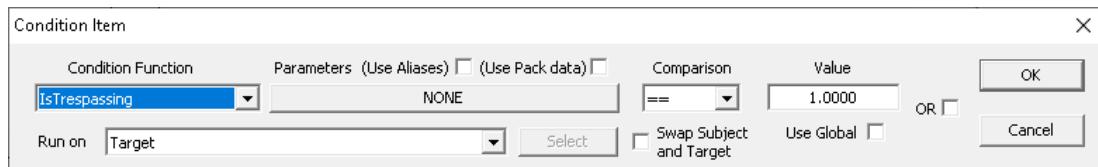


Figure 836 - IsTrespassing condition function.

The second condition is used to check the current trespass warning level. We can use the GetTrespassWarningLevel condition function to do this.

For the first trespass warning, set the Value field to 0 then click OK.

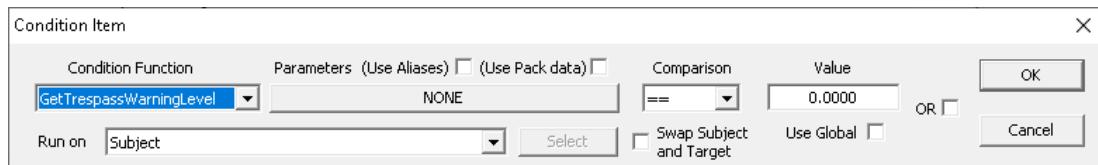


Figure 837 - GetTrespassWarningLevel condition function.

Our first trespass warning should look something like this:

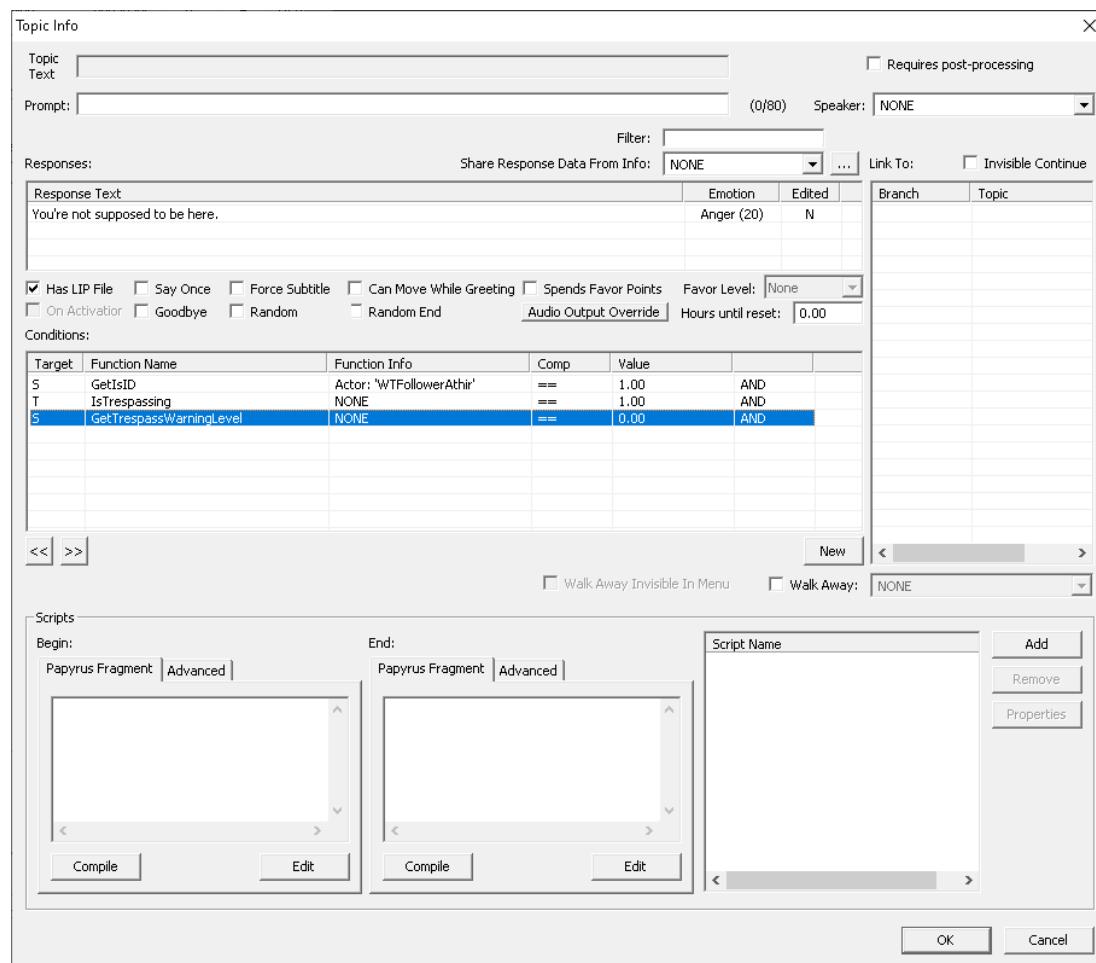


Figure 838 - Level 0 trespass warning set up.

The level 1 trespass warning comes a few moments after the first warning if the NPC detects that the player is still trespassing.

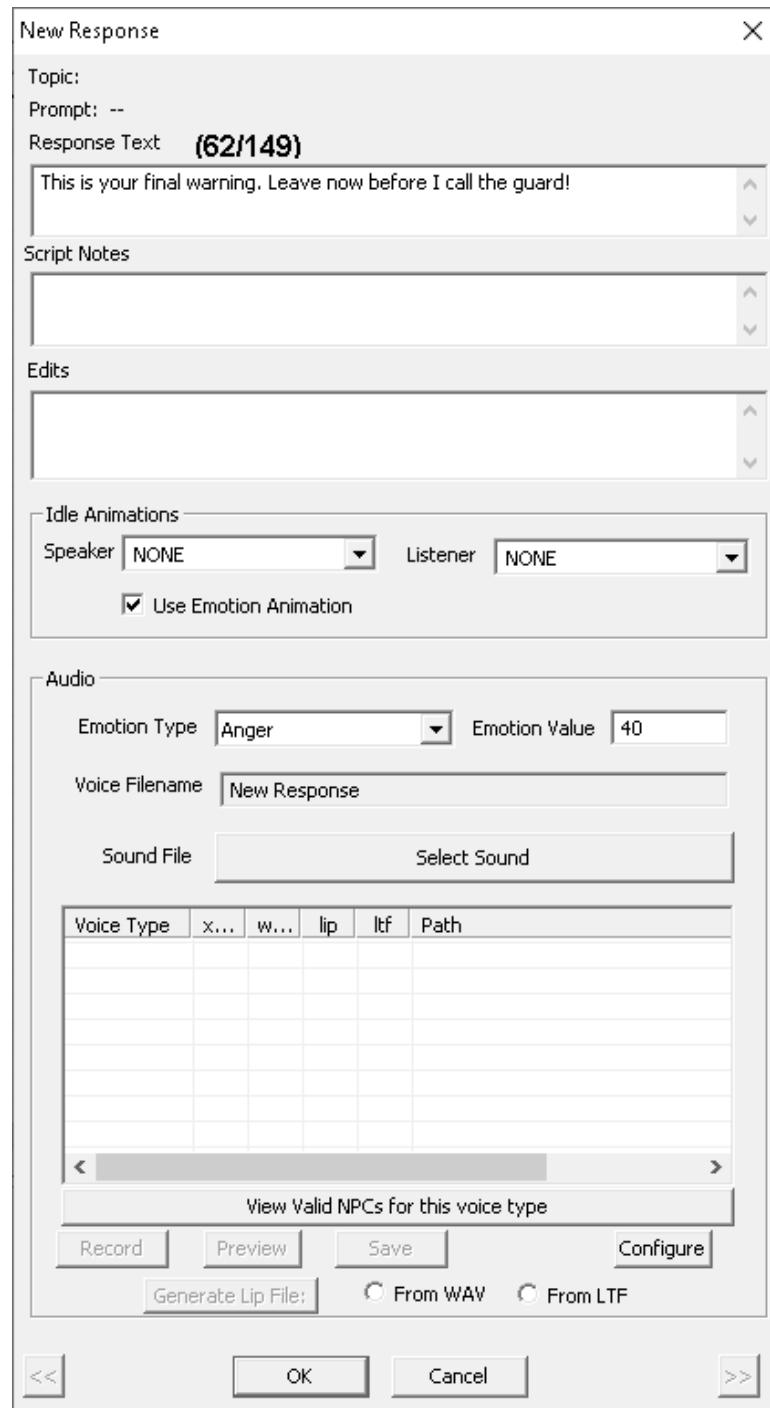


Figure 839 - A level 1 trespass warning.

For the level 1 warning, set the GetTrespassWarningLevel condition value to 1.

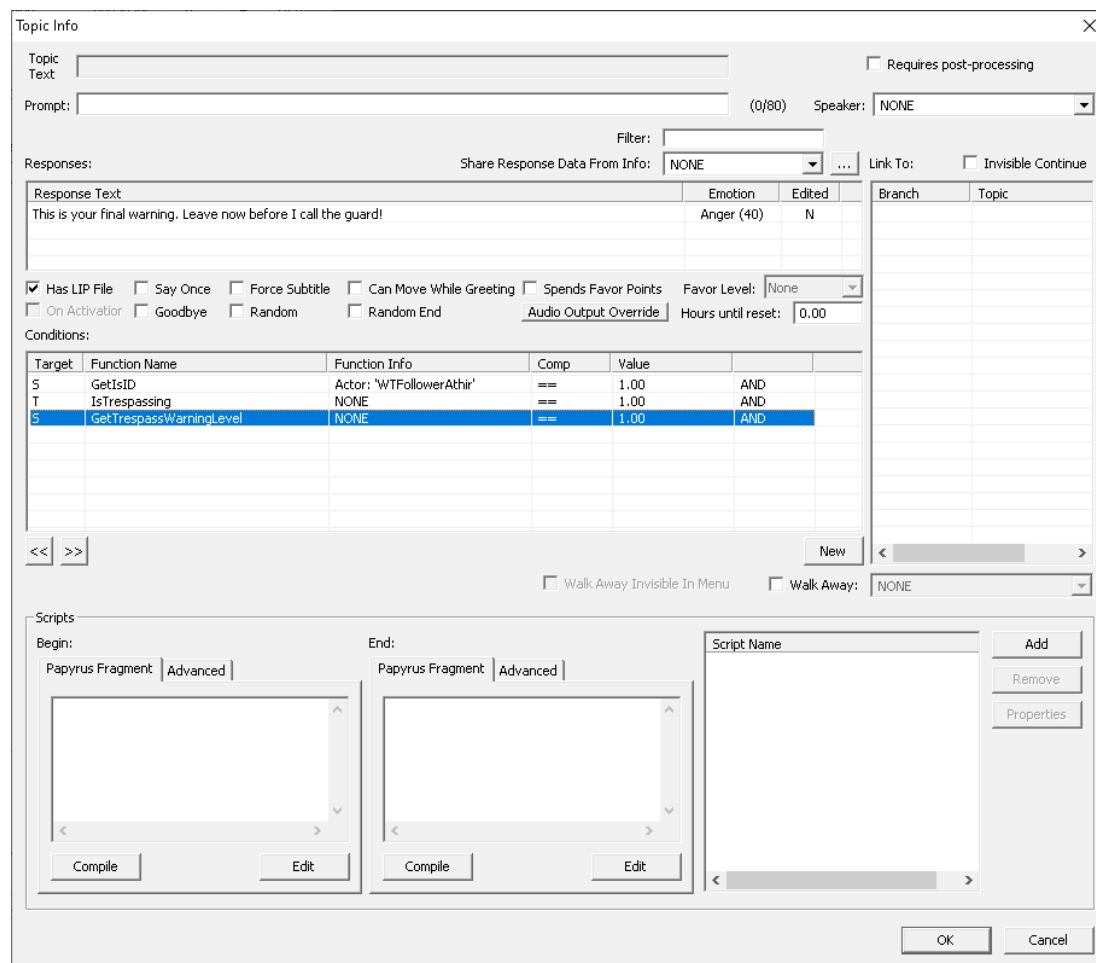


Figure 840 - Level 1 trespass warning set up.

Last is the level 2 warning at which point the guards are called to arrest the player.

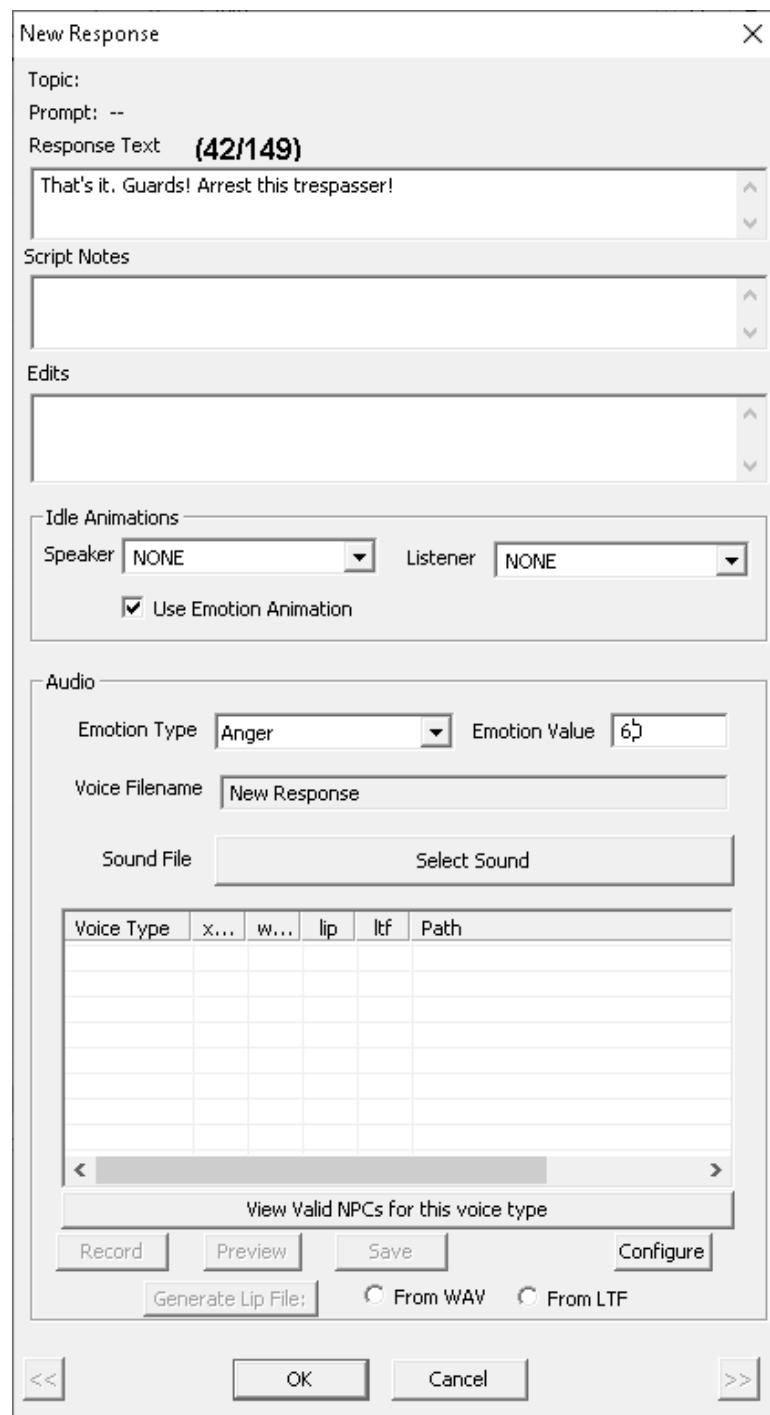


Figure 841 - Level 2 trespass warning.

Set the GetTrespassWarningLevel condition value to 2.

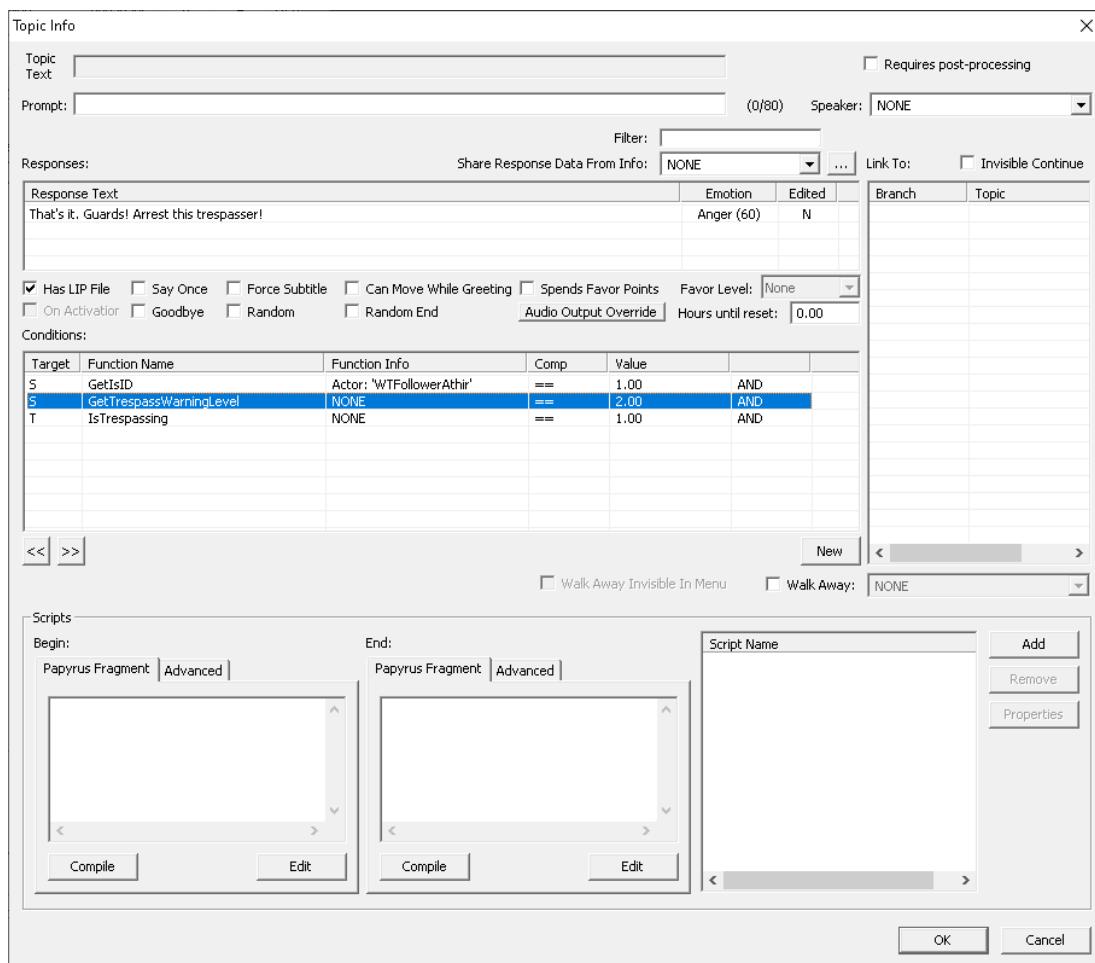


Figure 842 - Level 2 trespass warning set up.

The screenshot below shows trespass warnings set up for NPCs in Wyrmstooth. Each NPC has three trespass lines, one for each warning level.

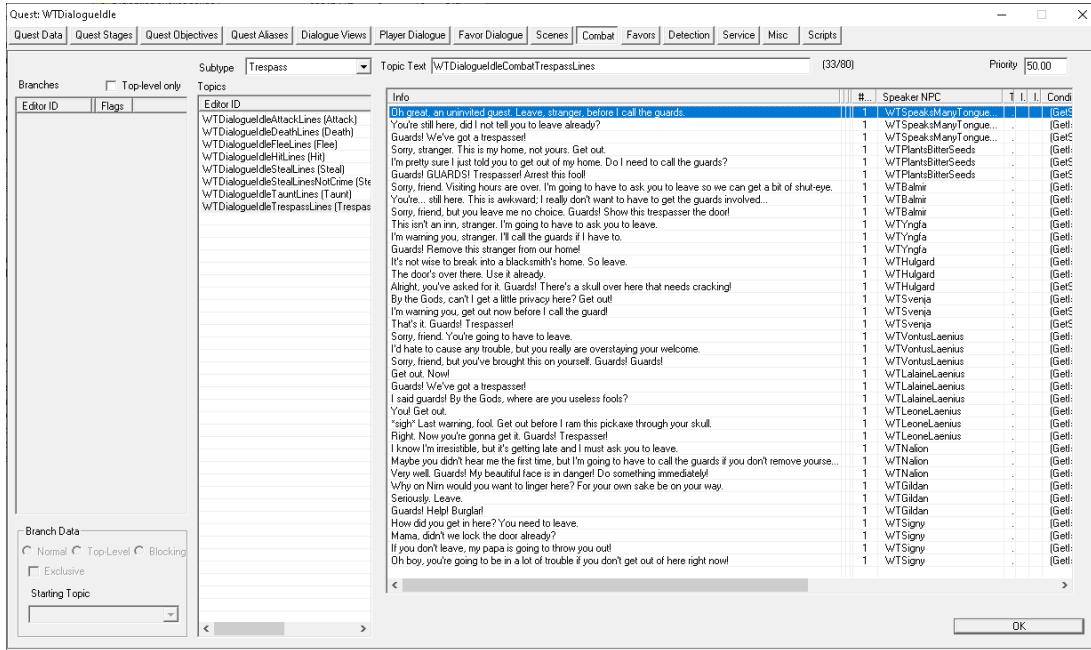


Figure 843 - Sample trespass warnings.

For more information on each topic type see the [Combat](#) tab article on the Creation Kit wiki.

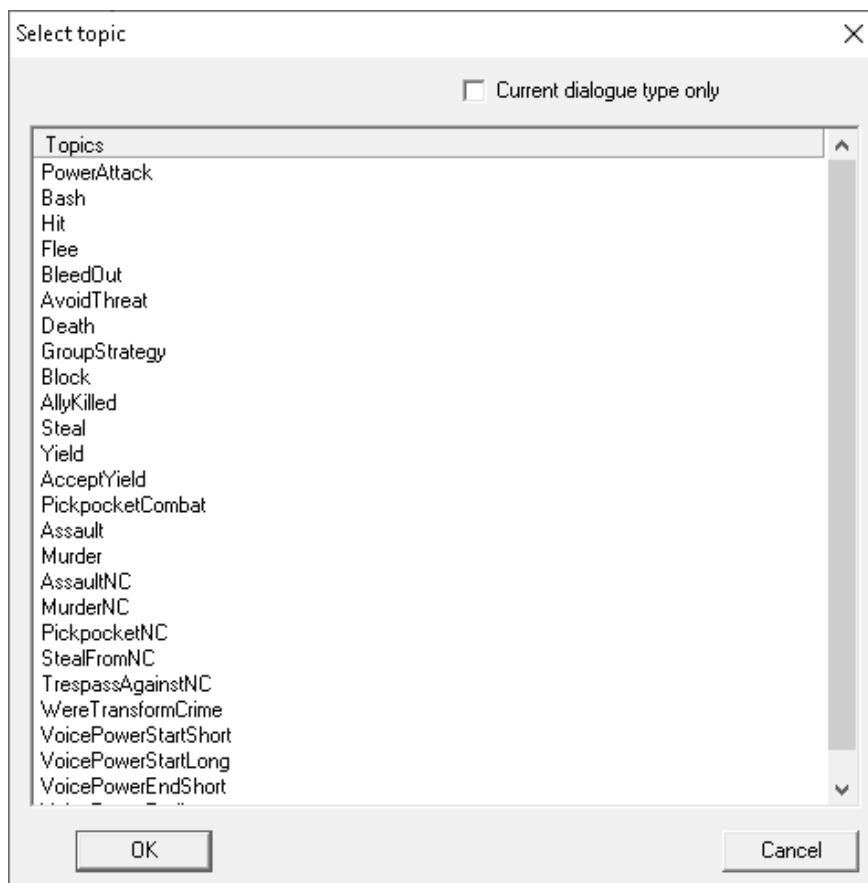


Figure 844 - List of topic types for the Misc tab.

AllyKilled, AvoidThreat, GroupStrategy and Yield aren't used. If you want the NPC to respond to a yield request with custom dialogue, use AcceptYield instead.

ADDING SCENES

Scenes are used to carefully choreograph NPC actions. Scenes are divided into phases and an NPC can be assigned a different package or dialogue action on each phase.

Before a phase can end, all actions defined in a phase need to complete.

For example, the intro carriage ride at the start of the game is set up using a scene.

In Wyrmsooth, a scene is triggered when you encounter Vulthurkrah for the first time on Ancient's Ascent.

In this section I'll be showing you how to set up a simple scene involving a couple NPCs.

Note: I'll be reusing the same sample quest WTTestQuest that I set up in the previous sections of this guide. If you want to create a new quest for this scene, just be sure to generate a new .seq file once this is done.

Go to the Scenes tab in Quest properties.

Right-click in the scenes list on the left hand side and select New.

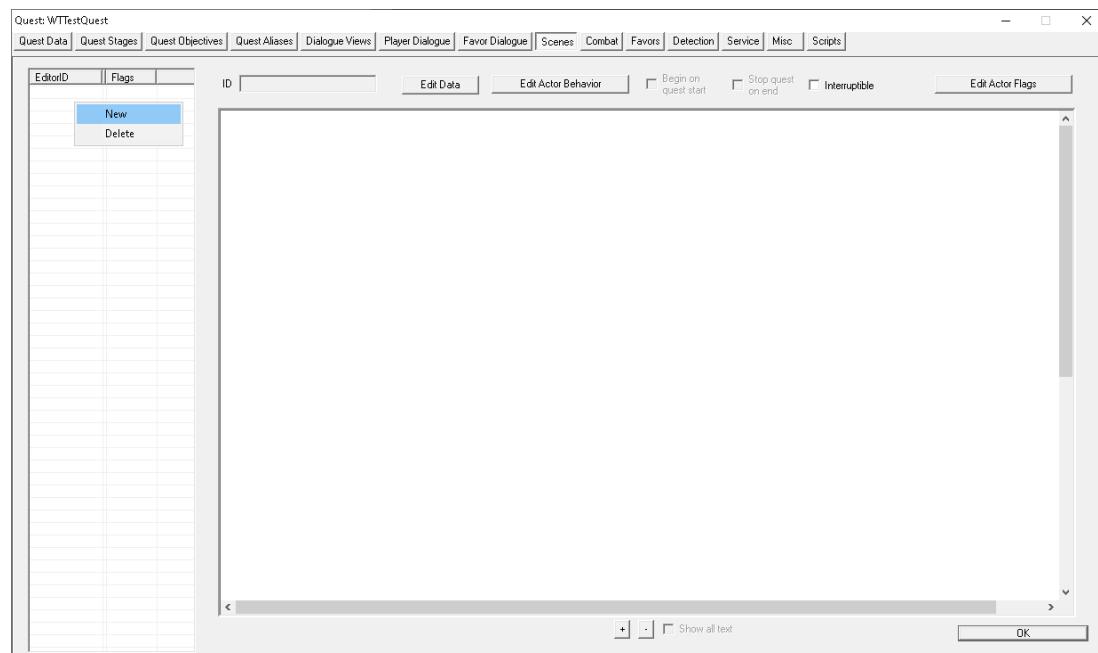


Figure 845 - Scenes tab.

Enter in a scene ID then click OK. For this example, I'm just going to call it WTTTestQuestSampleScene.

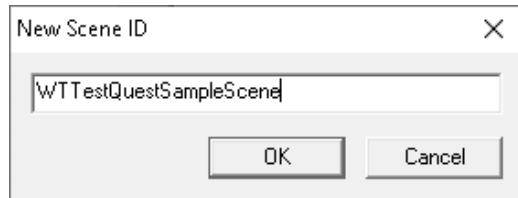


Figure 846 - Scene ID.

Click on the new scene in the scenes list to highlight it, then right-click in the scene layout section to the right and select New Actor.

Note: Only NPCs added to your Quest Alias tab can be chosen here.

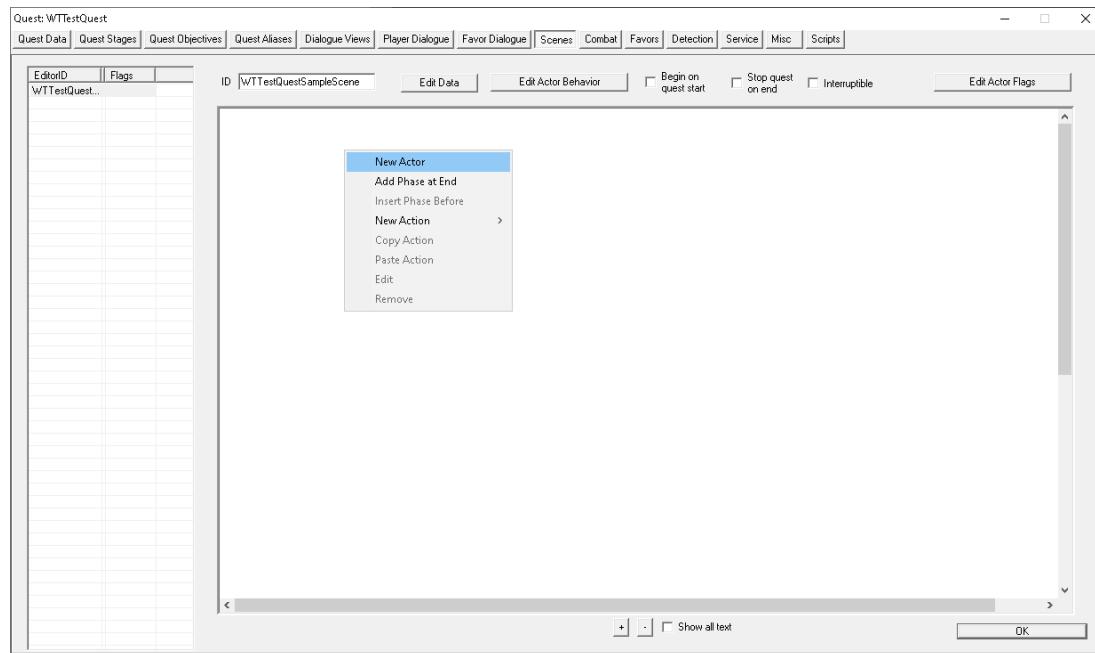


Figure 847 - Adding a new actor.

Select the NPC to add from the list of quest aliases then click OK.

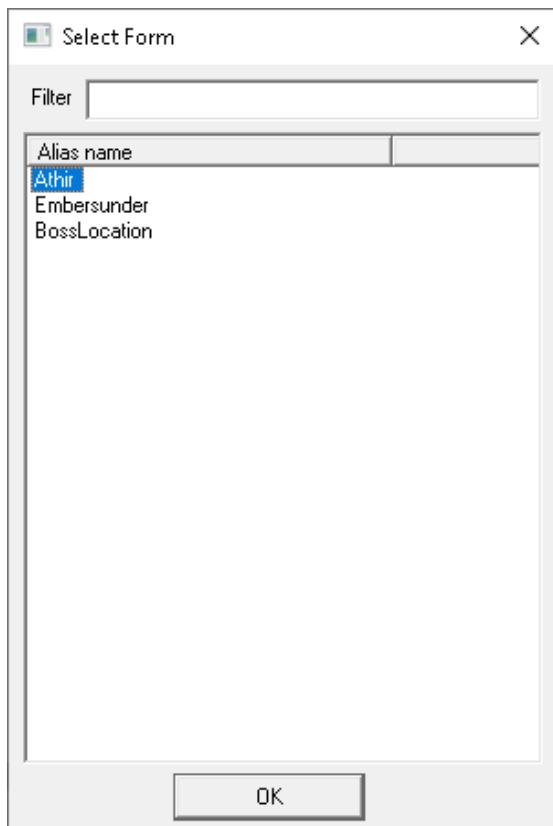


Figure 848 - Selecting the first NPC.

The NPC should appear in the scene layout as per the screenshot below:

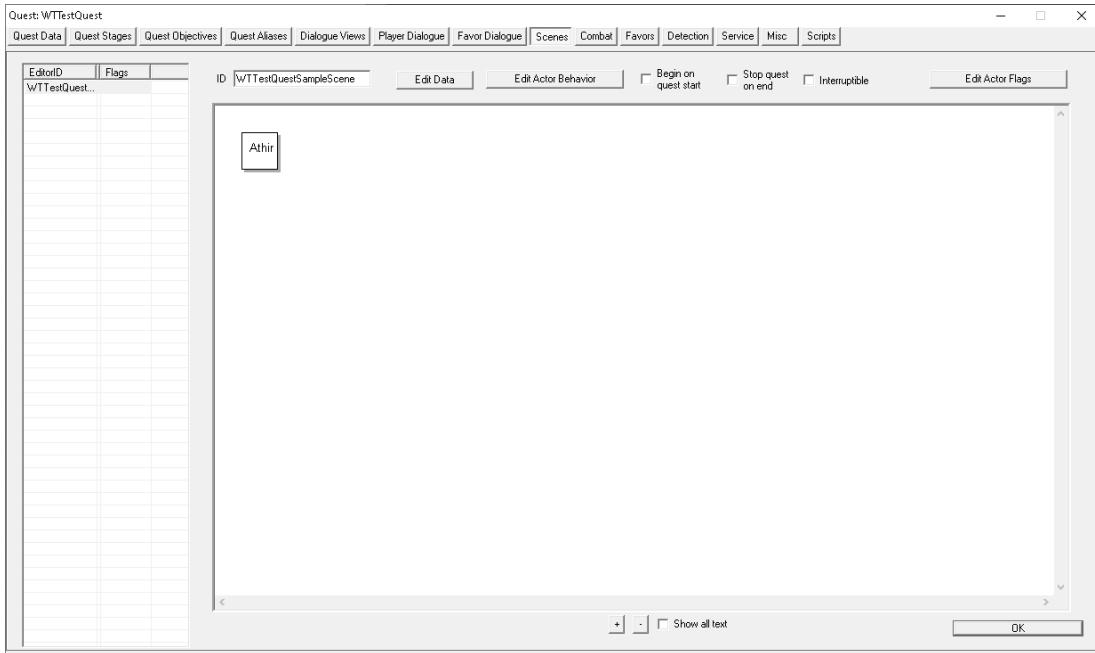


Figure 849 - An NPC added to the current scene.

Our scene will need to consist of at least one phase. To add a phase, right-click and select ‘Add Phase at End’.

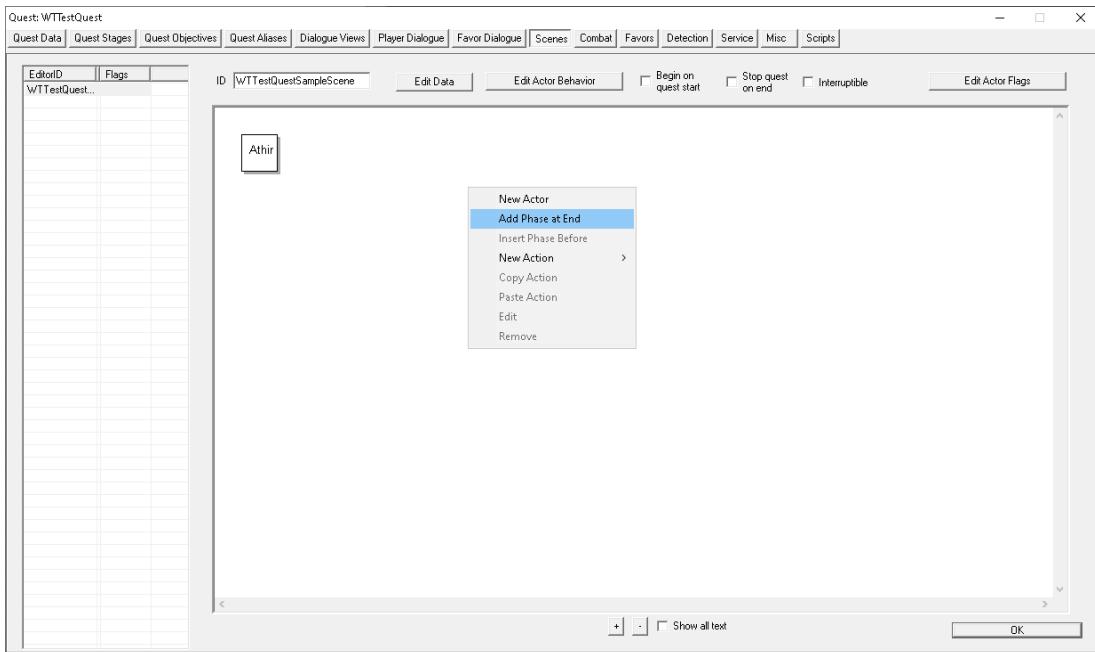


Figure 850 - Adding the first phase.

Our scene layout should now look like this.

Currently we haven't assigned any actions to Athir, so let's give him something to do next.

In this example, I'm going to give him some banter with the other mercenaries following the player through Wyrmsooth Barrow.

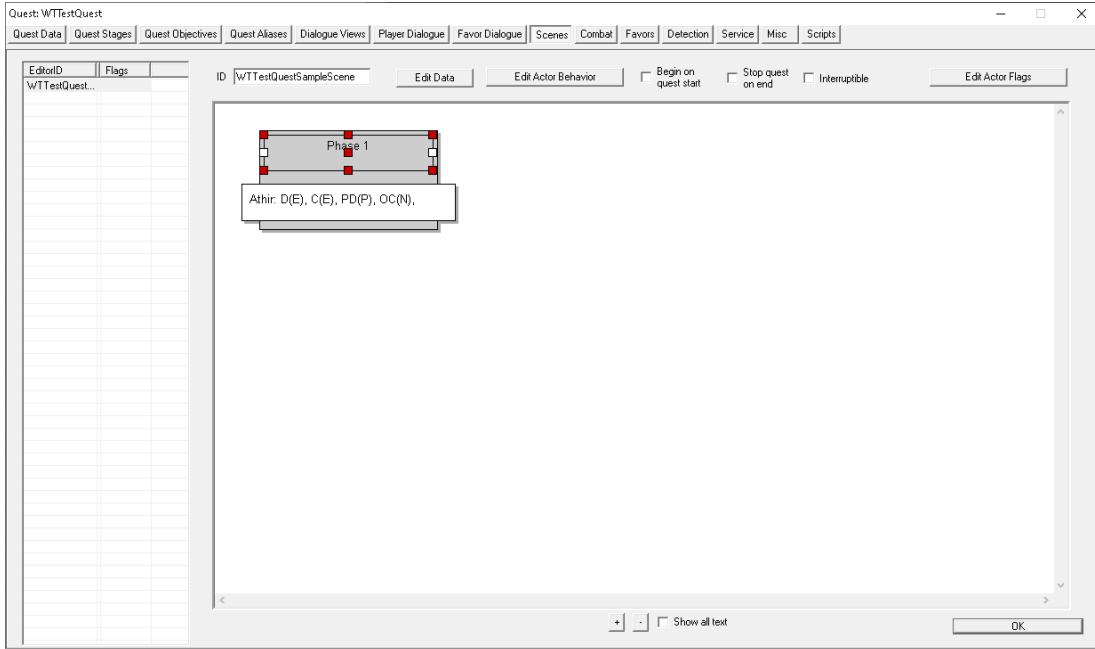


Figure 851 - First phase added.

I added a couple XMarkerHeadings in the render window that I'll be using in this example scene to tell the NPCs involved where to stand.

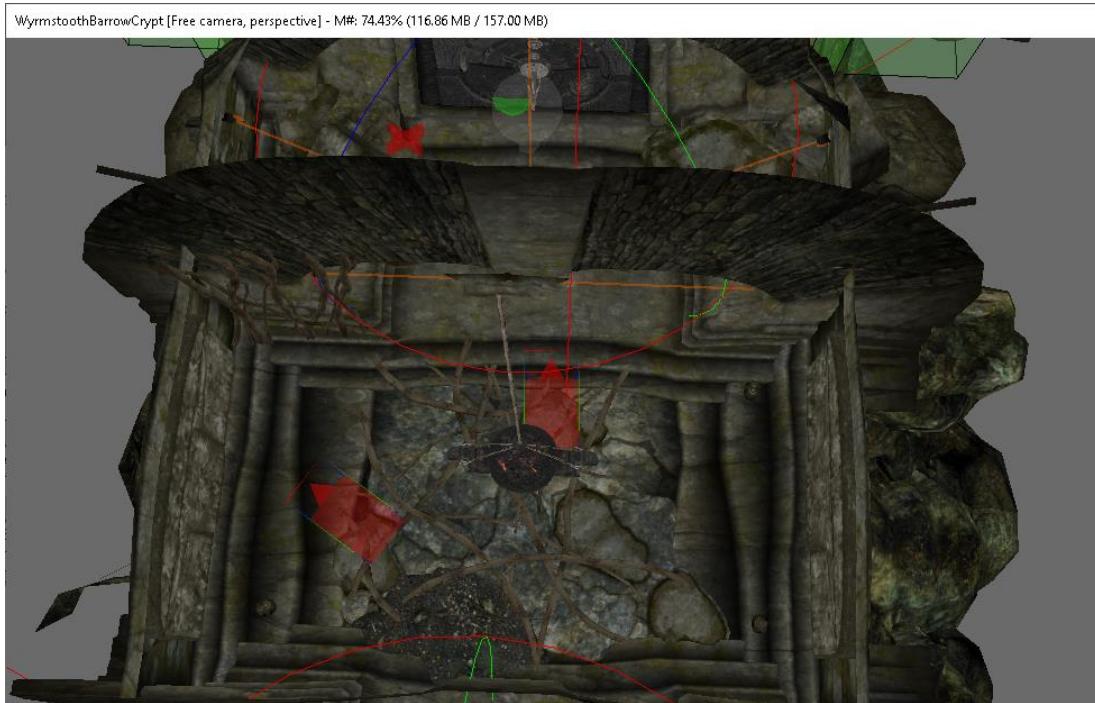


Figure 852 - XMarkerHeadings that'll be used for the example scene.

To tell Athir where to stand when the scene starts up, right-click on him in the scene layout and go to New Action > Package.

Important: Avoid assigning non-completing packages to NPCs in a scene. Some packages such as Sandbox never complete, so the phase they're assigned on won't complete and the scene will stall.

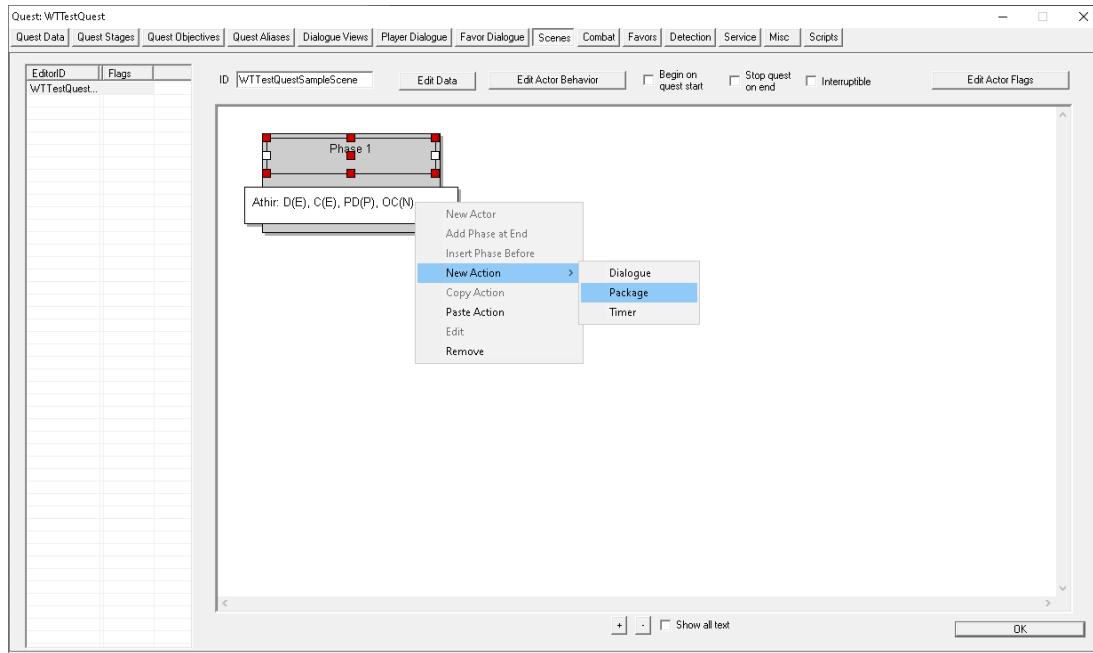


Figure 853 - Adding a package to Athir for phase 1.

Right-click in the package list and select New.

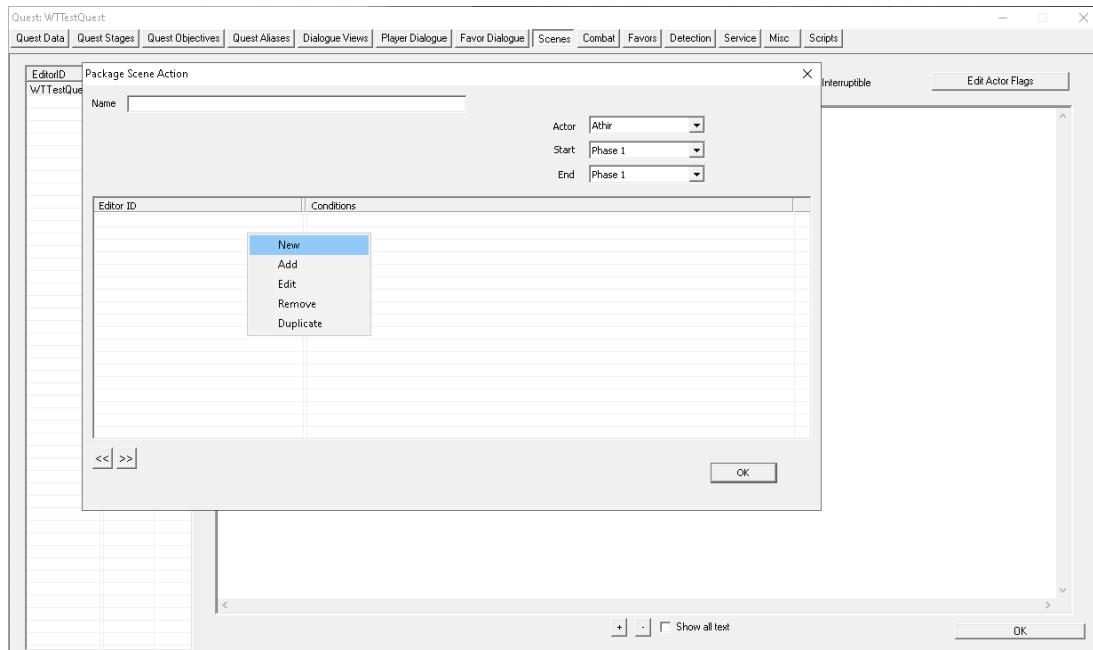


Figure 854 - Creating a new package.

Enter in the package ID. I just called it WTTTestSceneAthirWalkToPackage for this example.

Make sure the ‘Owner quest’ drop-down is set to the name of the quest you’re setting up the scene in.

In this example, I set up a new package using the Travel template and made Athir travel to the XMarkerHeading facing the left wall.

Click OK to add the new package to Athir for phase 1 of the scene.

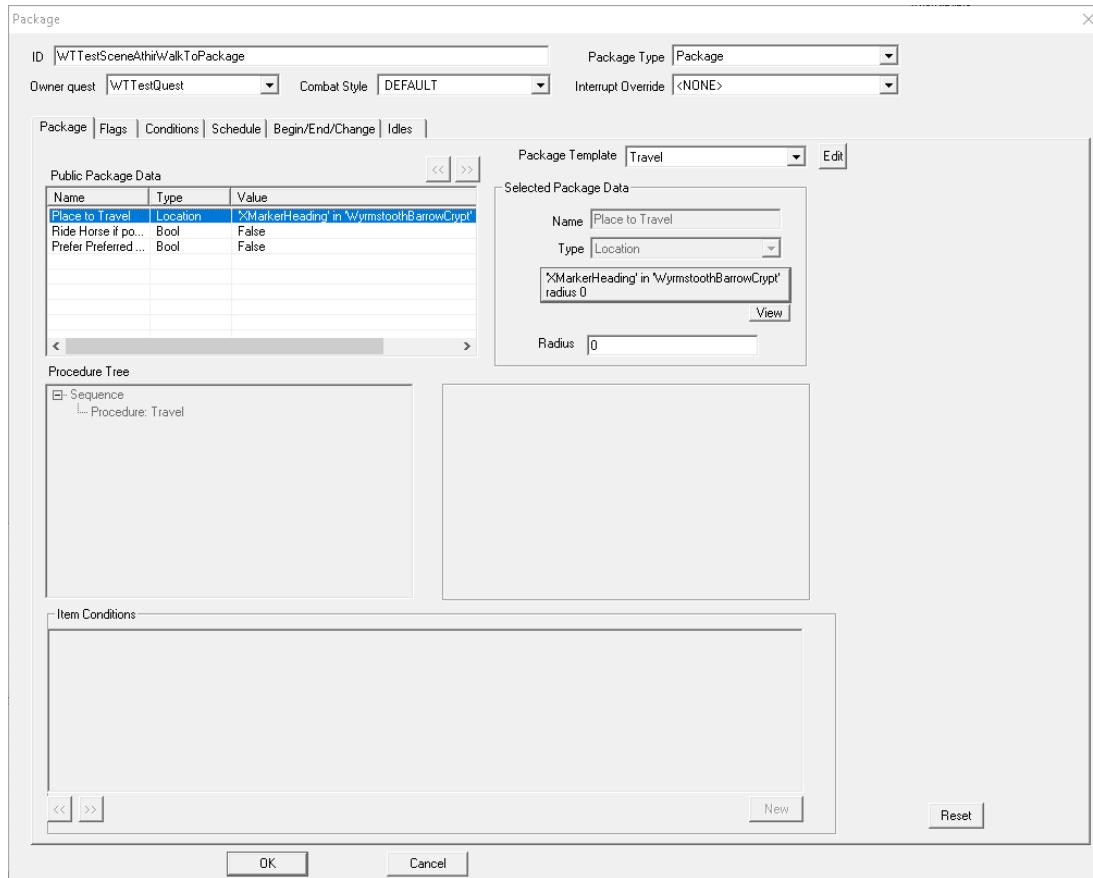


Figure 855 - Athir's scene package.

Click OK to close out of Package Scene Action properties.

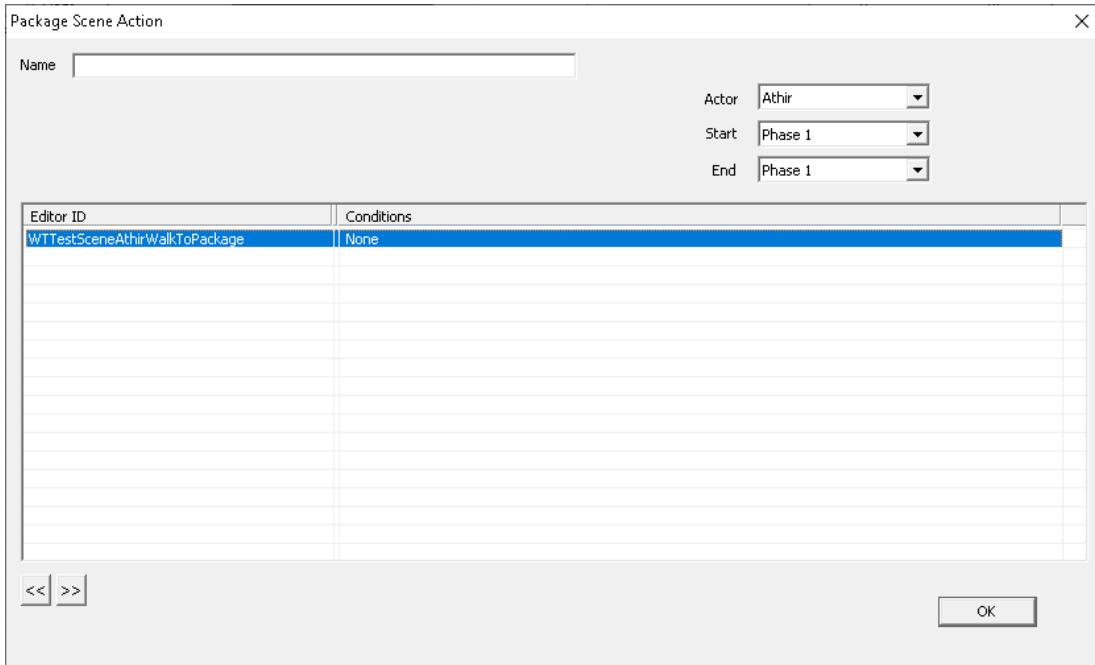


Figure 856 - Package Scene Action properties.

Now Athir has a package for phase 1.

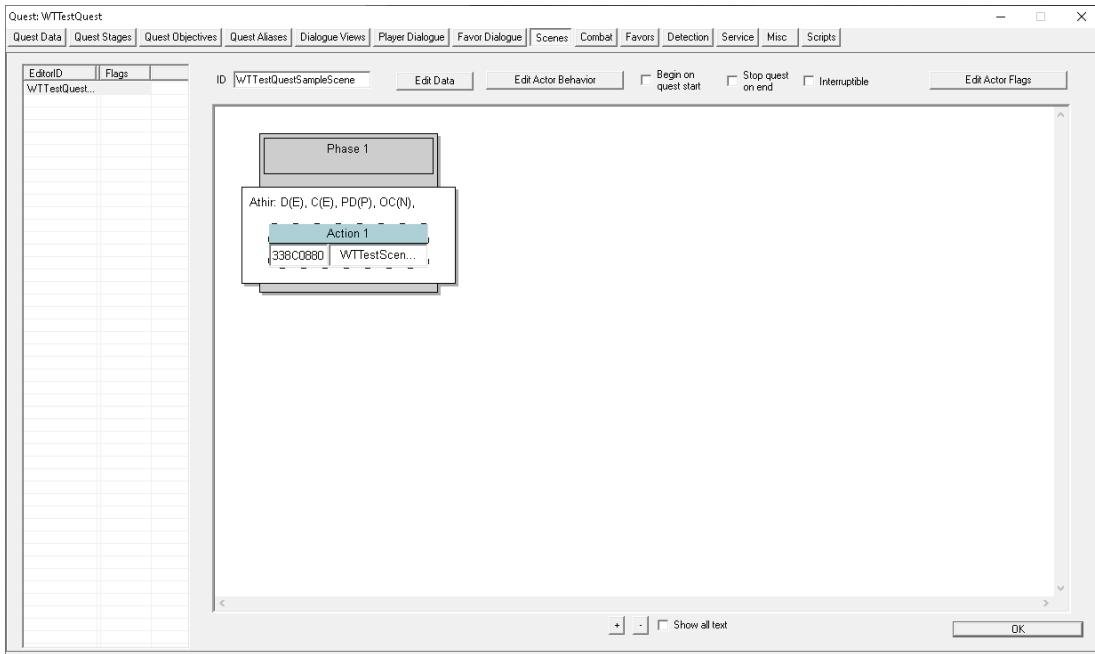


Figure 857 - Athir with a package set up for phase 1.

Next, let's give him something to say while he walks over to the XMarkerHeading.

Right-click on Athir again. This time select New Action > Dialogue.

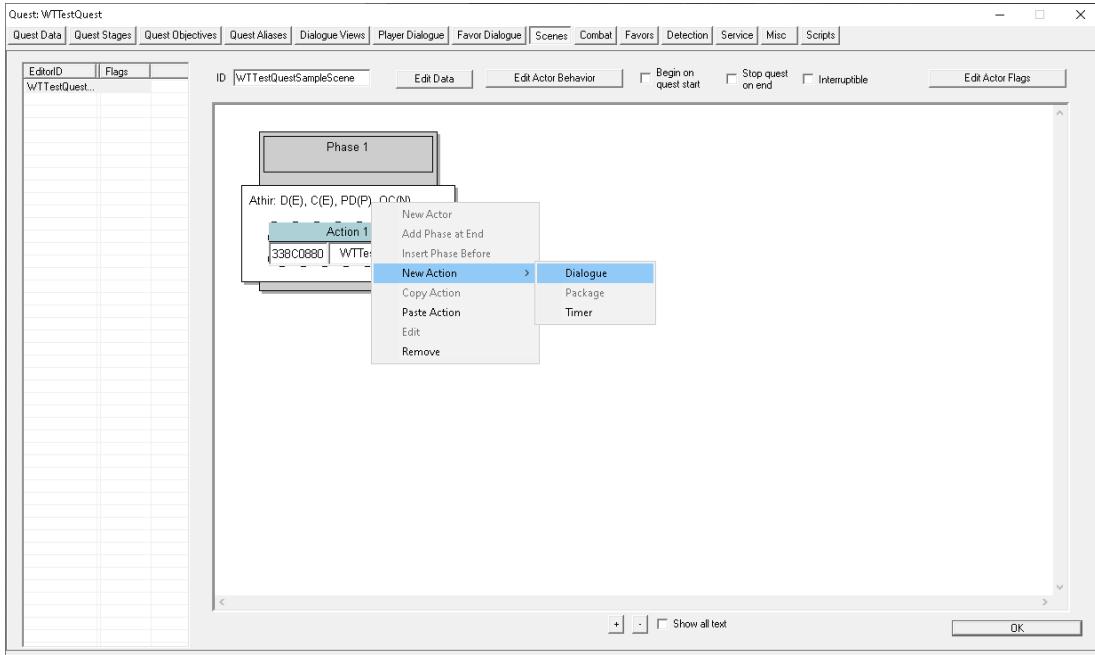


Figure 858 - Adding a new dialogue scene action.

Right-click in the dialogue list and select New.

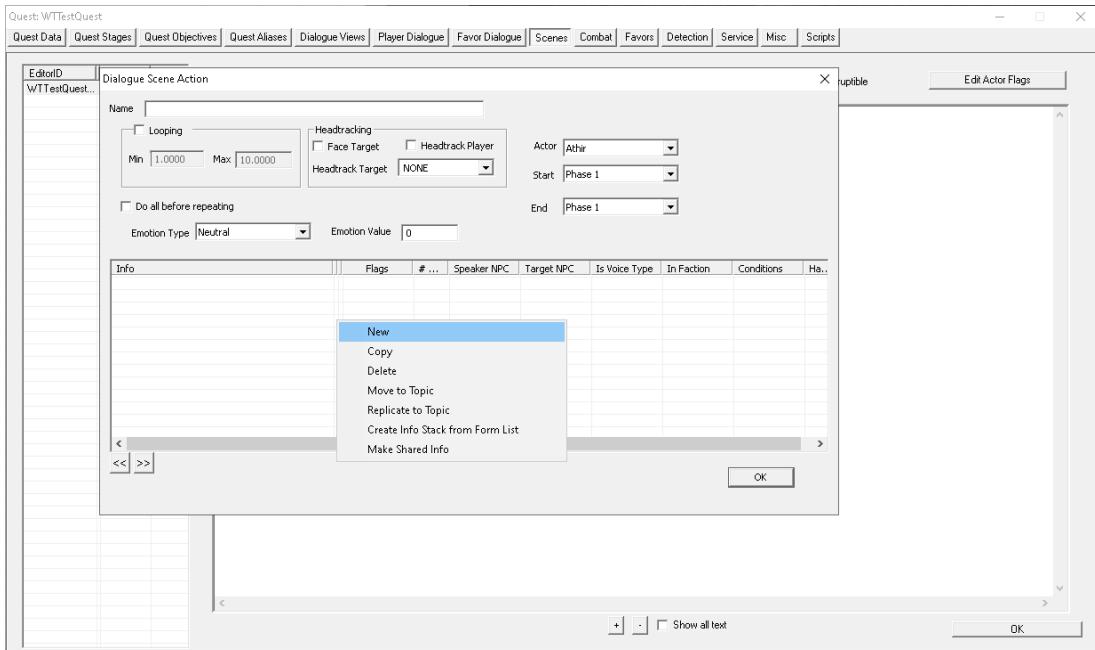


Figure 859 - Adding a new line of dialogue for phase 1.

Enter in dialogue in the Response Text field. Change the Emotion Type and Emotion Value if necessary then click OK.

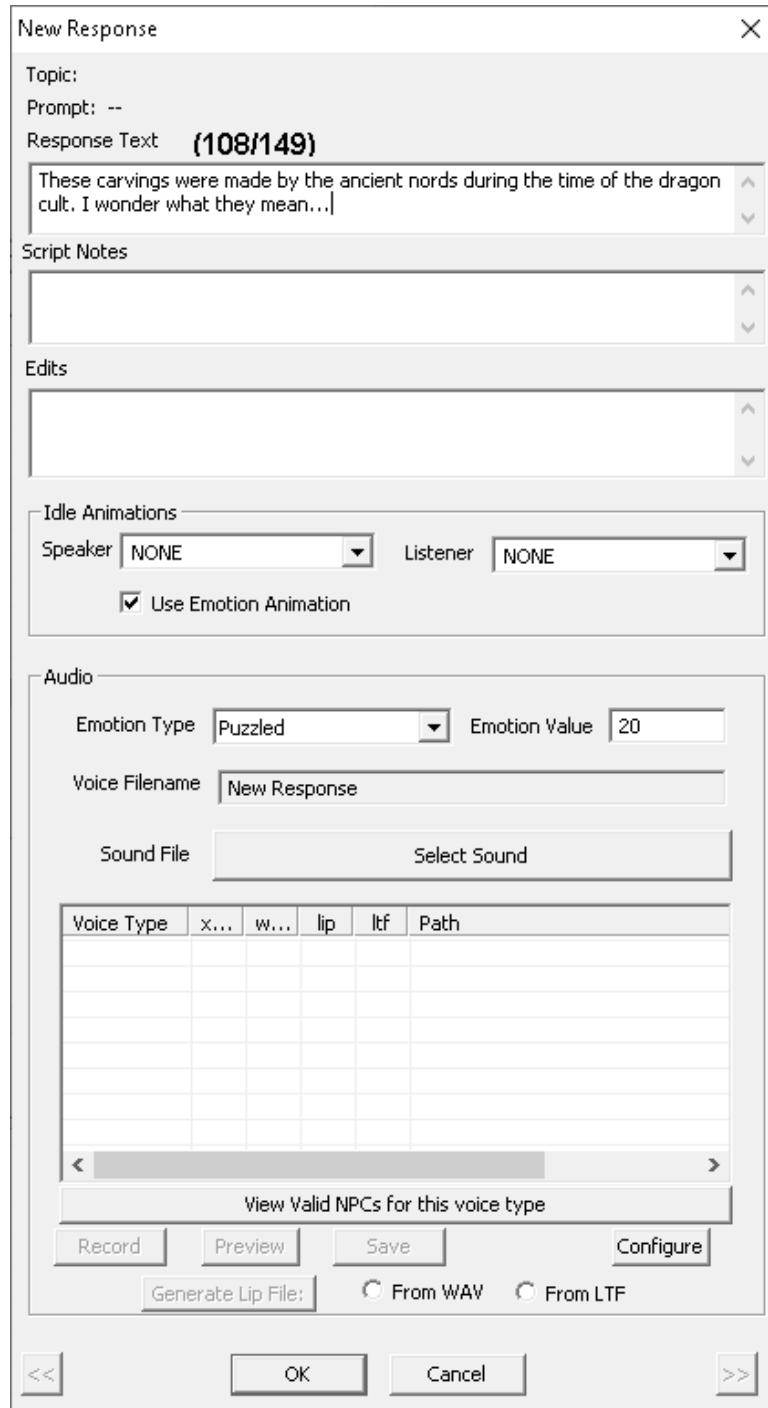


Figure 860 - Adding response text.

Right-click in the conditions list and select New.

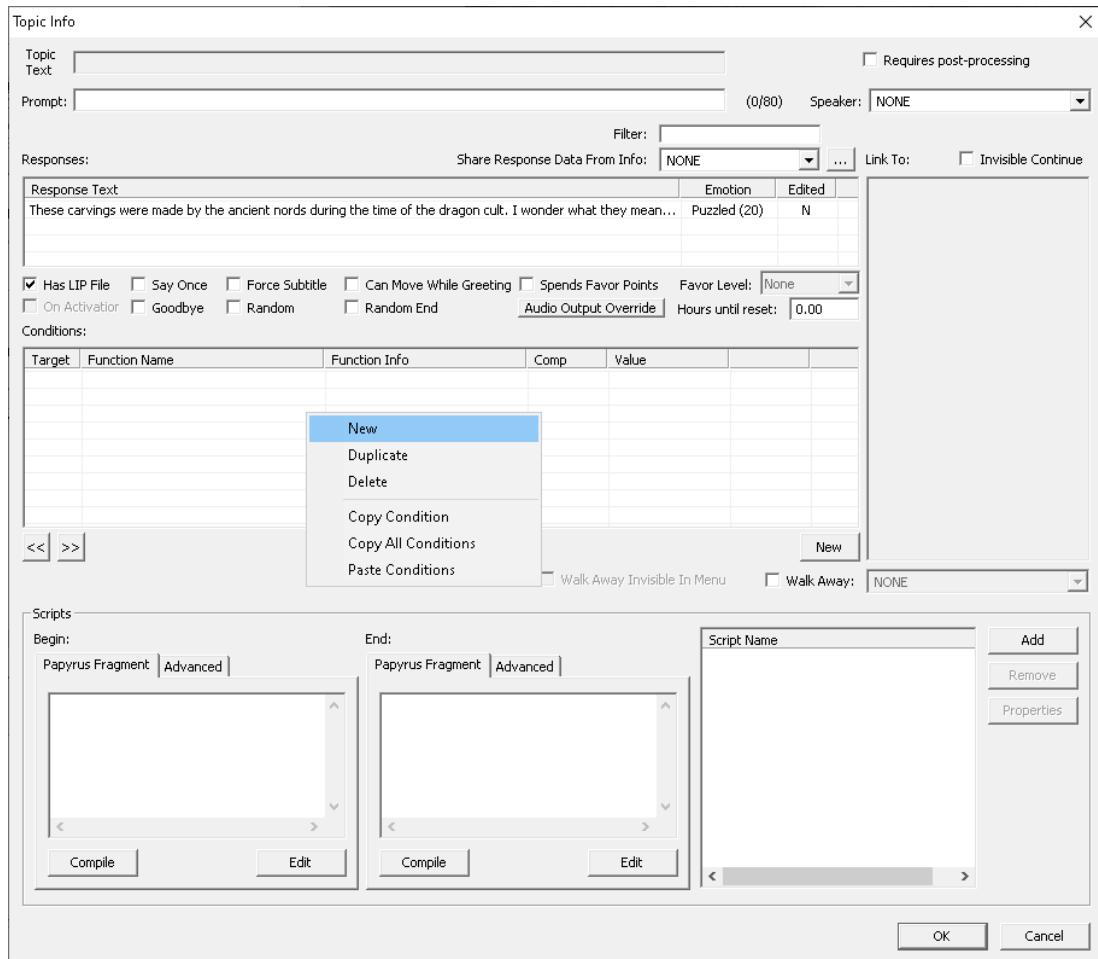


Figure 861 - Adding a new dialogue condition.

I added a GetIsID condition to ensure only Athir can speak this line of dialogue.

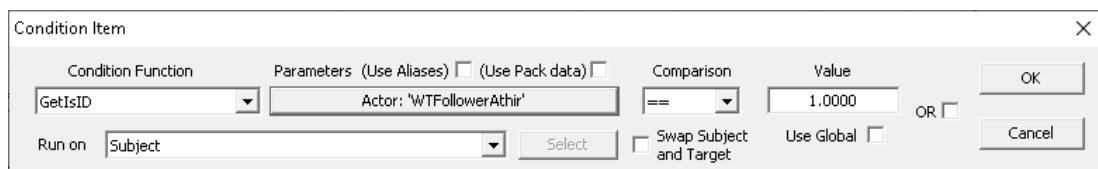


Figure 862 - GetIsID condition.

Once the condition has been added, click OK.

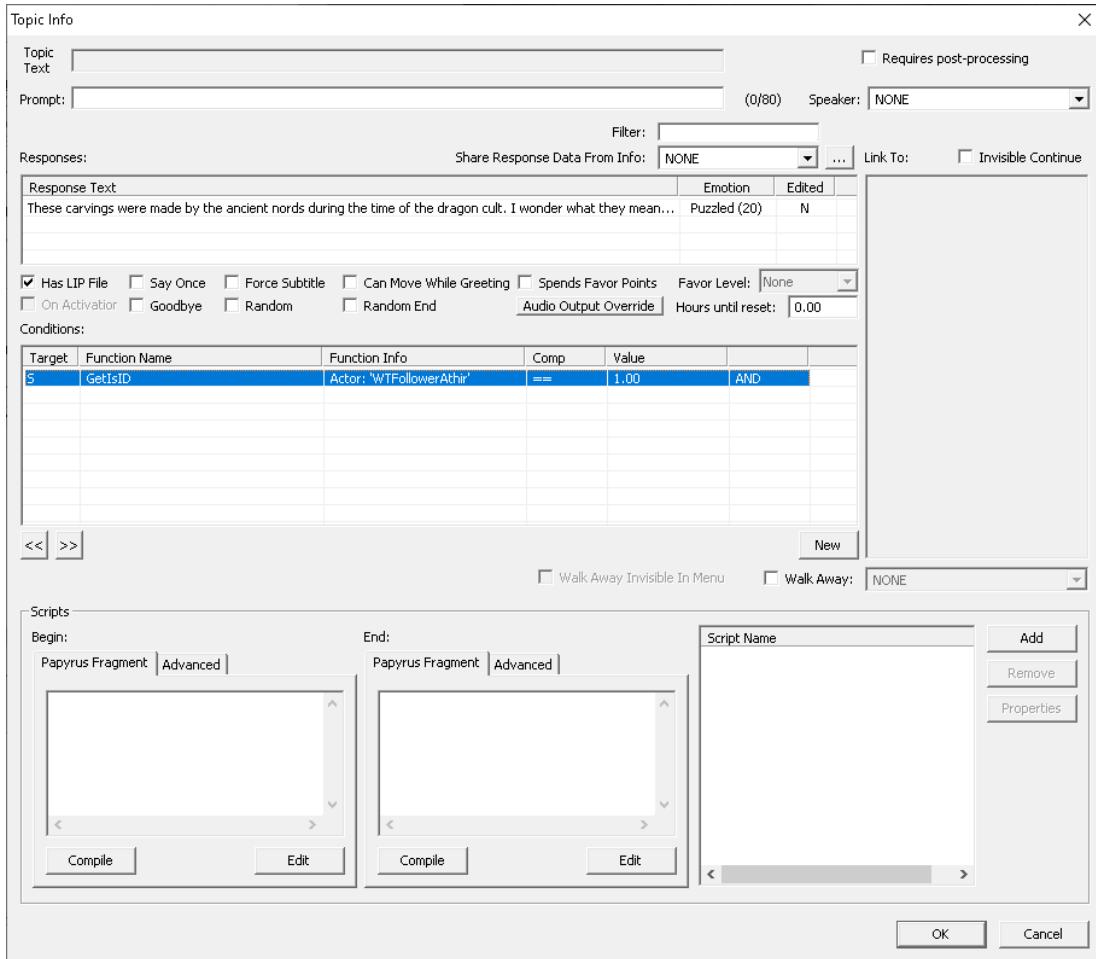


Figure 863 - The first line of dialogue for the scene set up.

Click OK to close out of Dialogue Scene Action properties.

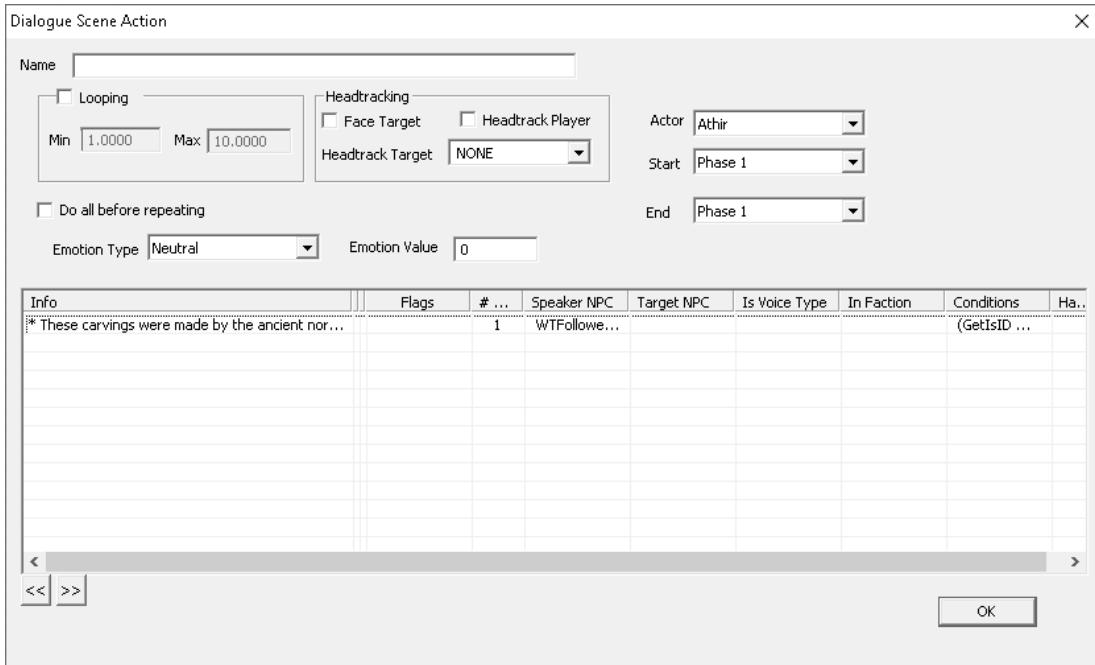


Figure 864 - Dialogue Scene Action with a line of dialogue set up.

When this scene starts up, Athir should now walk over to the XMarkerHeading on the left while commenting about the carvings.

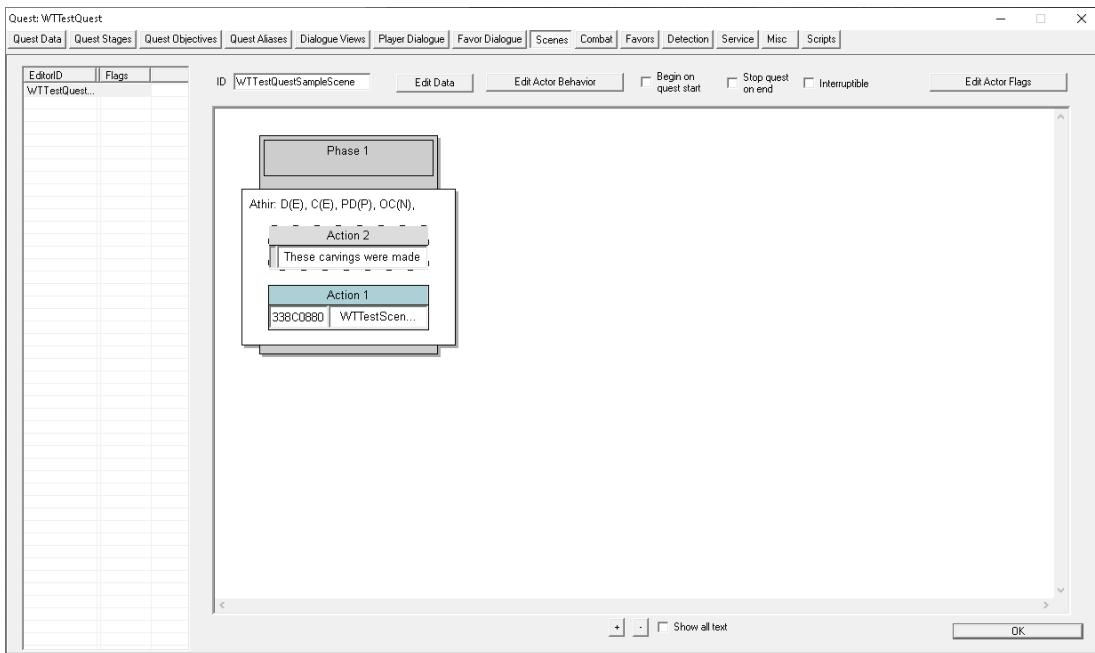


Figure 865 - Phase 1 set up with a line of dialogue and a package.

Let's add in another NPC to get a bit more banter going.

Go to the Quest Aliases tab, right-click in the aliases list and select New Reference Alias.

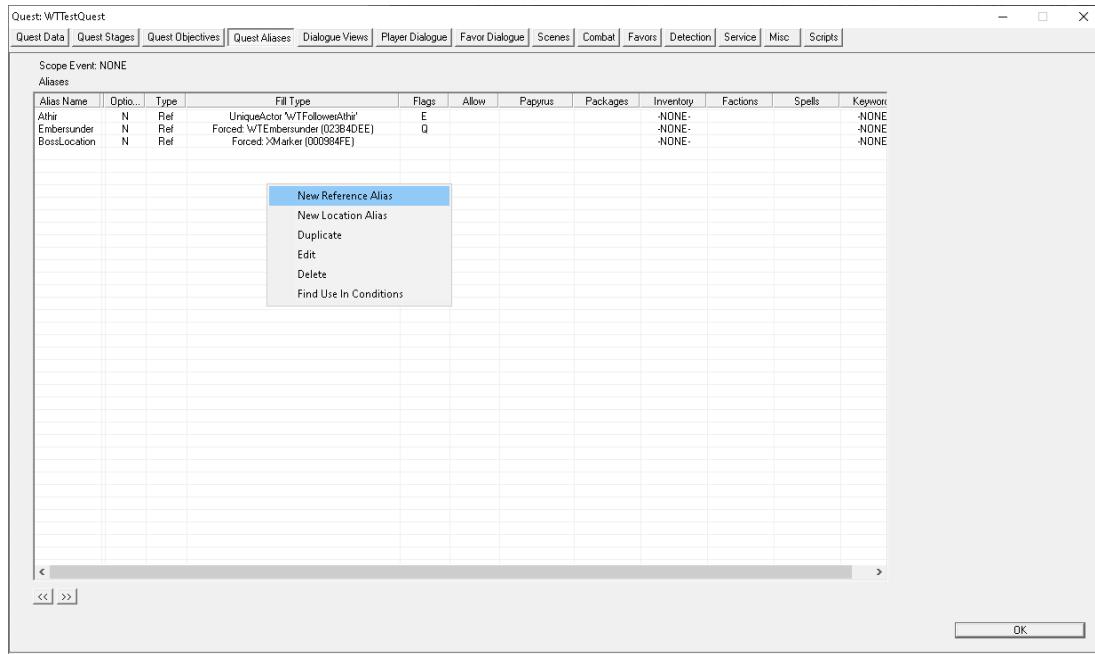


Figure 866 - Creating a new reference alias.

I'm going to add in Daenlit so she has something to say and flag her as essential.

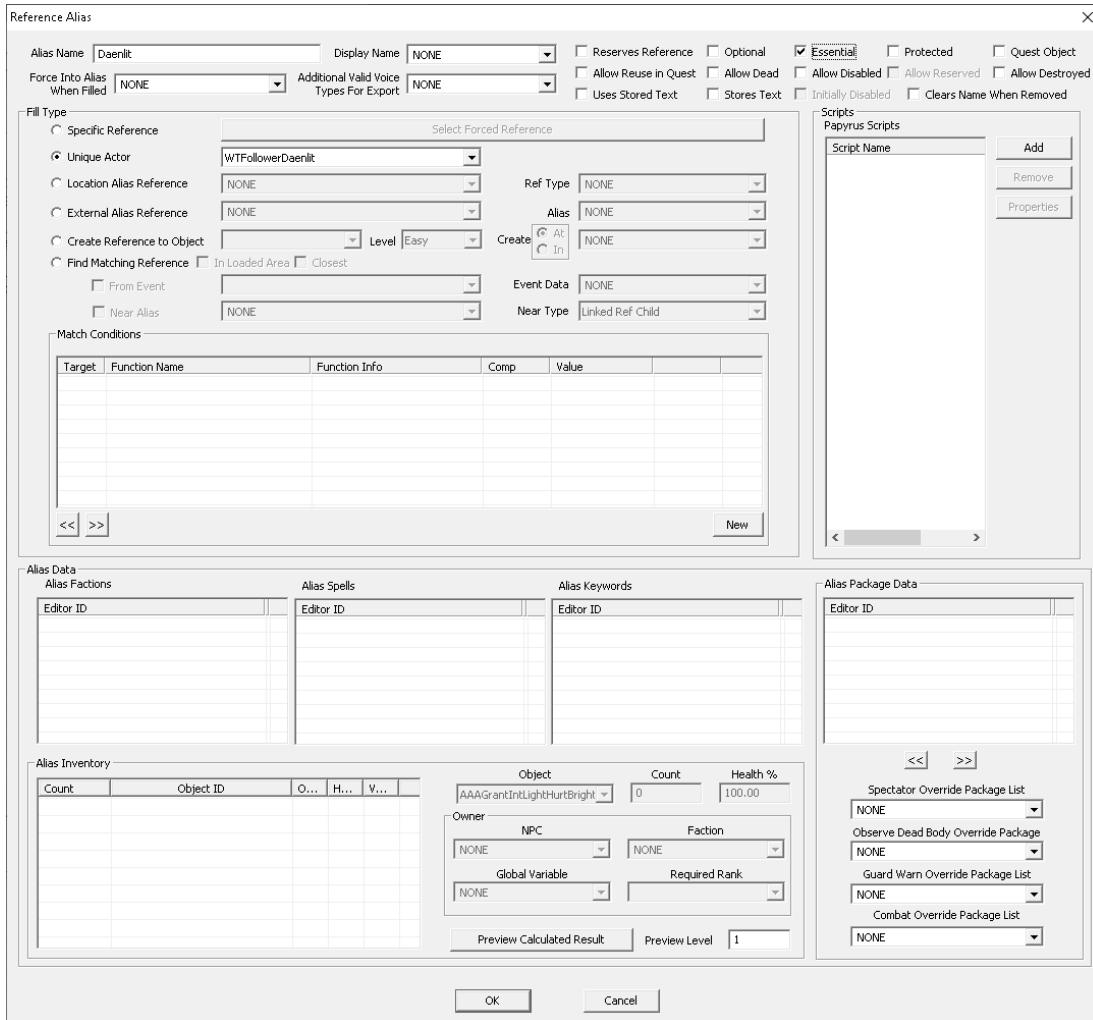


Figure 867 - Setting Daenlit as the reference in the new reference alias.

Click OK to close out of the Reference Alias properties.

Go back to our scene in the Scenes tab.

Right-click to the right of Phase 1 and select ‘Add Phase at End’.

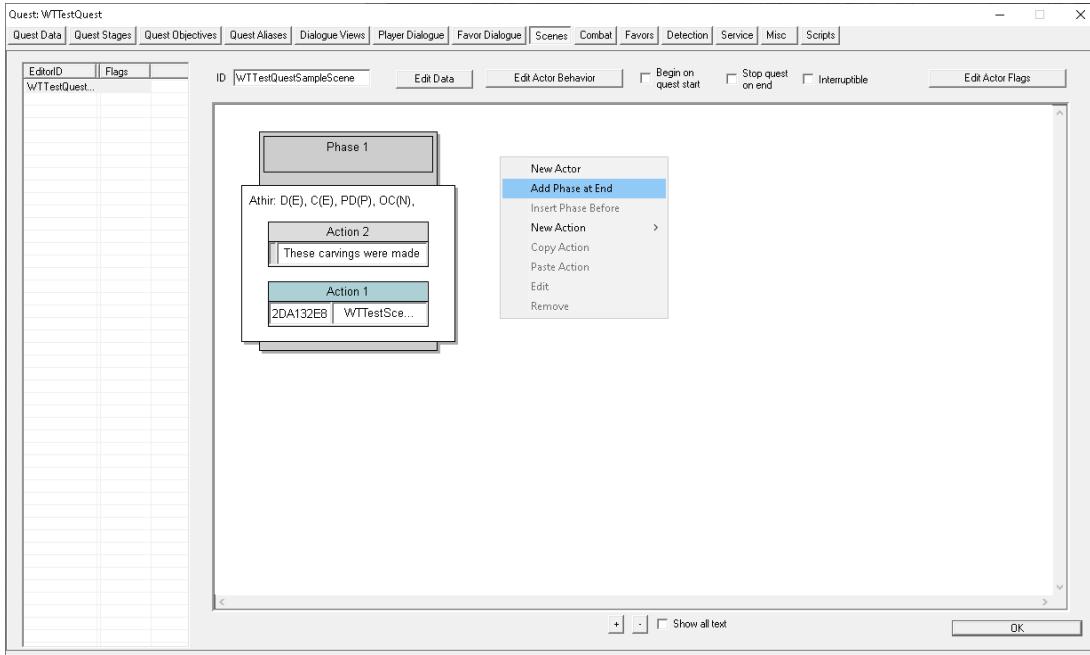


Figure 868 - Adding in a second phase.

You should now see Phase 2 after Phase 1 as per the screenshot below:

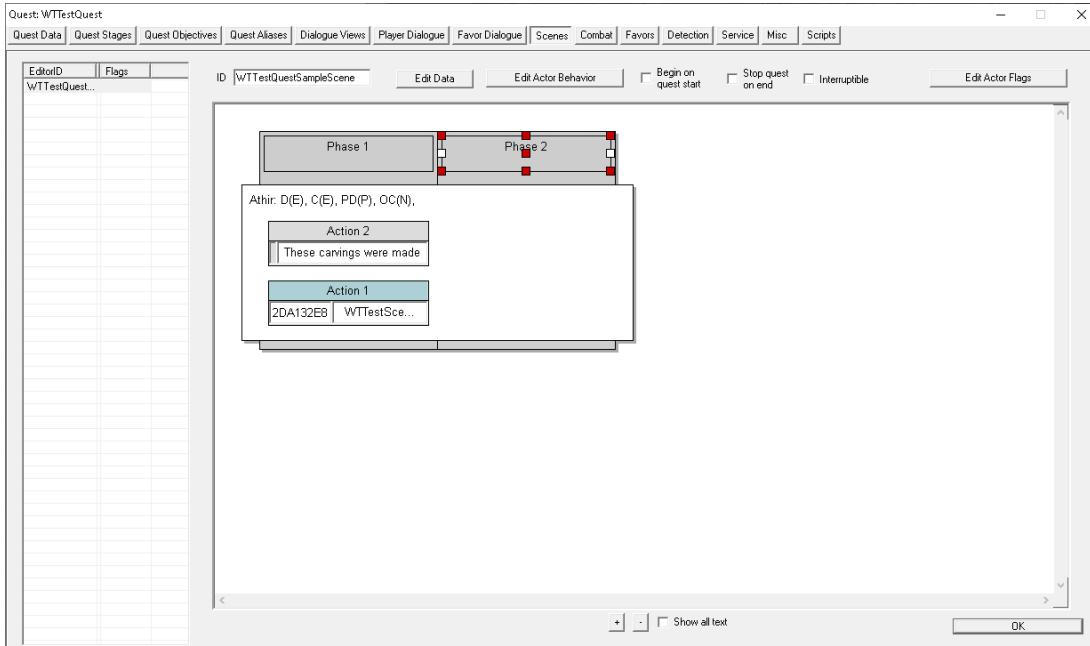


Figure 869 - Second phase added.

Right-click outside of the phases and select New Actor.

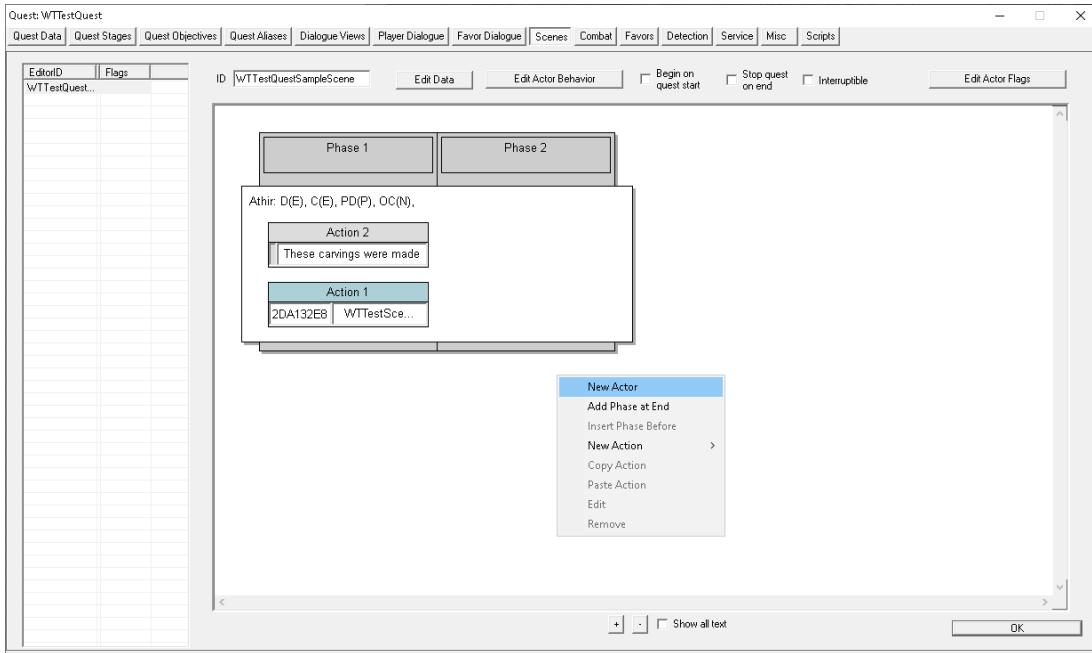


Figure 870 - Adding a second actor into our scene.

Select Daenlit in the Select Form pop-up then click OK.

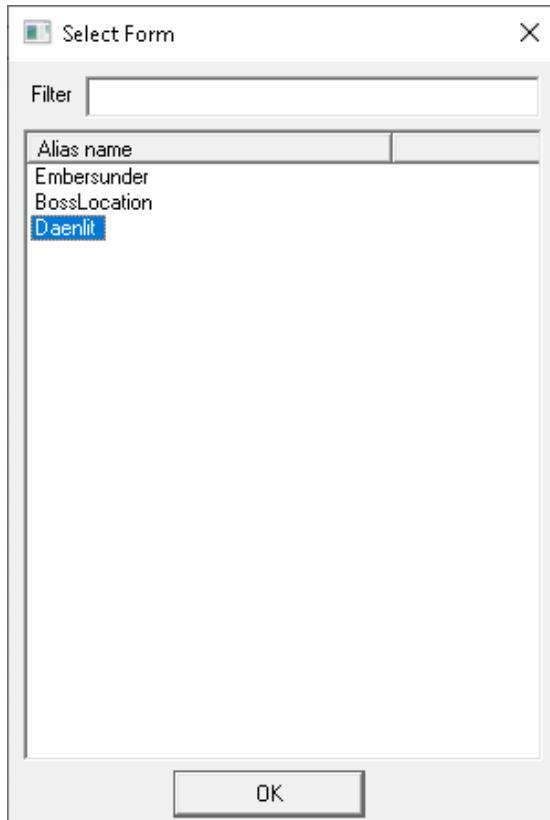


Figure 871 - Selecting Daenlit's alias.

We should now see Daenlit listed below Athir.

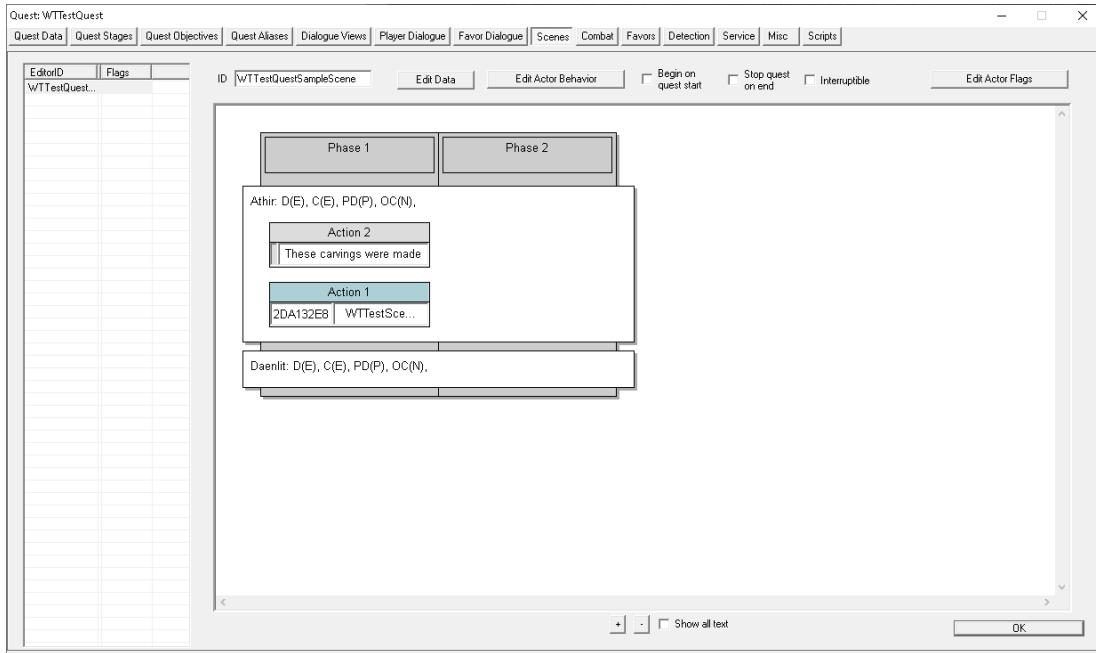


Figure 872 - Scene with two NPCs.

Now let's give her something to say. Right-click under Phase 2 beside Daenlit and select New Action > Dialogue.

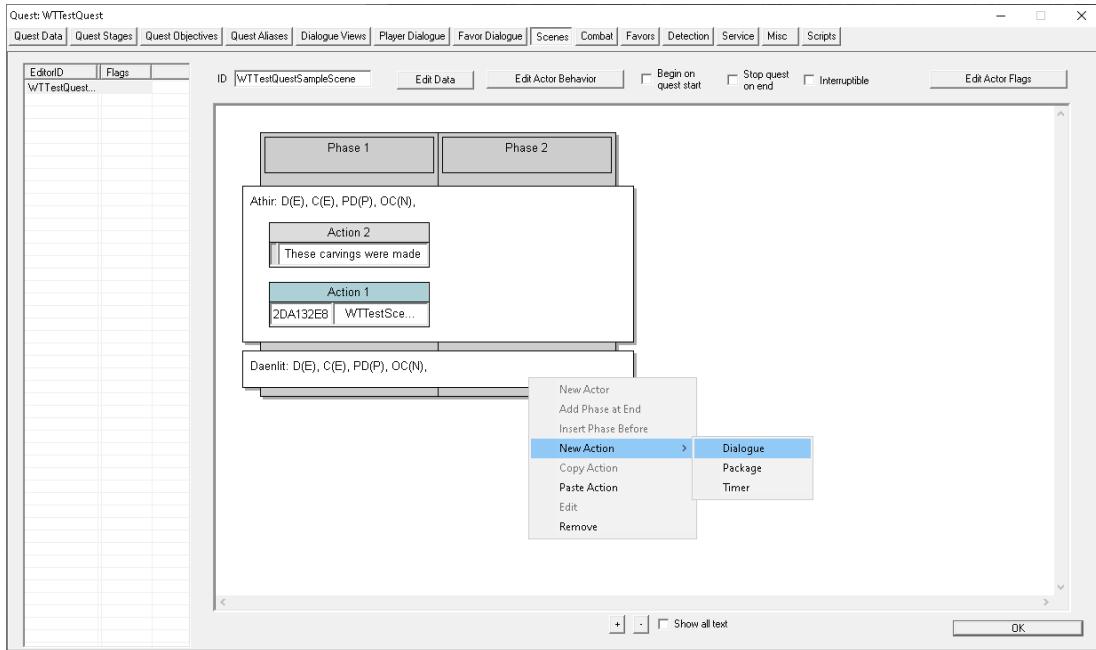


Figure 873 - Assigning scene dialogue to Daenlit.

Right-click in the dialogue list and select New.

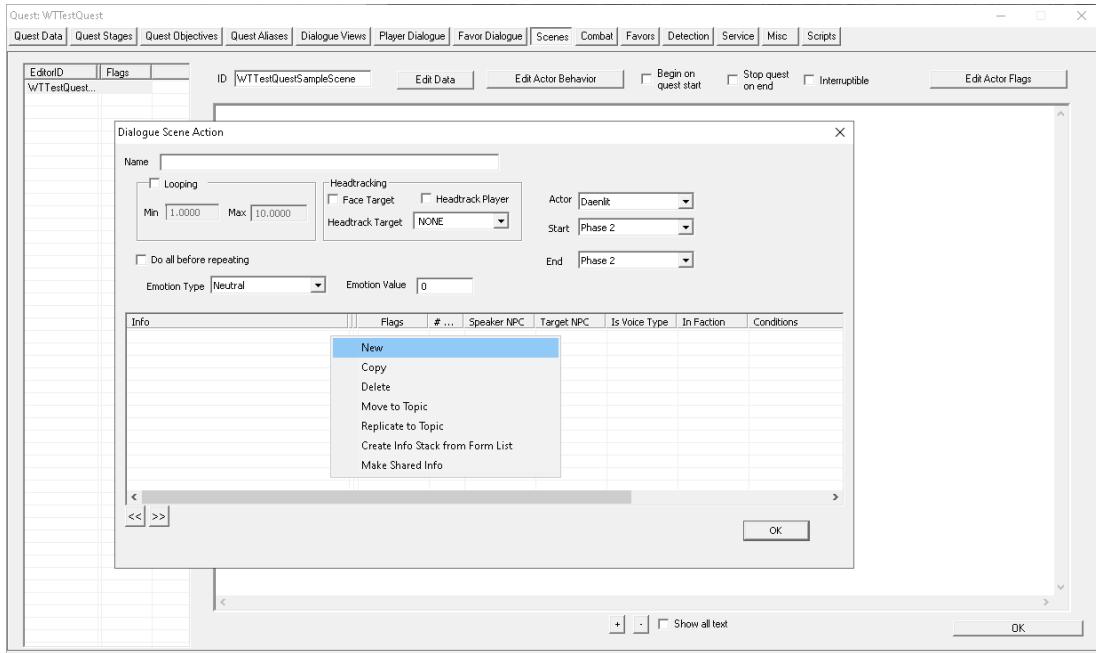


Figure 874 - Adding dialogue to the Dialogue Scene Action.

Enter in her dialogue into the Response Text field. I set her Emotion Type to Fear and the Emotion Value to 20 to match her dialogue.

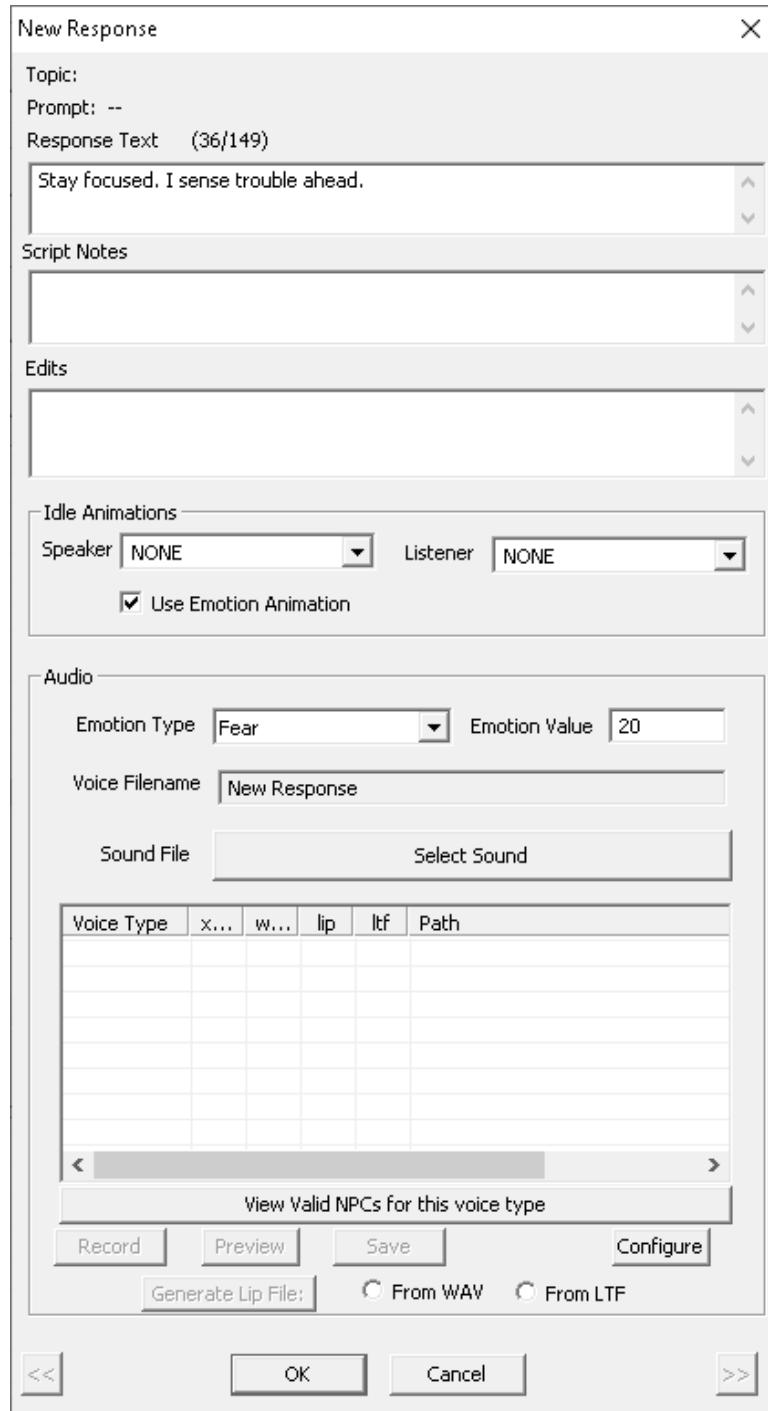


Figure 875 - Adding response text.

Click OK to close out of New Response.

Right-click in the conditions list and select New.

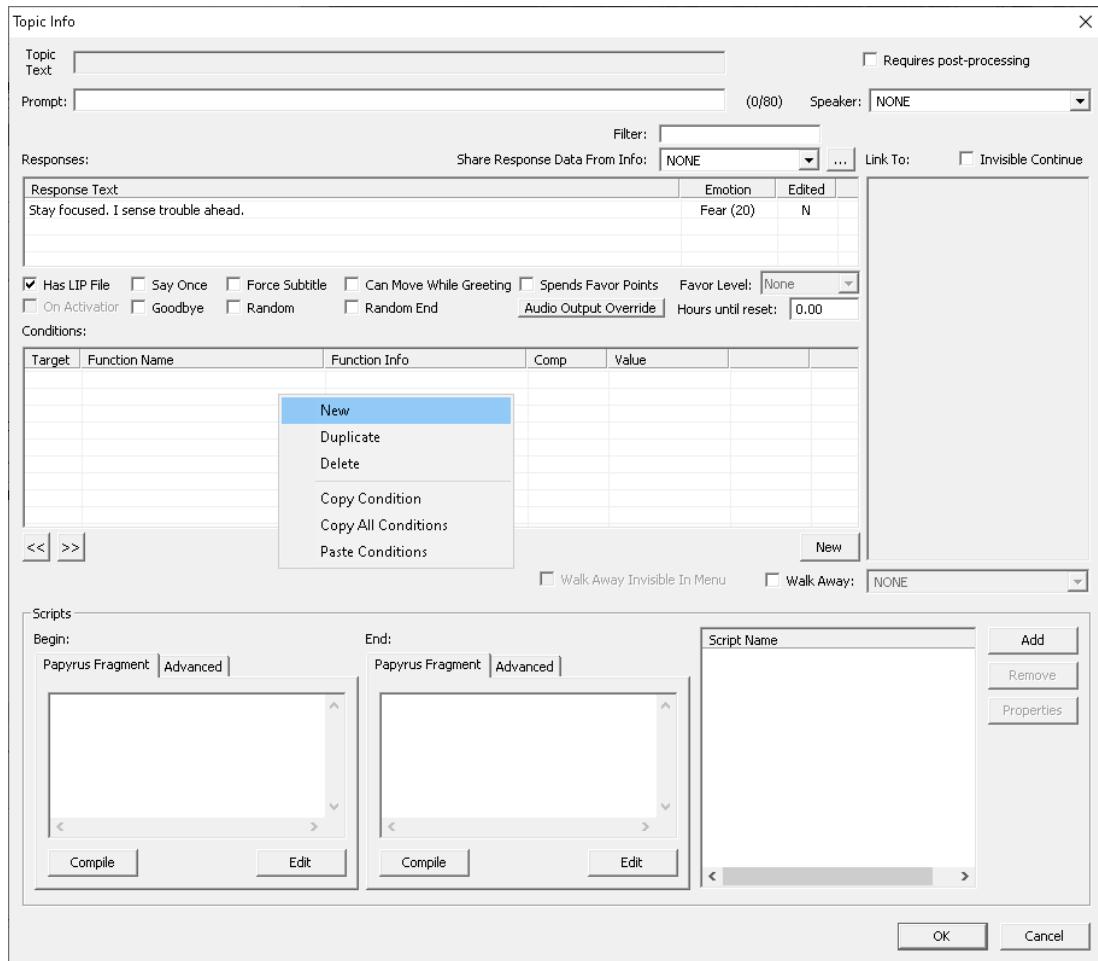


Figure 876 - Adding a condition to Daenlit's response.

Again, I set a GetIsID condition to ensure Daenlit is the only actor able to deliver this line.

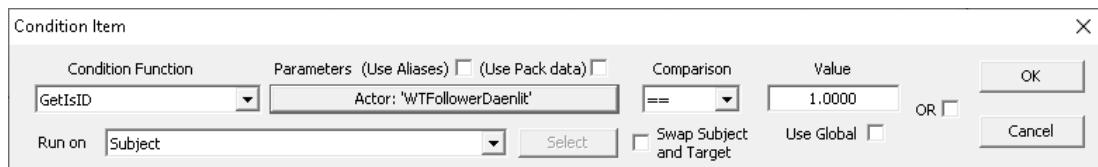


Figure 877 - GetIsID condition.

Click OK to close out of Condition Item properties.

Click OK to close out of Topic Info.

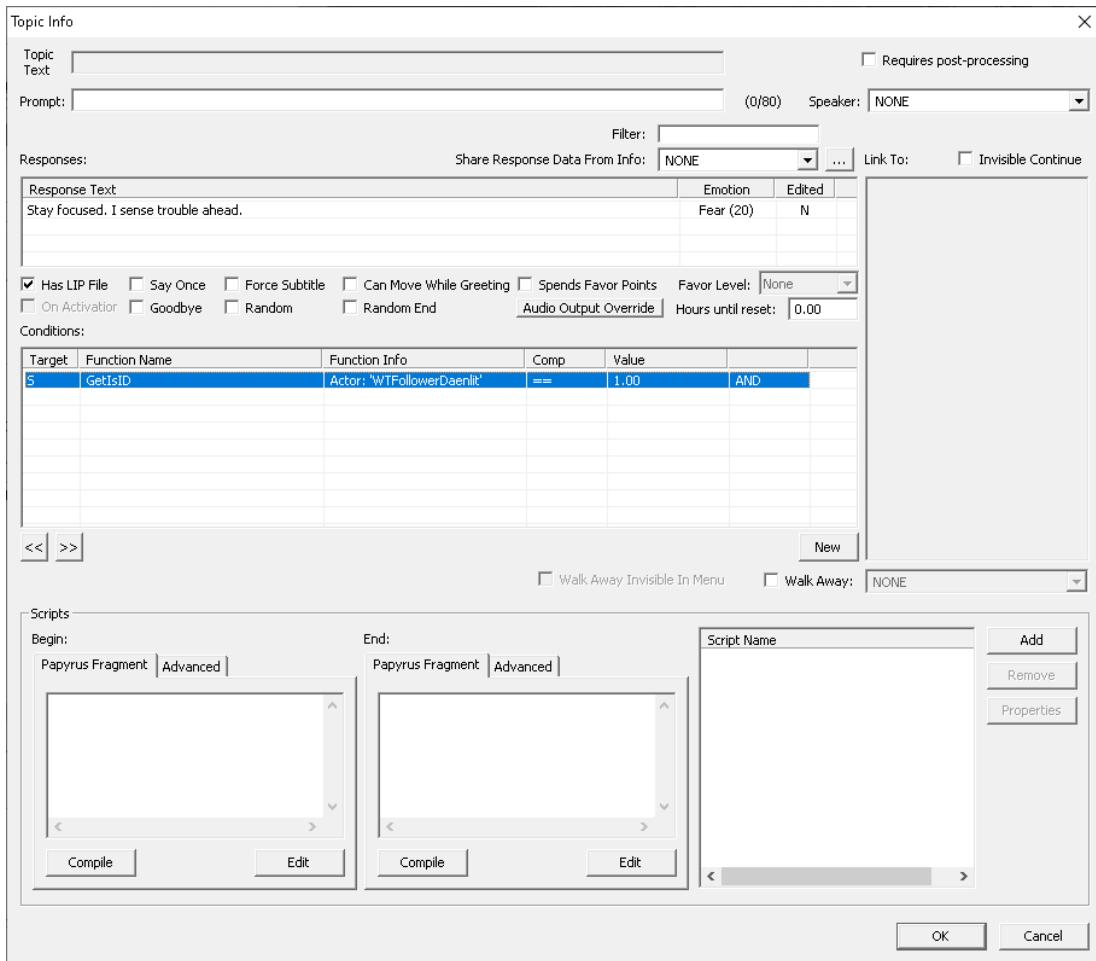


Figure 878 - Topic Info with condition.

Set Headtrack Target to Athir, then click OK to close out of Dialogue Scene Action.

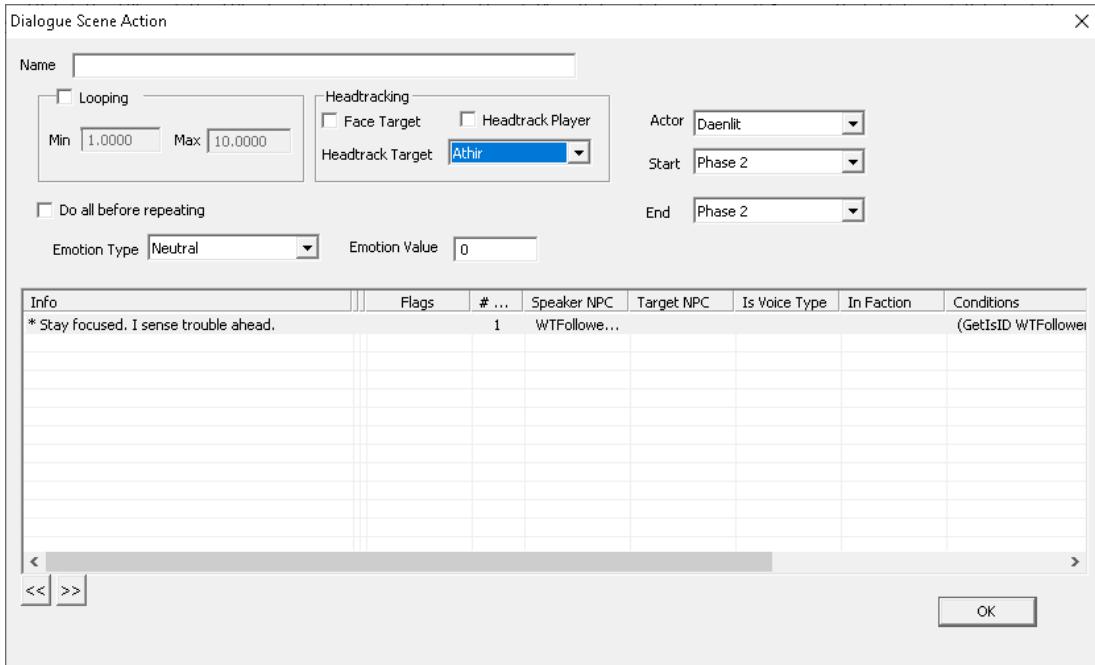


Figure 879 - Dialogue Scene Action with response text.

We should now see our dialogue action in Phase 2. After Athir delivers his line and walks over to the glyphs on the wall, Daenlit should deliver her line. We're not overriding her package so she should continue to follow the player while speaking.

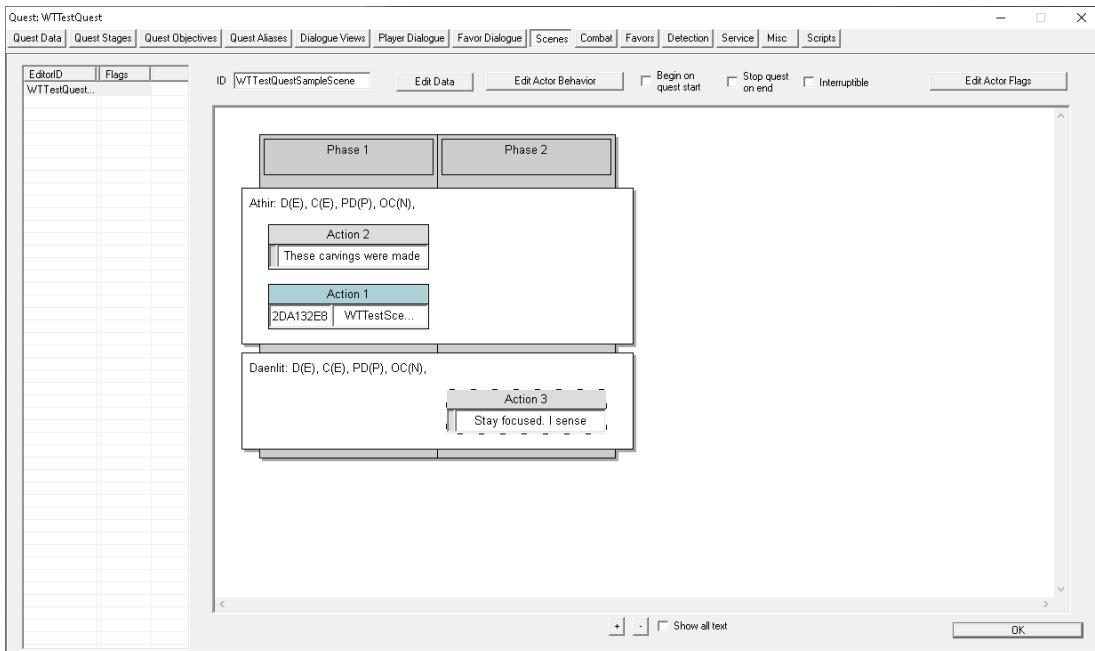


Figure 880 - Phase 2 with a dialogue action.

Let's add in a quick delay before the next action.

Right-click in the empty space after Phase 2 and select 'Add Phase at End'.

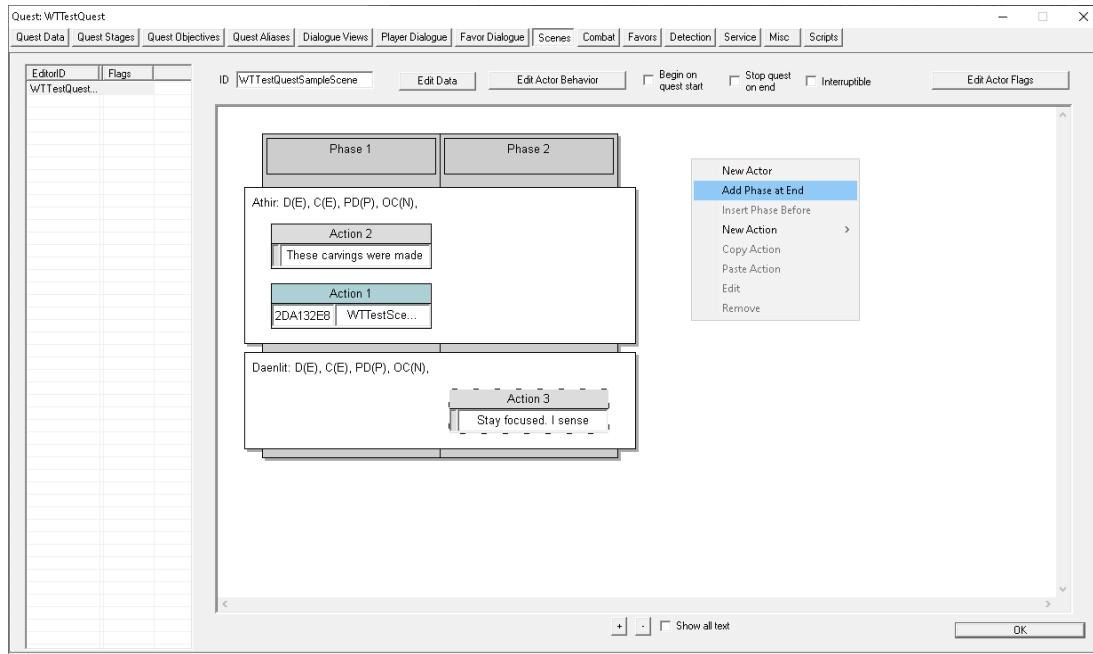


Figure 881 - Adding a third phase.

We should now have three phases as per the following screenshot:

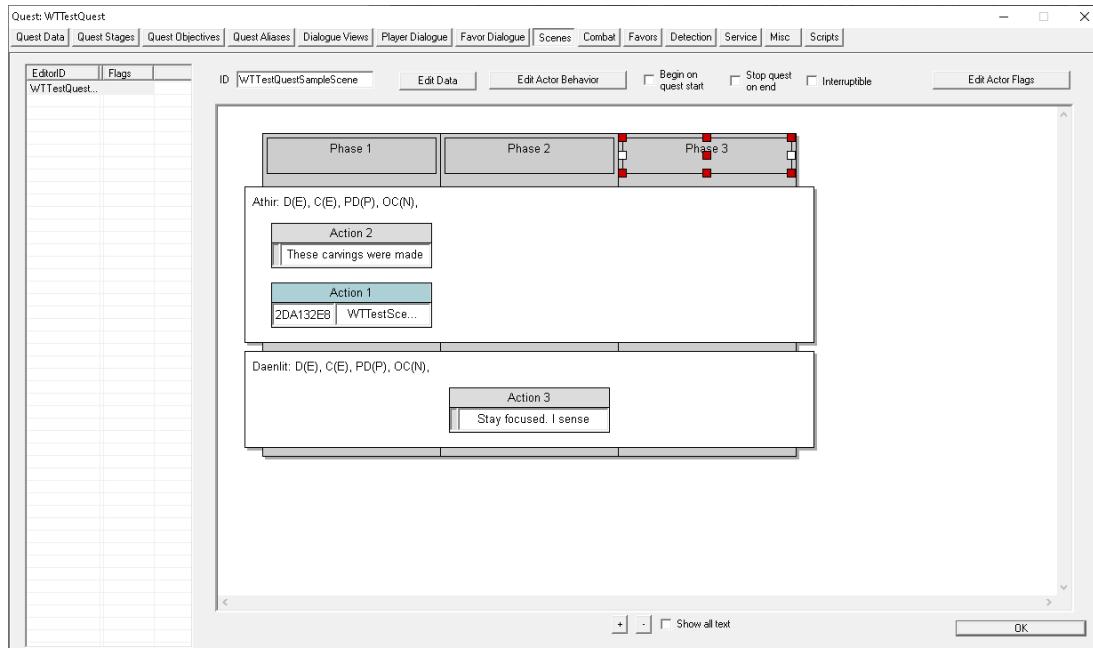


Figure 882 - Our scene with three phases.

Right-click under Phase 3 beside Daenlit and select New Action > Timer.

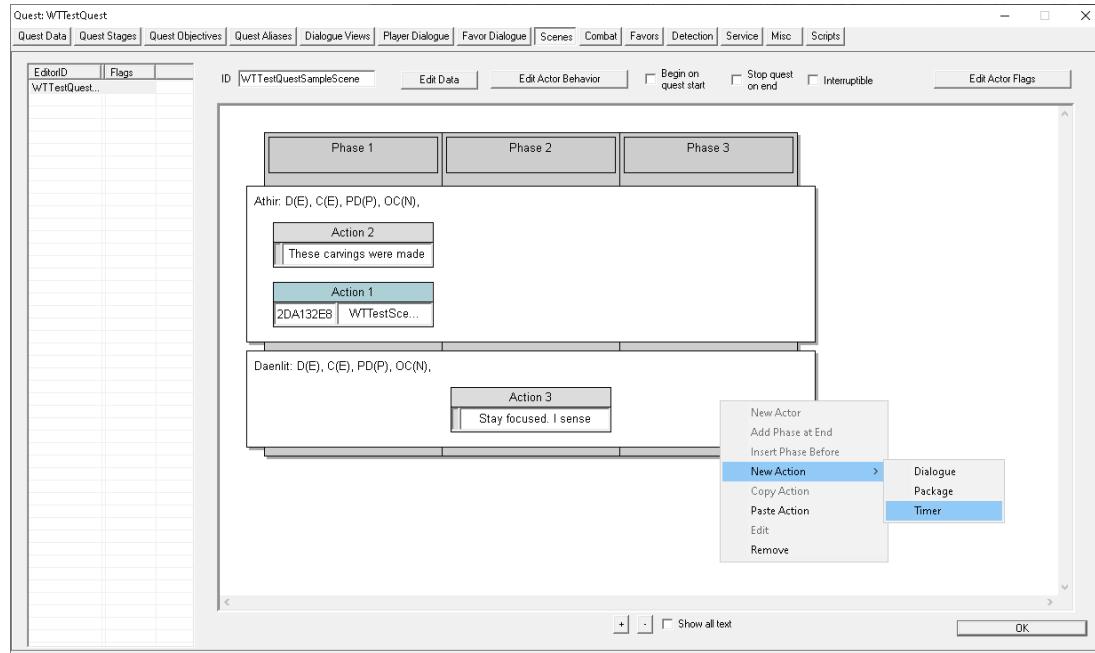


Figure 883 - Adding in a timer.

Set the Seconds field to 3.0000 then click OK.

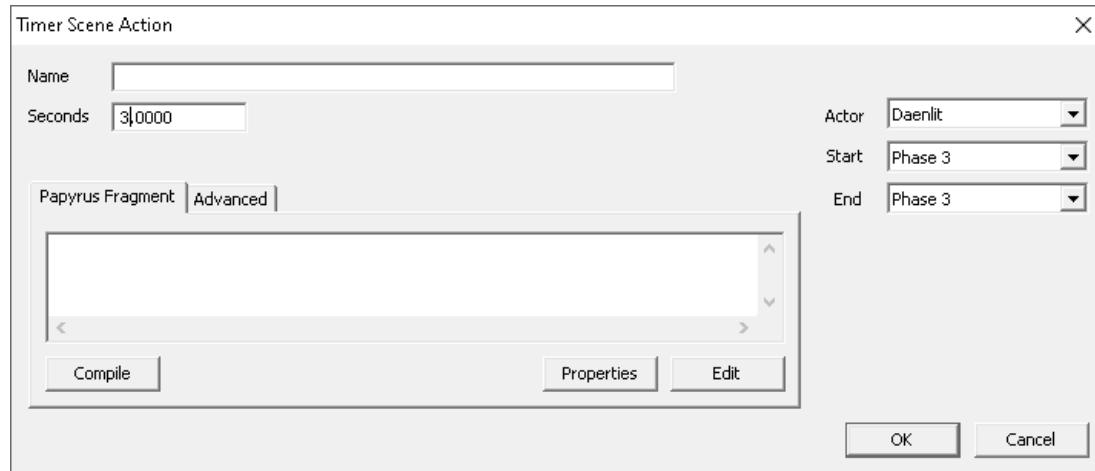


Figure 884 - Adding a three second delay before the next action.

A three second delay should now appear highlighted in yellow under Phase 3.

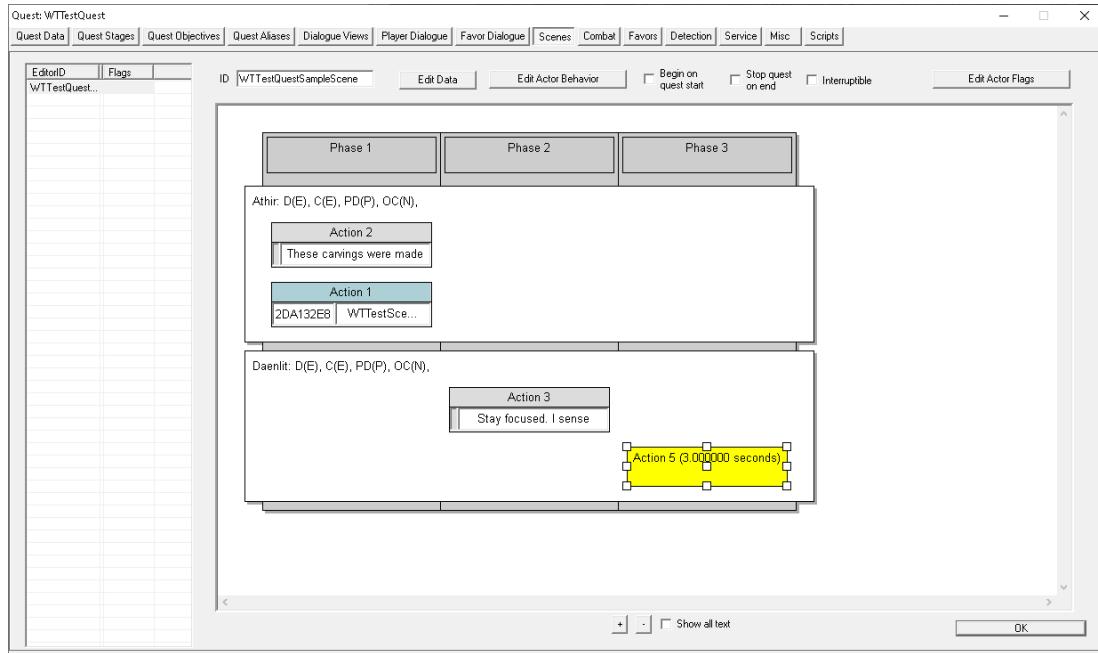


Figure 885 - A delay added to Phase 3.

Now let's add in a third NPC.

Go back to the Quest Aliases tab, right-click in the aliases list and select New Reference Alias.

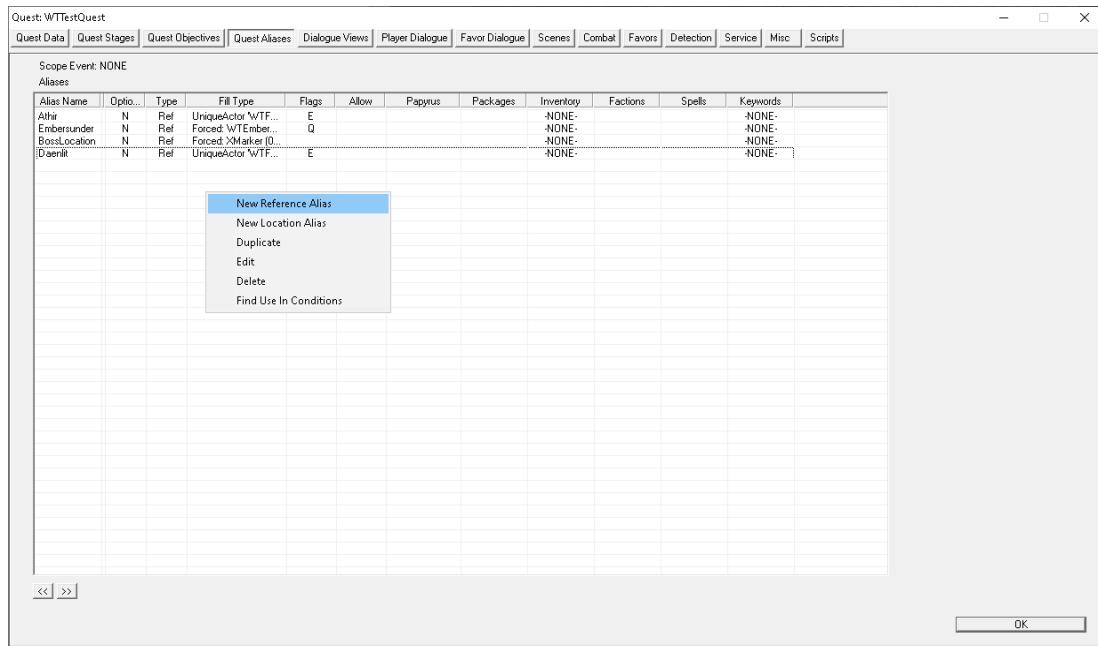


Figure 886 - Adding in a third NPC alias.

For the third NPC alias, I select Shargam and flagged him as essential.

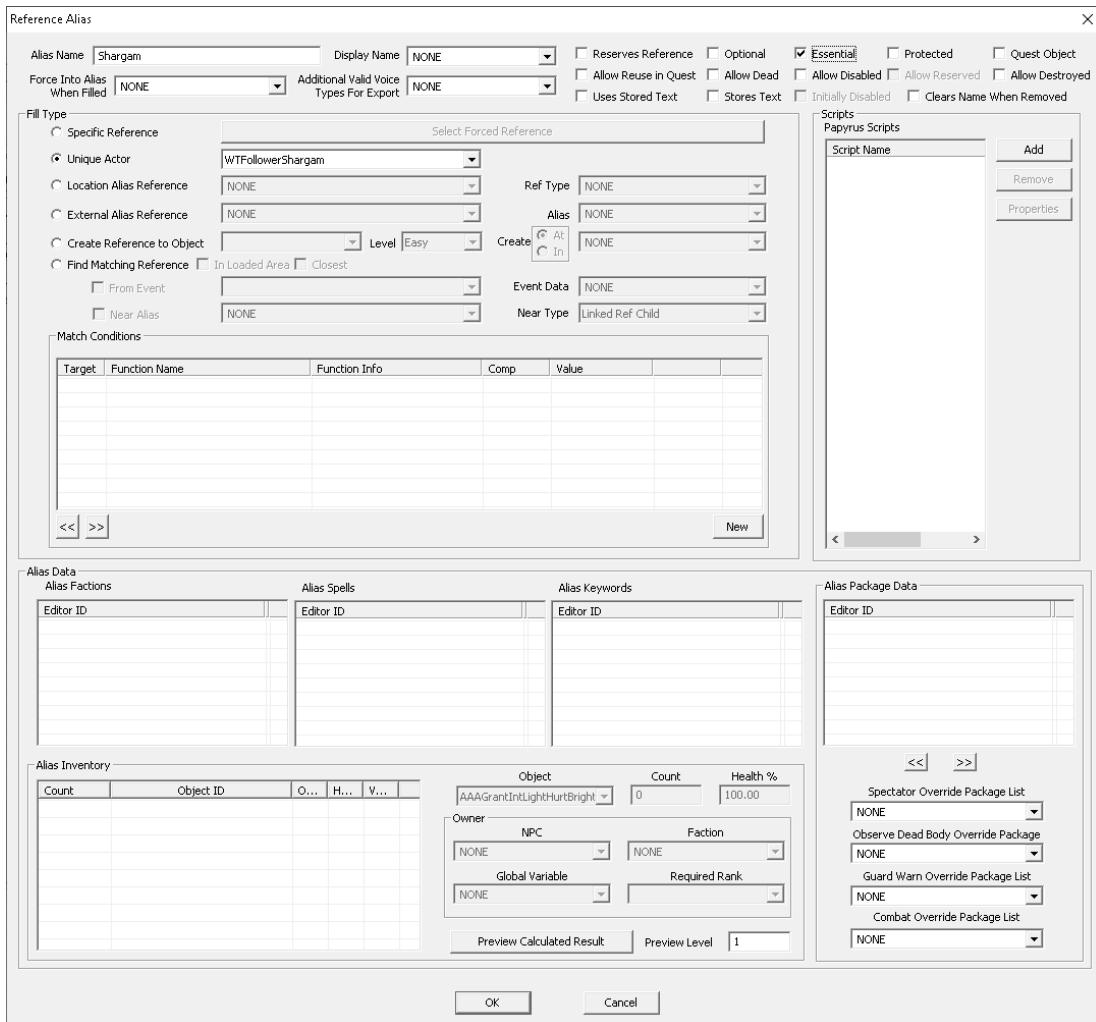


Figure 887 - Setting up our third NPC reference alias.

Click OK to close out of Reference Alias properties.

Go back to our scene in the Scenes tab.

Right-click in the space after Phase 3 and select ‘Add Phase at End’.

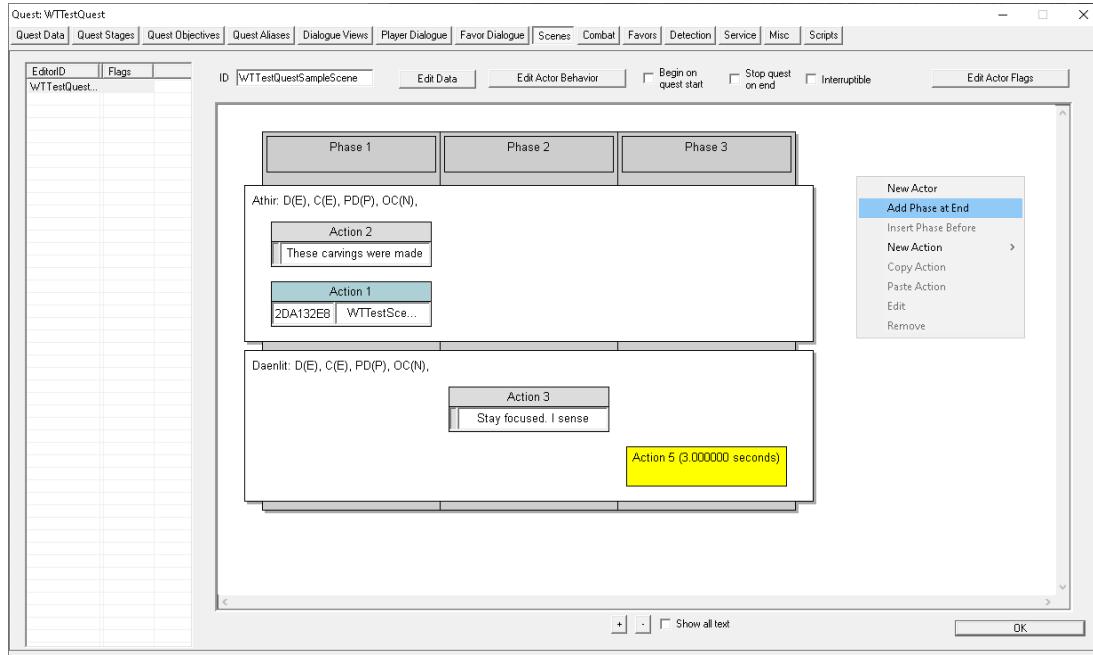


Figure 888 - Adding a fourth phase.

We should now see four phases in our scene as per the screenshot below:

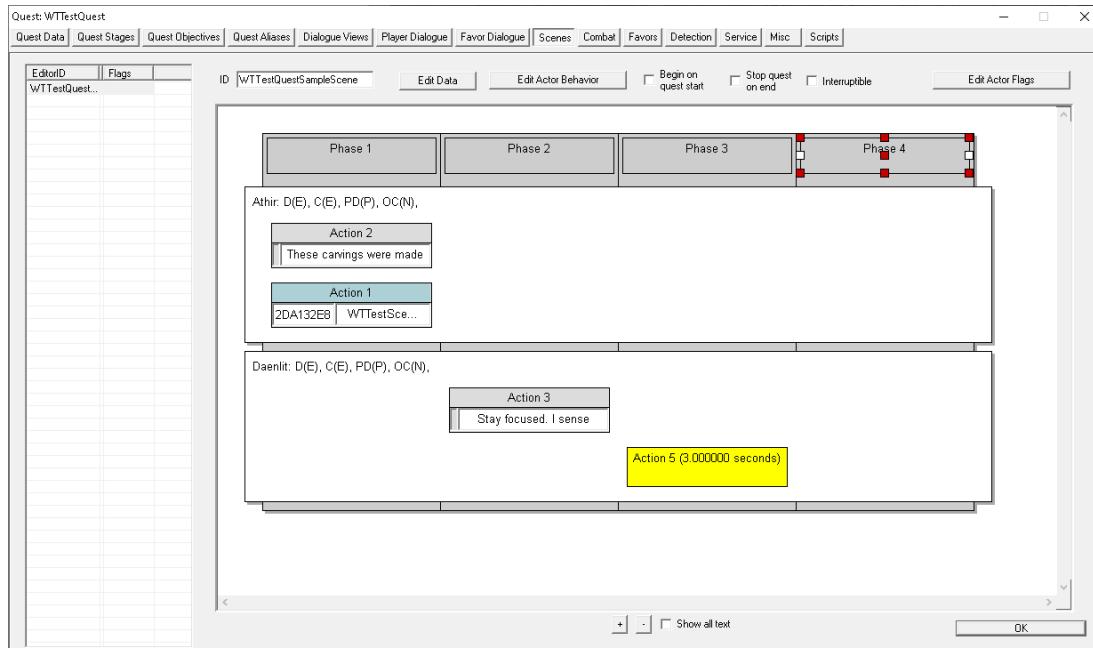


Figure 889 - Our scene with four phases.

Right-click in the area outside of the phases and select New Actor.

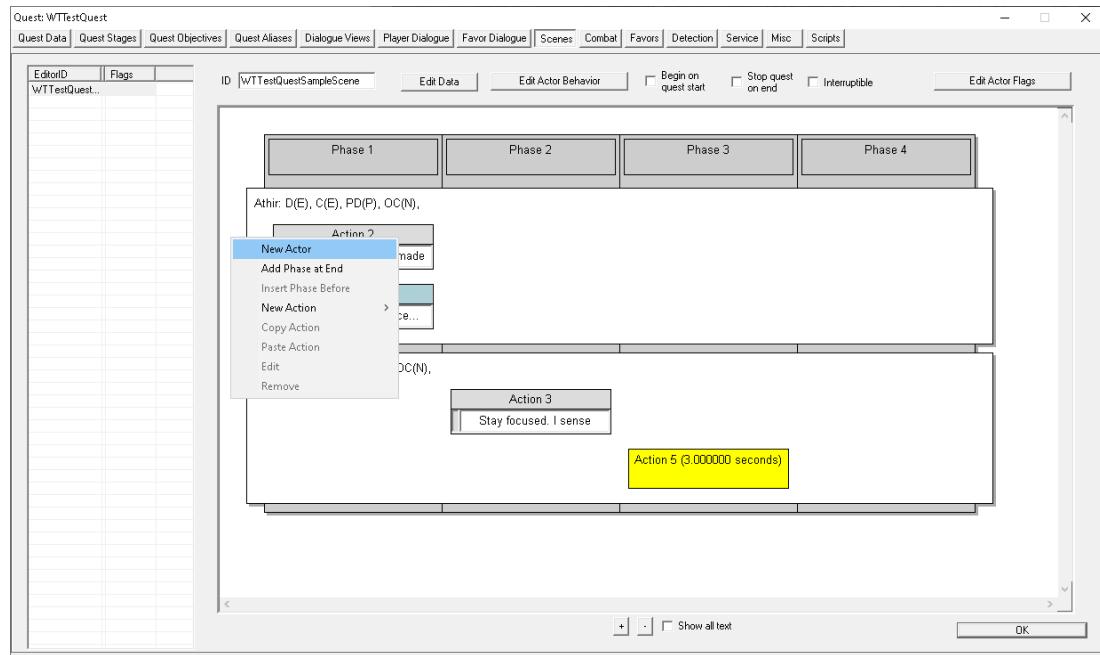


Figure 890 - Adding in our third NPC.

Select Shargam in the alias list then click OK.

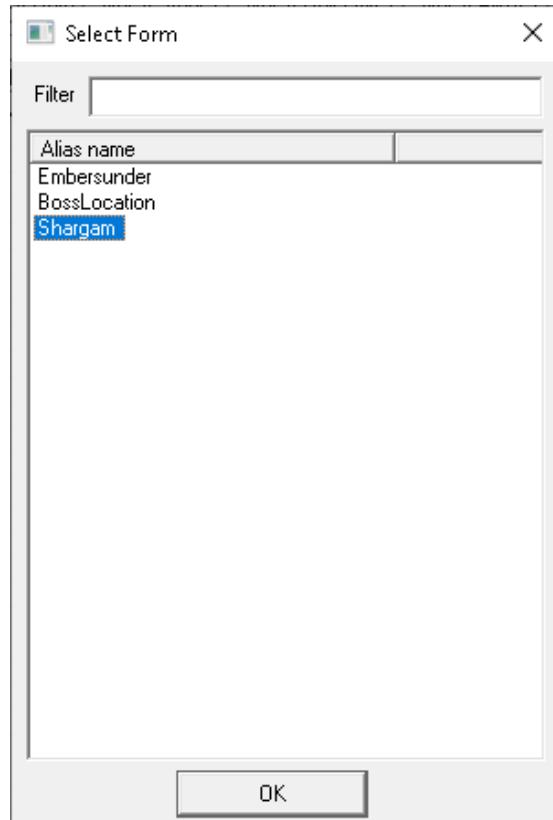


Figure 891 - Selecting Shargam's alias.

Let's set up a travel package for Shargam first. Right-click under Phase 4 next to Shargam and select New Action > Package.

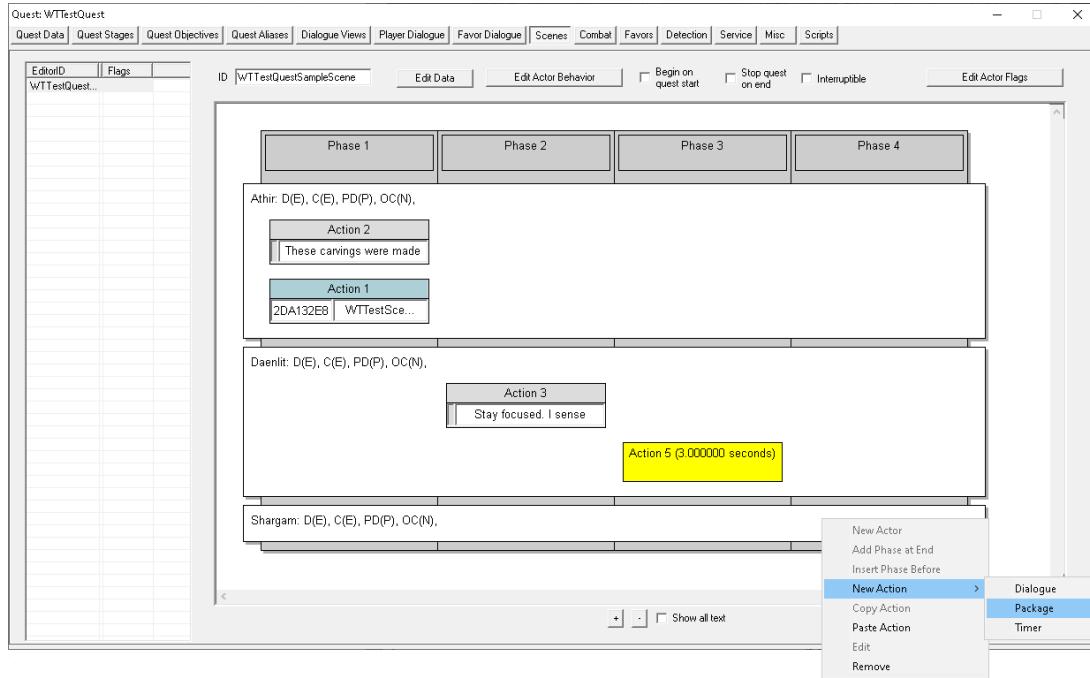


Figure 892 - Creating a new scene package for Shargam.

Right-click in the packages list and select New.

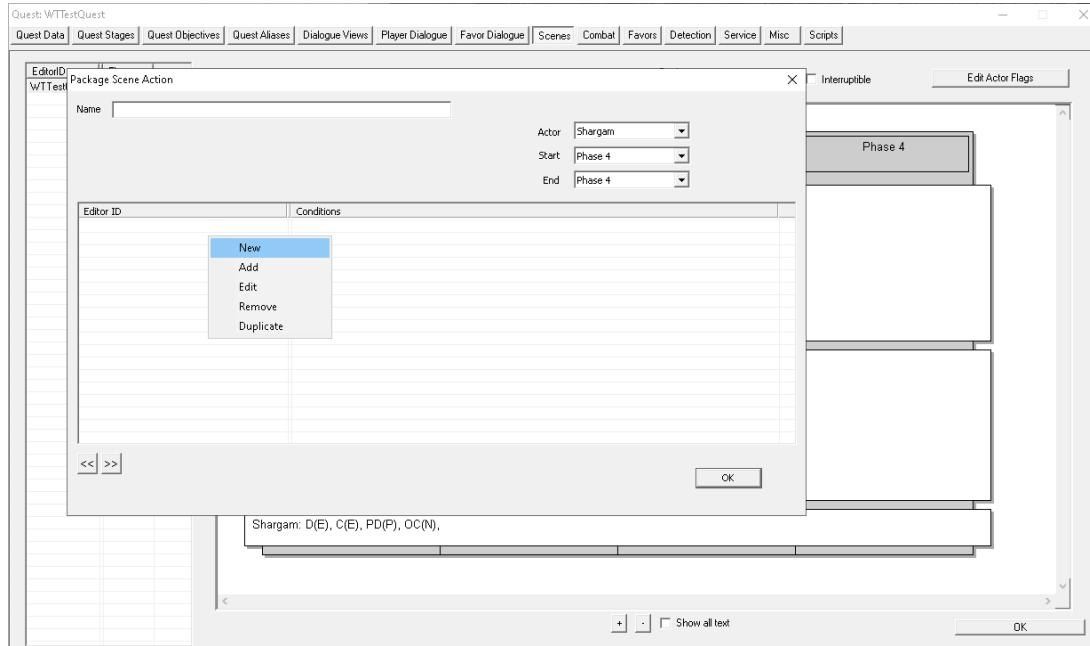


Figure 893 - Adding a new package to Package Scene Action.

Enter in the package ID. I just called it WTTTestSceneShargamWalkToPackage for this example.

Make sure the ‘Owner quest’ drop-down is set to the name of the quest you’re setting up the scene in.

In this example, I set up a new package using the Travel template and made Shargam travel to the XMarkerHeading facing the puzzle door.

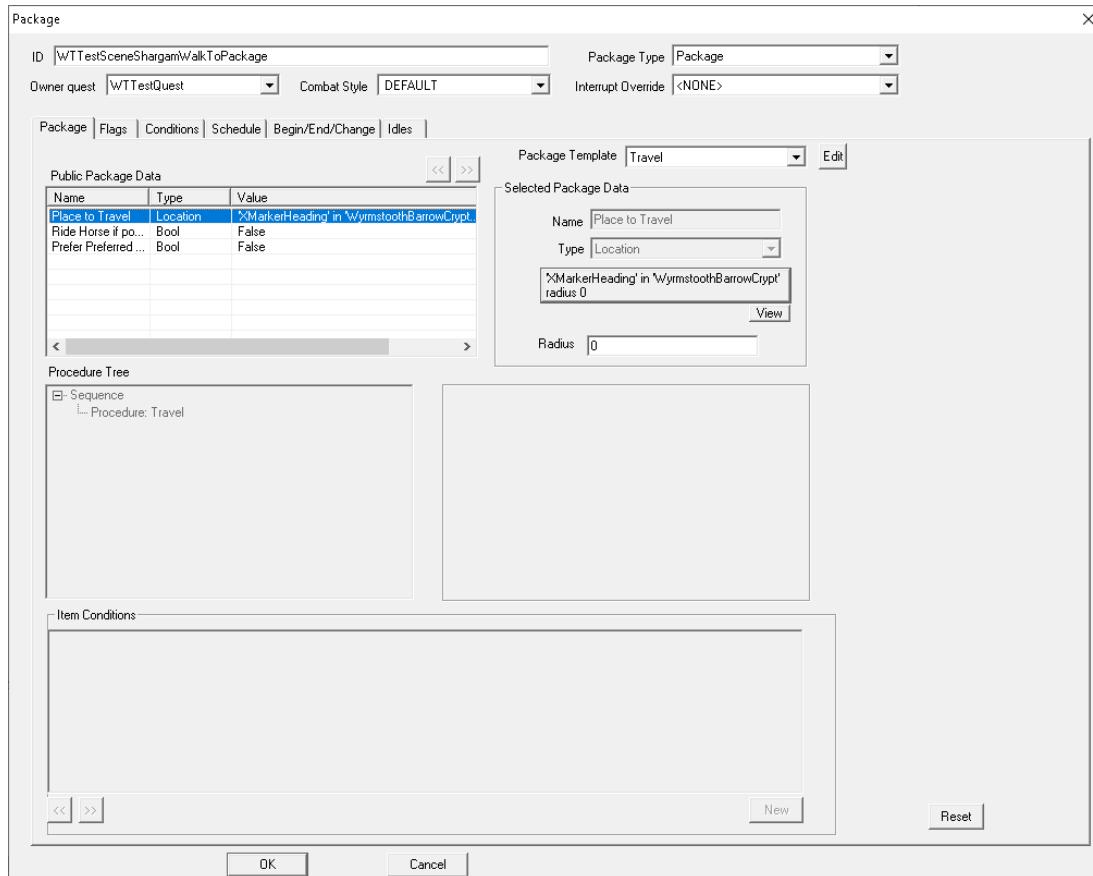


Figure 894 - Shargam's package for phase 4.

Go to the Flags tab.

Tick Preferred Speed and set the drop-down beside it to Run.

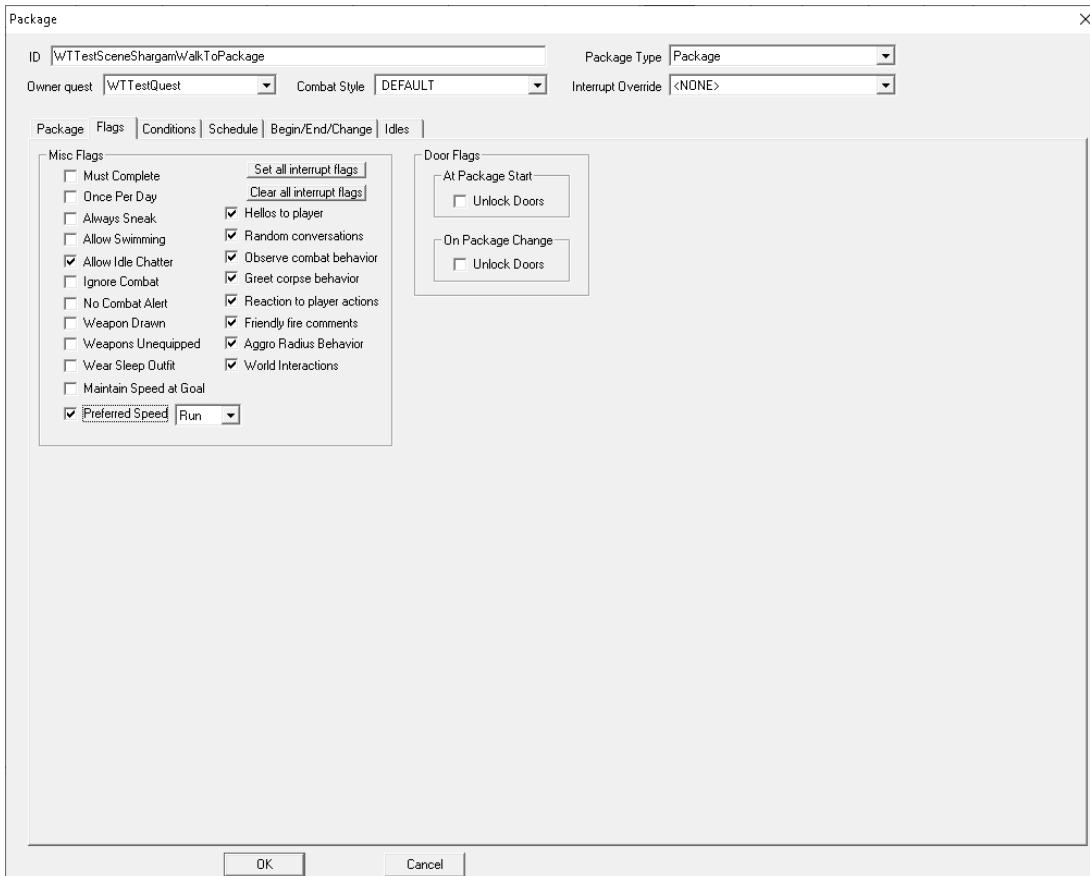


Figure 895 - Setting Shargam's movement speed for this package.

Click OK to add the new package to Shargam for phase 4 of the scene.

Click OK to close out of Package Scene Action.

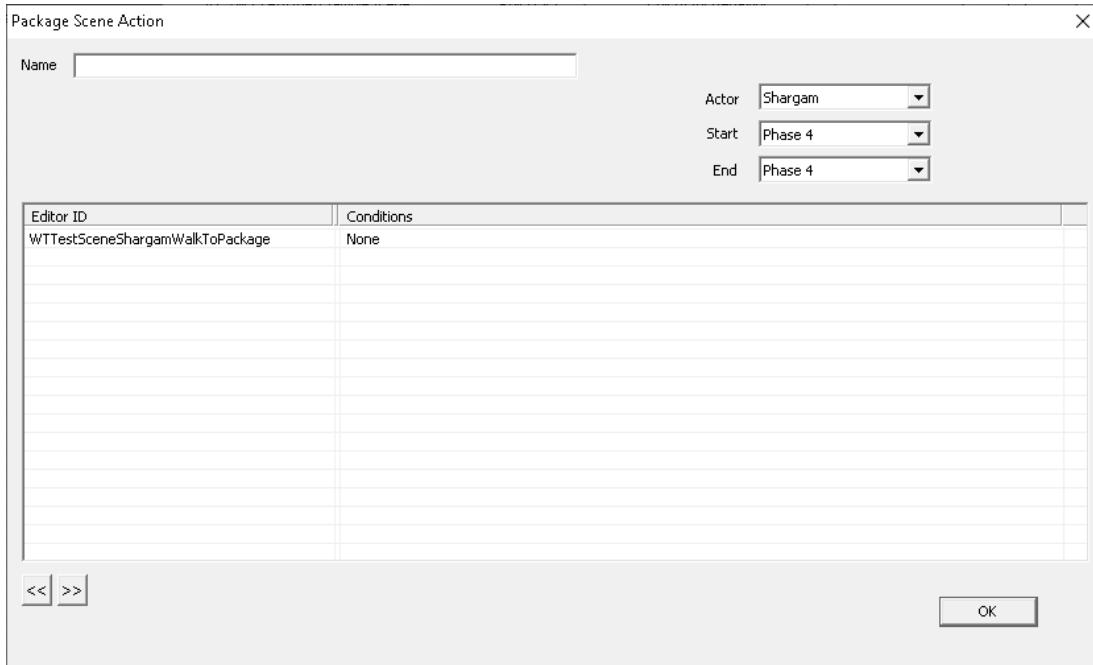


Figure 896 - Package Scene Action with a package set up for phase 4.

We should now see Shargam's phase 4 package.

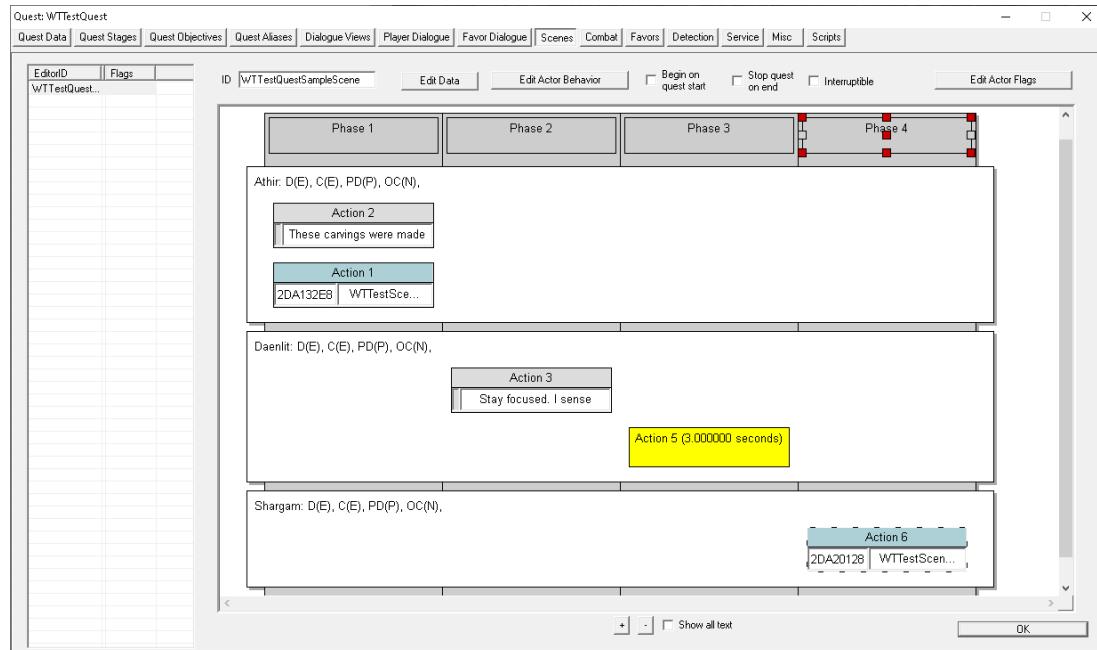


Figure 897 - Phase 4 with a package set up for Shargam.

Lastly, let's give him something to say.

Right-click under Phase 4 next to Shargam and select New Action > Dialogue.

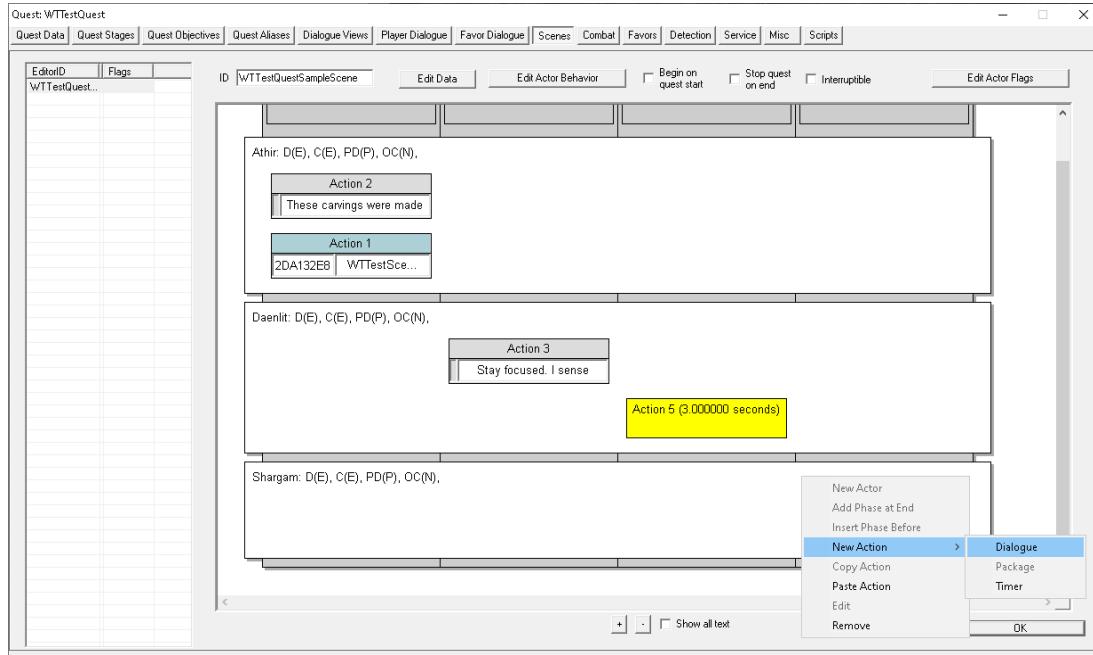


Figure 898 - Adding a dialogue action for Shargam in Phase 4.

Right-click in the responses list and select New.

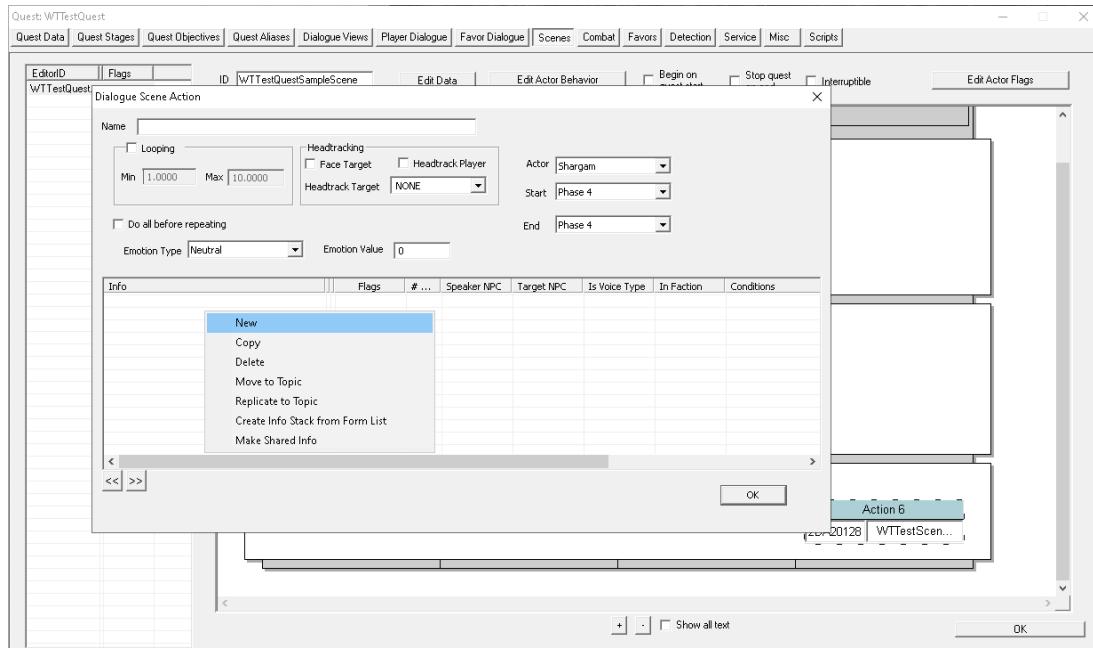


Figure 899 - Adding a line of dialogue to the Dialogue Scene Action.

Enter Shargam's line in the Response Text field. I also set his Emotion Type to Anger and the Emotion Value to 40.

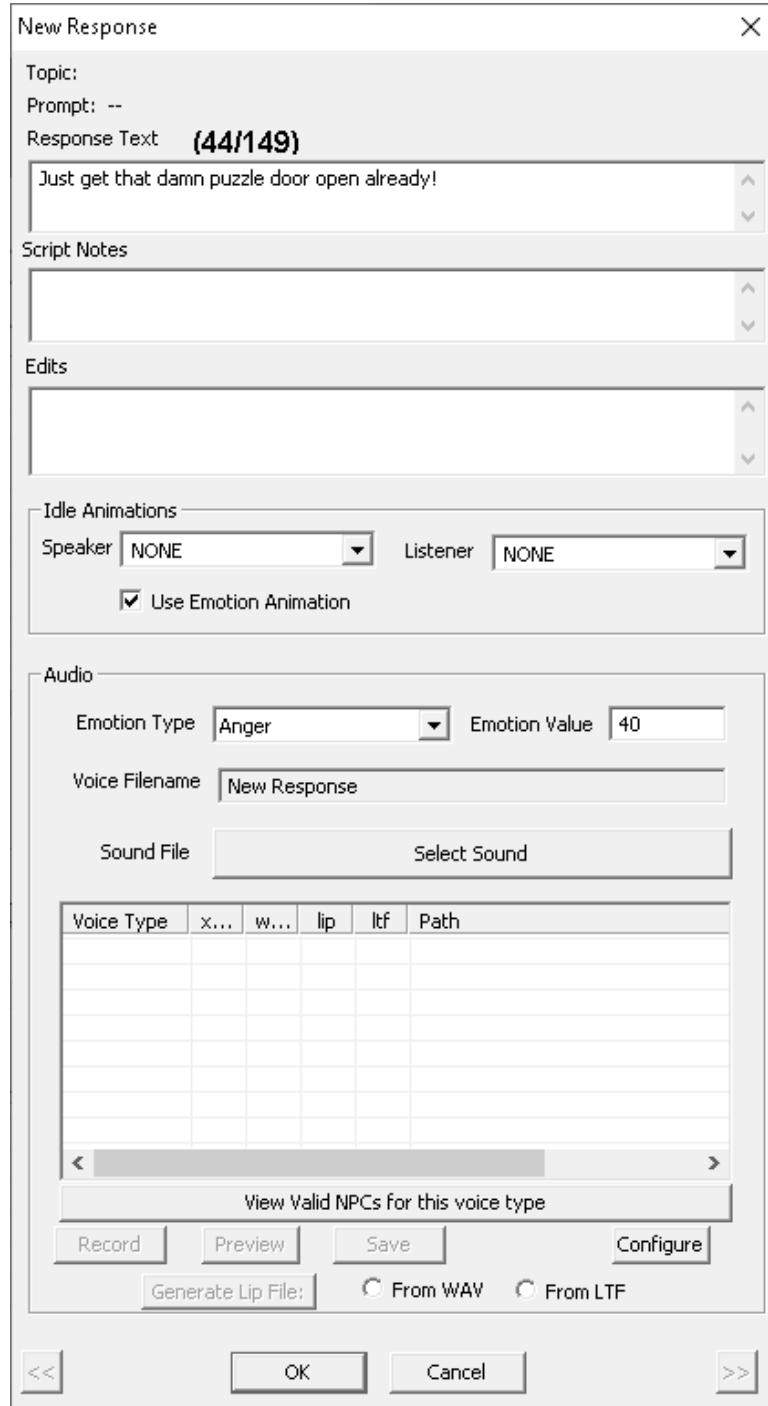


Figure 900 - Shargam's response for phase 4.

Click OK to close out of New Response.

Right-click in the conditions list and select New.

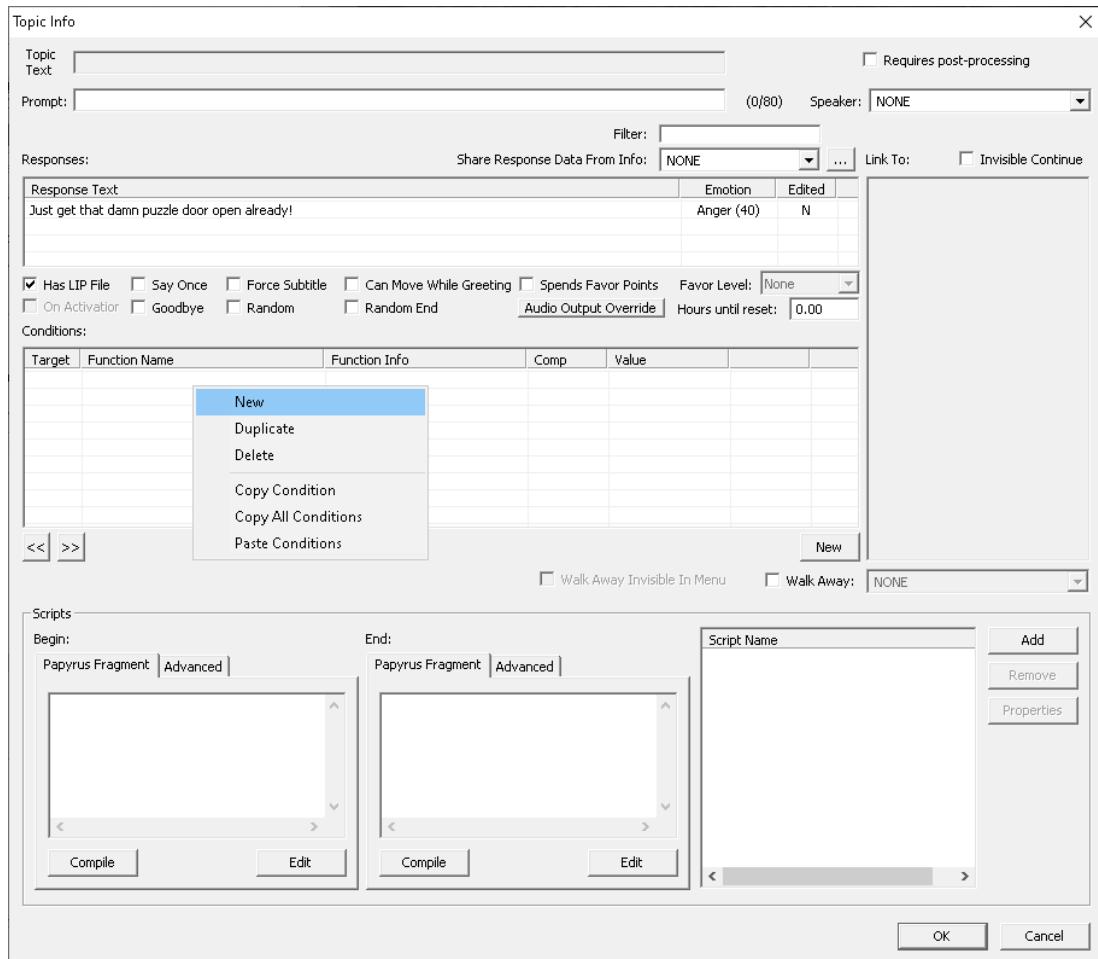


Figure 901 - Adding a new condition to Shargam's line of dialogue.

Again I set the Condition Function to GetIsID and set the actor to Shargam to ensure only Shargam can say this line of dialogue.

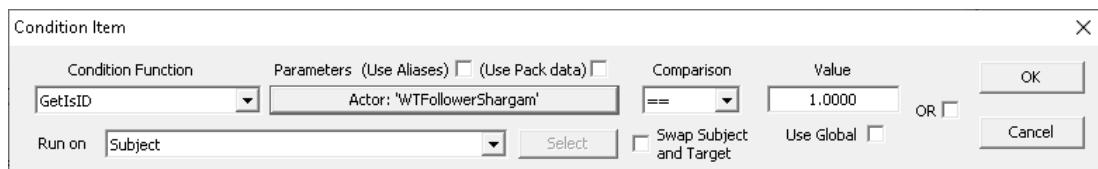


Figure 902 - GetIsID condition set to Shargam.

Click OK to close out of Condition Item.

Click OK to close out of Topic Info.

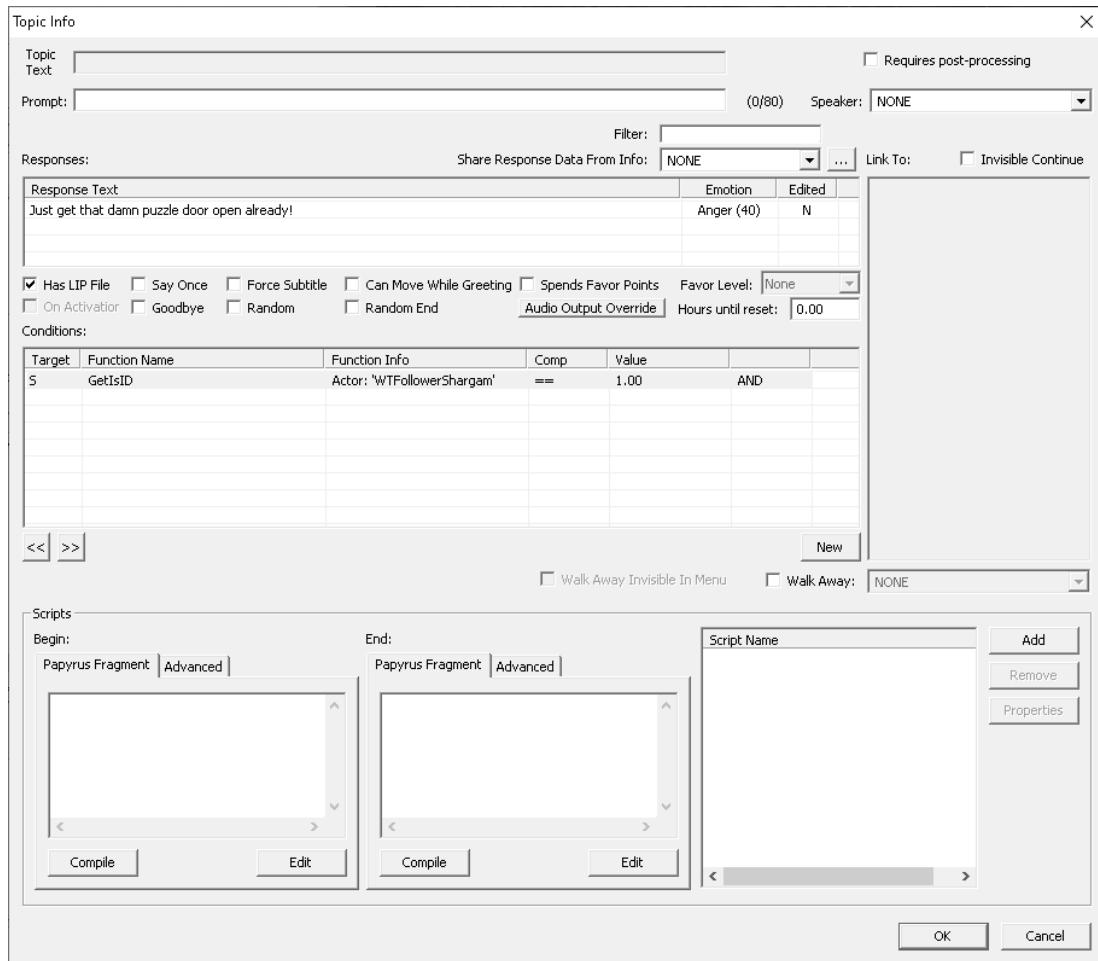


Figure 903 - Topic Info with its condition set up.

Tick Headtrack Player so Shargam looks at the player while saying this line, then click OK to close out of Dialogue Scene Action.

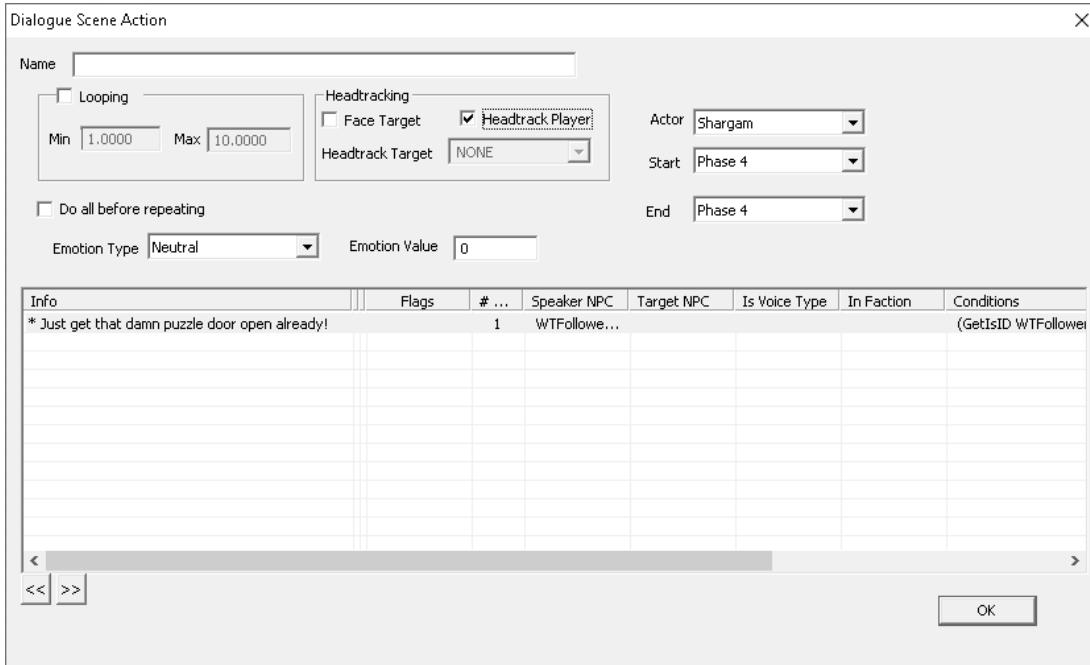


Figure 904 - Dialogue Scene Action with a line of dialogue.

So now after Daenlit speaks, there'll be a three second pause before Shargam runs over to the puzzle door and demands that the player hurries up and opens it.

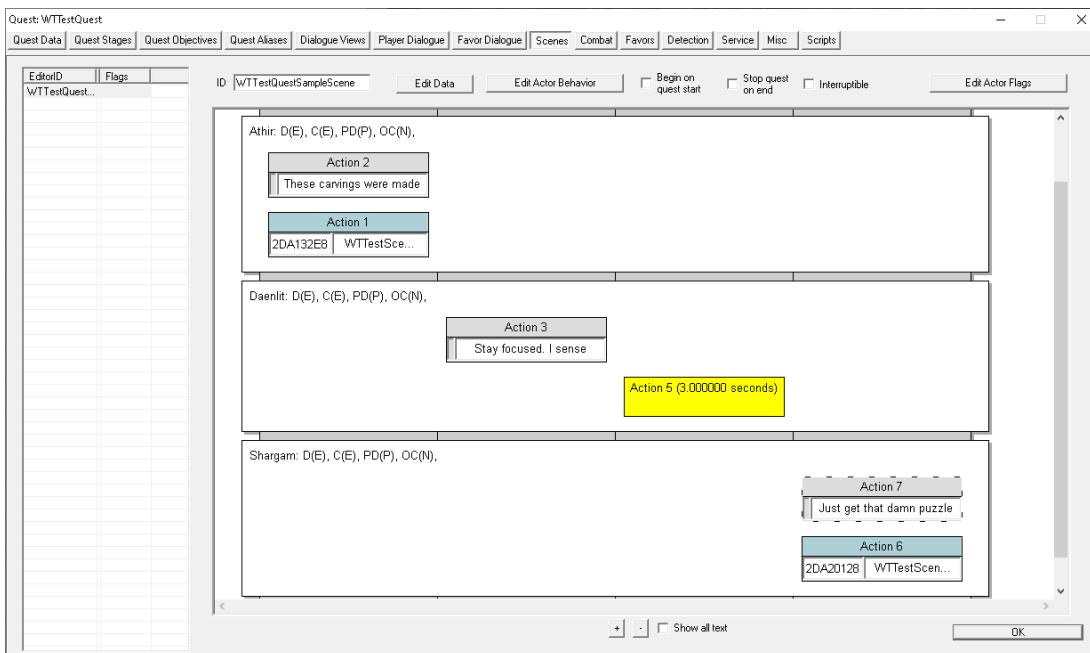


Figure 905 - Four phases set up.

To set up voice acting for this scene, follow the steps in the section [Adding Voice Acting and Lip Syncing](#).

We can configure interrupt behaviour for this scene by clicking on the Edit Actor Behaviour button.

By default, the scene will end automatically if any NPCs involved die or engage in combat. You can change this here if you wish but for this example, I'm just going to leave it at its defaults.



Figure 906 - Configuring interrupt behaviour.

The final thing we need to do is add in a trigger to begin this scene. In the render window, I highlighted one of the Hall Of Stories tileset pieces to set where I want a new trigger to be placed when I click on the Create Trigger button.

To add in a new trigger, click on the Create Trigger button in the toolbar at the top.

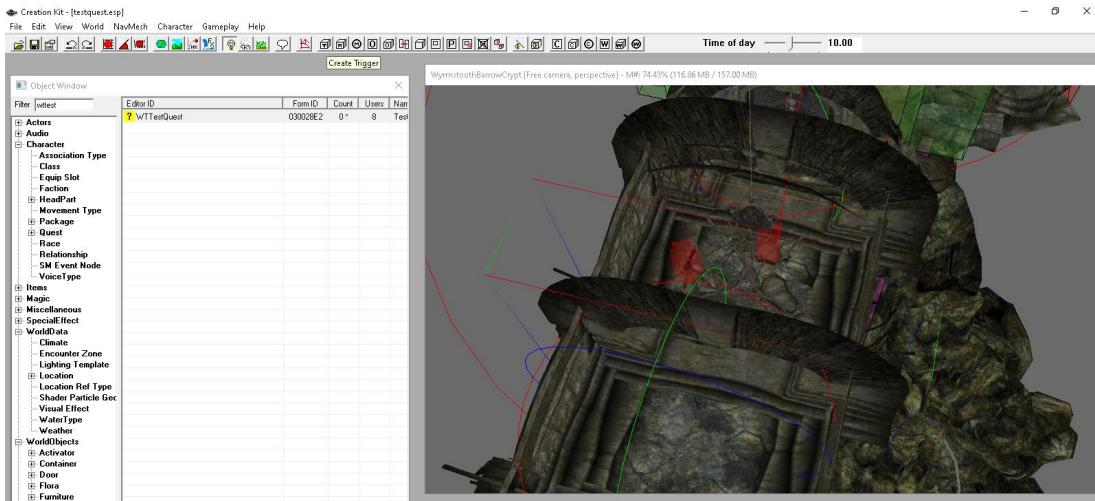


Figure 907 - Create trigger button.

Filter by 'defaultstartscene', select defaultStartSceneTrigger then click OK.



Figure 908 - Finding the defaultStartSceneTrigger script.

Our defaultStartSceneTrigger box should appear in the render window. You can resize it as necessary. Just ensure that the player can't bypass it by walking around it.

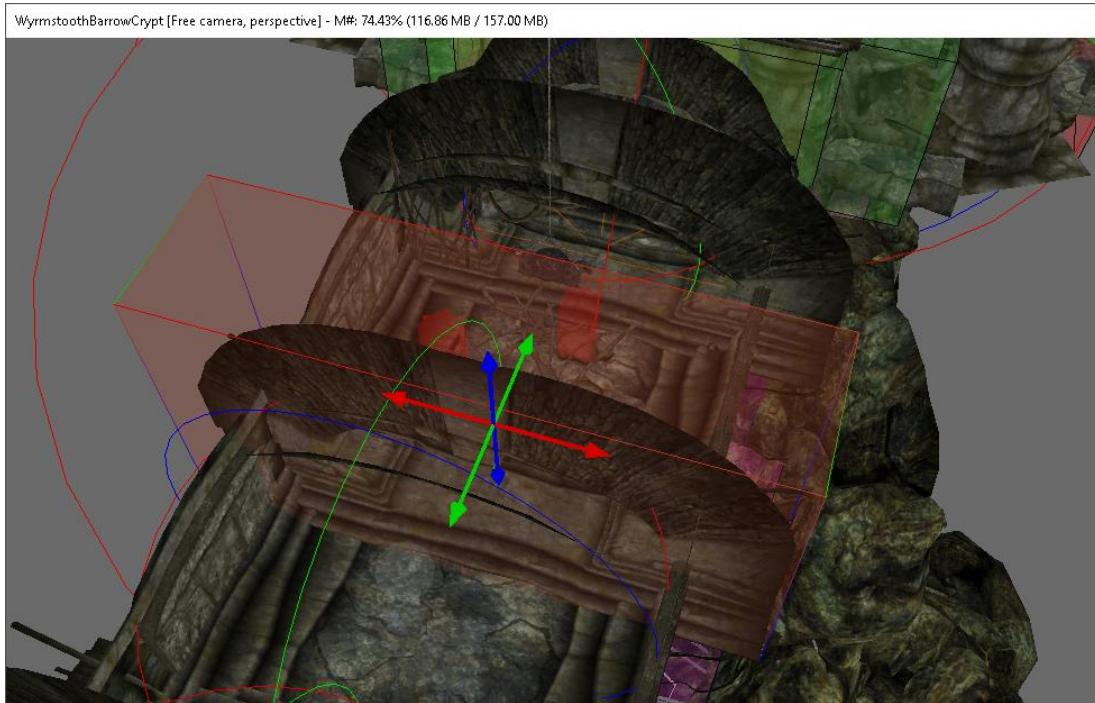


Figure 909 - *defaultStartSceneTrigger* box.

Double-click or right-click on the trigger box in the render window and select Edit.

Go to the Scripts tab.

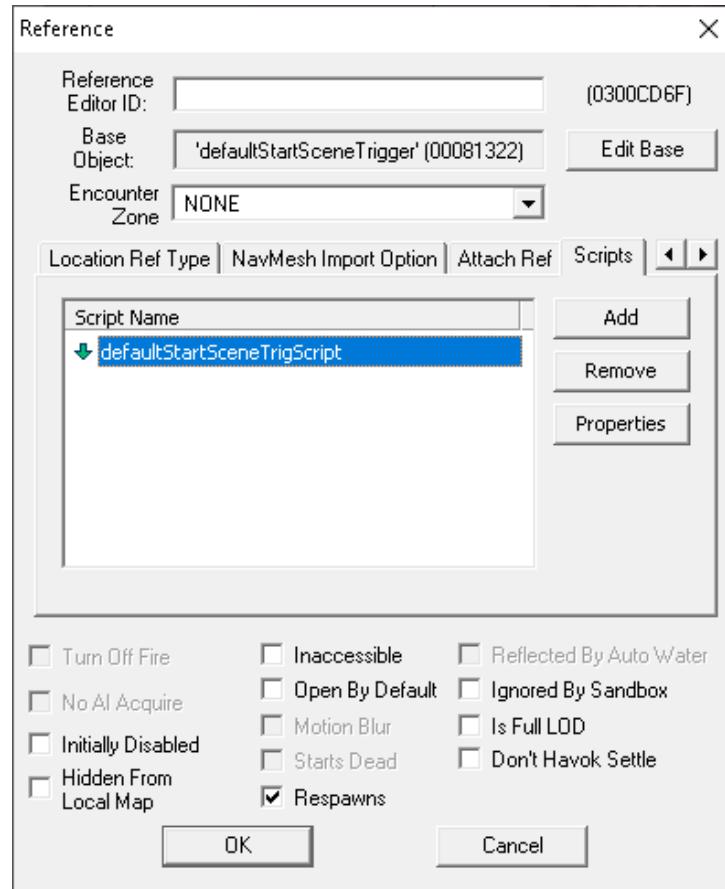


Figure 910 - Scripts tab in Reference properties.

Highlight the defaultStartSceneTrigScript script and select Properties.

First, click on doOnce then click on the Edit Value button.

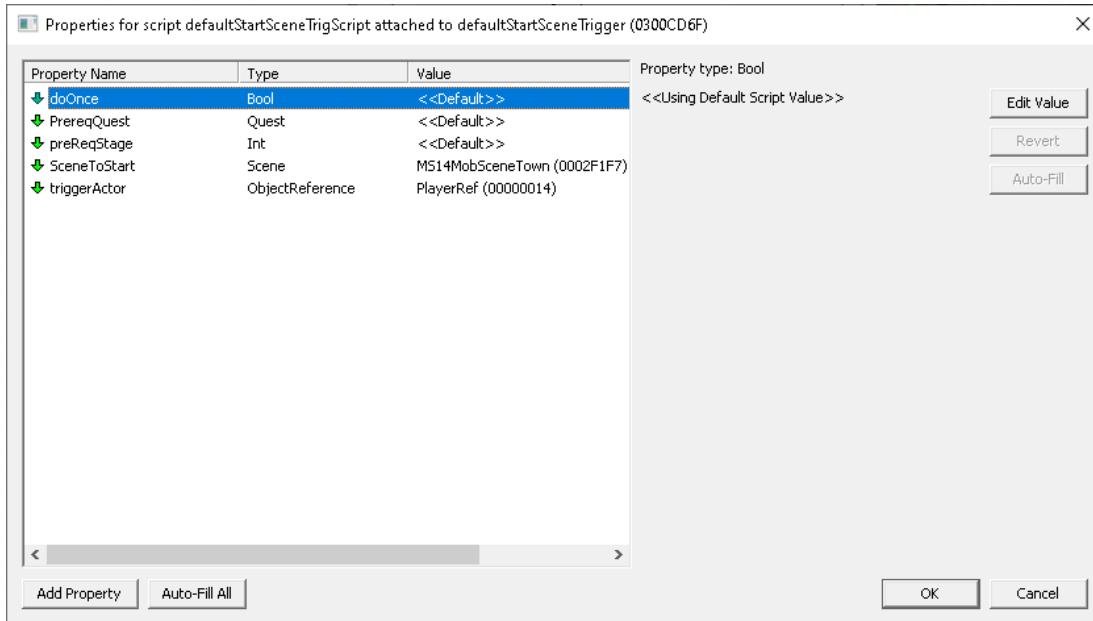


Figure 911 - Changing the default value on doOnce.

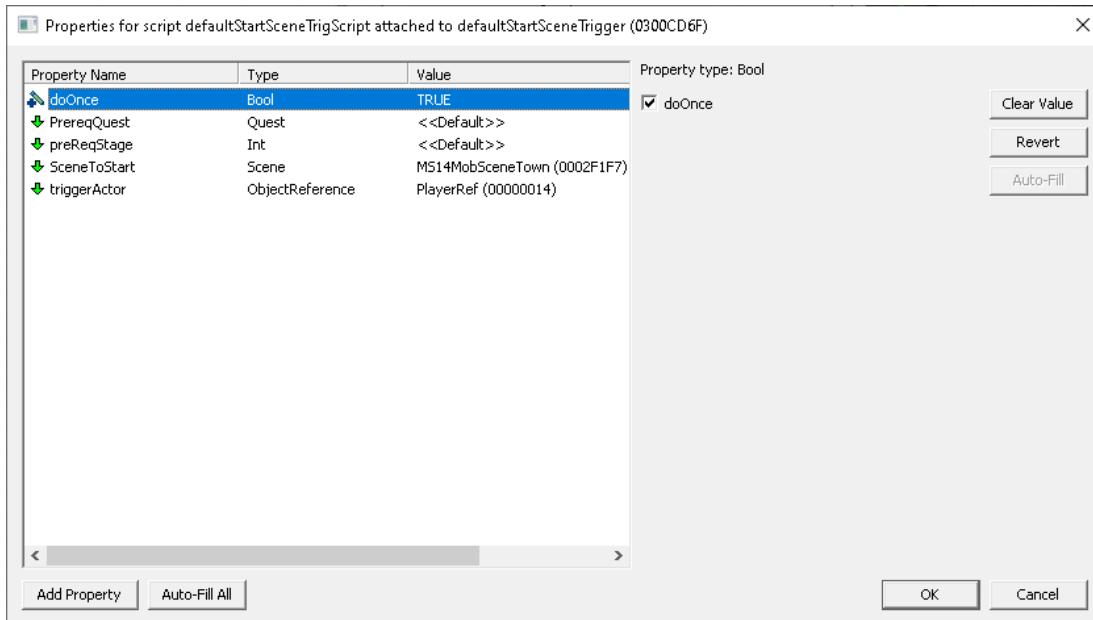


Figure 912 - doOnce set to TRUE.

Next, go to SceneToStart and set the Pick Object drop-down to the ID of scene we want to run when this script is triggered.

For our example, that's going to be WITTestQuestSampleScene.

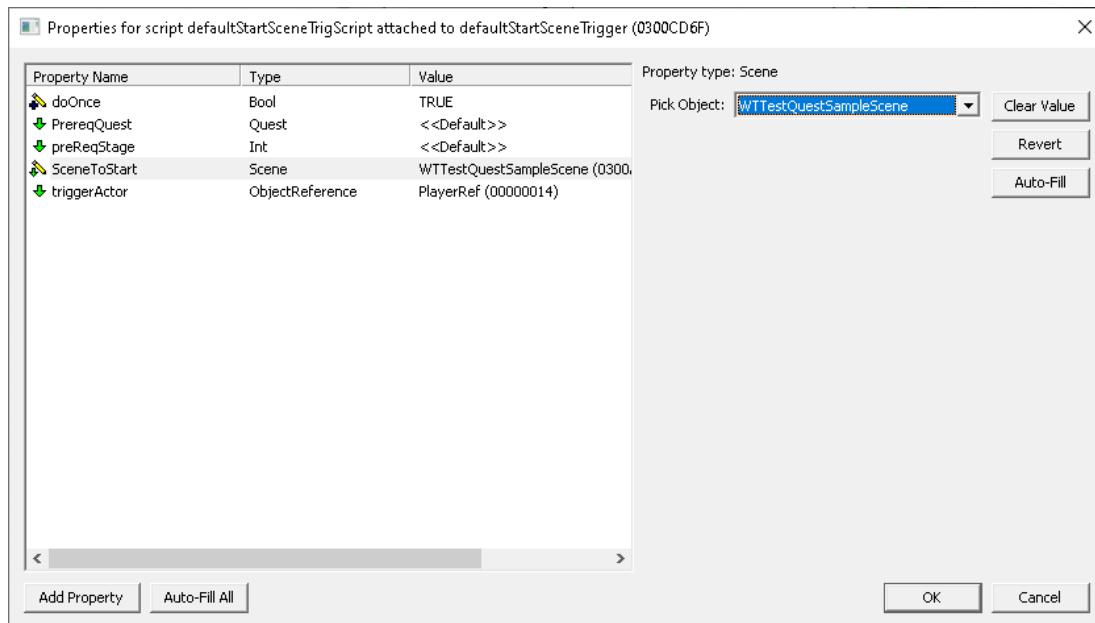


Figure 913 - Setting SceneToStart to WITTestQuestSampleScene.

By default, the script will be triggered when the player enters the trigger box. But what if the NPCs are lagging far behind? We can set this scene to occur when they enter the trigger box instead.

Go to triggerActor and set it to WTFollowerAthir. His in-game reference is in the WyrmstoothHolderCell interior cell.

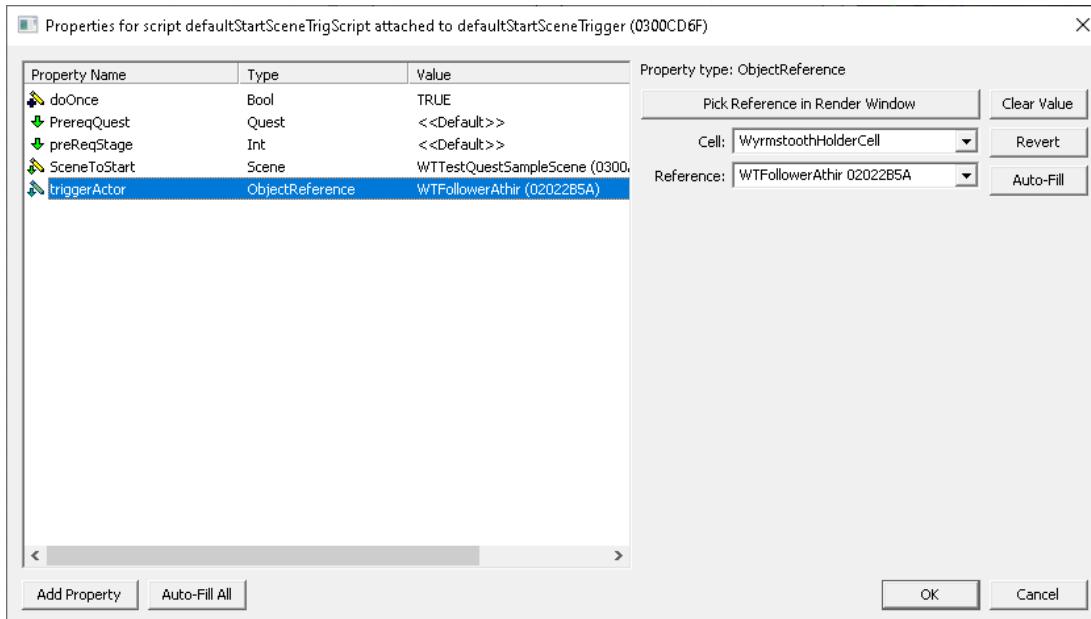


Figure 914 - Setting the NPC to trigger the script.

Click OK to close out of script properties.

Click OK to close out of Reference properties.

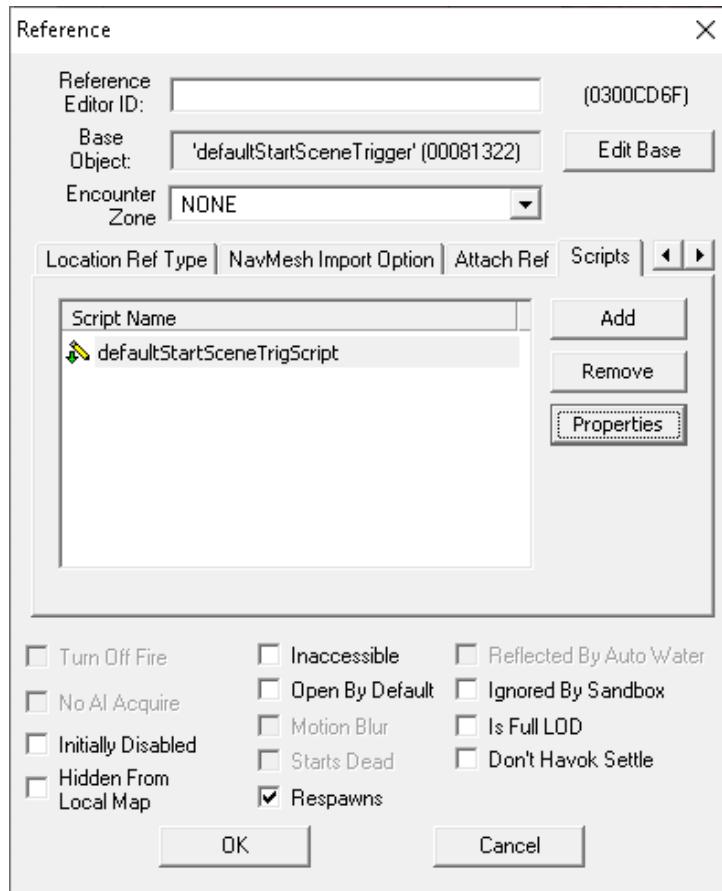


Figure 915 - defaultStartSceneTrigScript properties now flagged as modified.

Our scene will now start once Athir passes through the trigger box instead.

USING TALKING ACTIVATORS

Talking Activators can be used when dialogue needs to be delivered by an inanimate object.

For example, in Wyrmsooth talking activators are used by Vulom both as a voice in the player's head and when talking to the skull. When Lurius Liore delivers his lines during the intro boat ride, a talking activator is used and animated alongside the ship.

Talking Activators can be found in the Object Window under Actors > TalkingActivator.

The easiest way to create a new Talking Activator is to duplicate an existing one. Right-click on, say, DA01VoiceofAzura and select Duplicate.

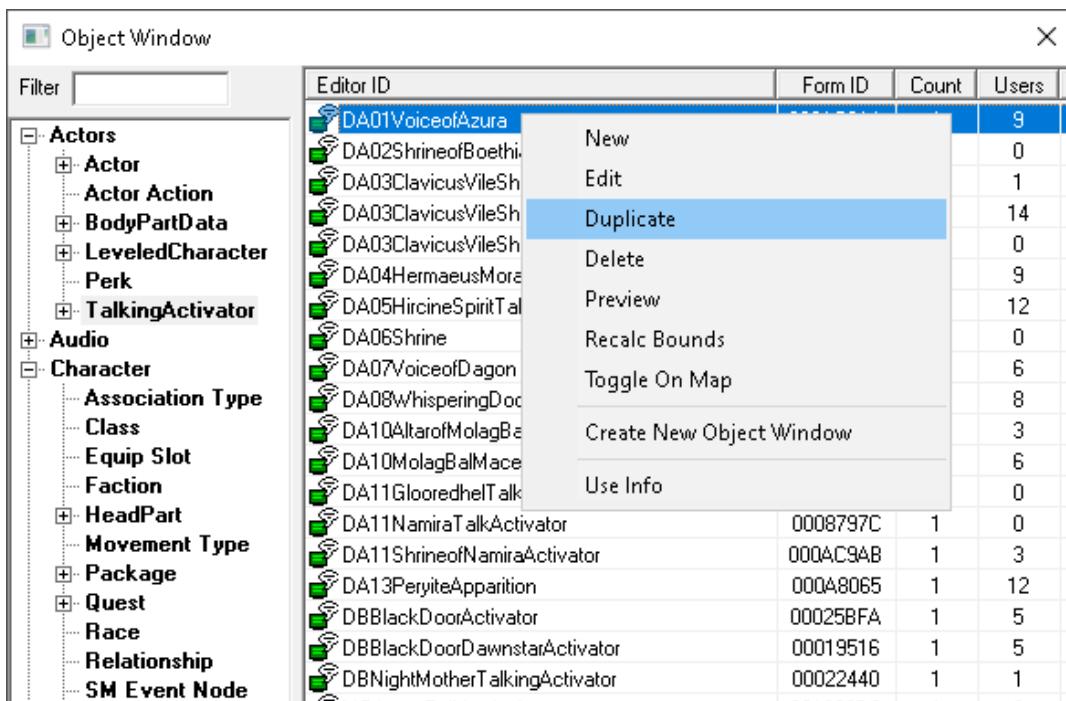


Figure 916 - Duplicating an existing talking activator.

Set a new form ID in the ID field. For this example I'm going to show you how to set up the Vulom talking activators, both for the voice in the player's head and the skull that the player can talk to. I'm going to start with the voice in the player's head so I just set the ID to WTVulomSceneTalkActivator since we'll be using this talking activator in a scene.

Enter a name into the Name field. This name is used when the player interacts with the talking activator.

The Model field defines the type of object we'll be placing in the world. Since we're copying the DA01VoiceofAzura form, this will be set to Markers\Misc\InvisibleCollision01.nif which will be an invisible marker.

It's important to set the Voice Type drop-down to match the voice type of the voice acting. For this example, I'm going to set it to WTVoiceType.

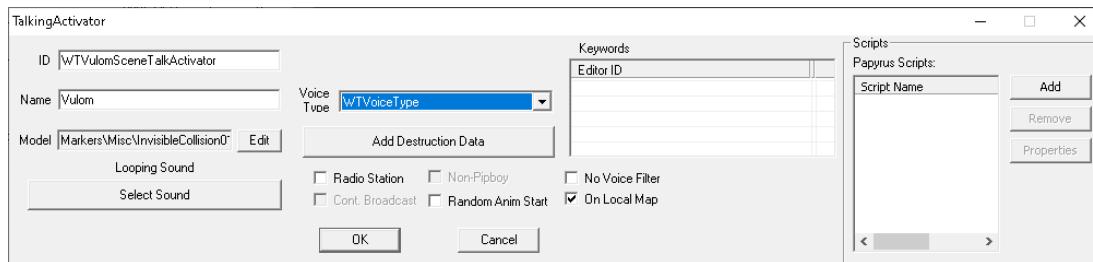


Figure 917 - First talking activator configuration.

Click OK to close out of Talking Activator properties.

When prompted to create a new form, select No.

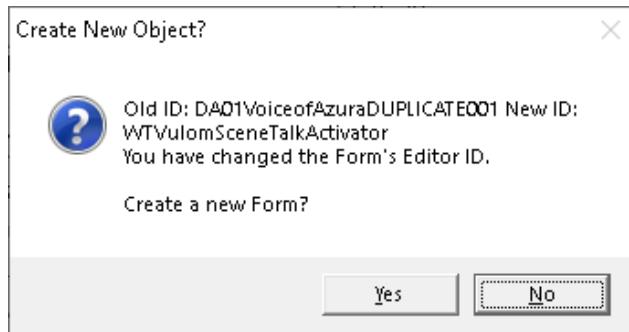


Figure 918 - Create New Object pop-up.

When prompted to confirm rename, select Yes.

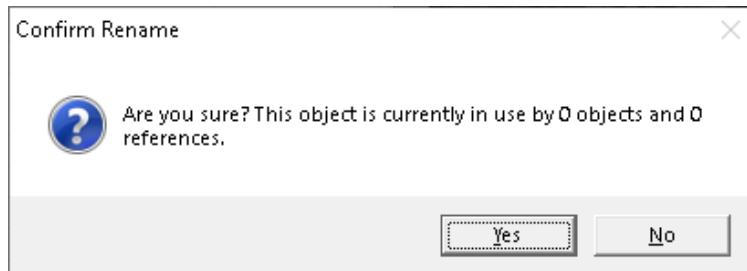


Figure 919 - Confirm Rename pop-up.

Drag and drop the talking activator from the Object Window into the render window to place it in the world.

I placed the WTVulomSceneTalkActivator near Vulom's sarcophagus, beneath the ground. Later I'll be changing the audio override on the voice acting that the talking activator will deliver to make it hearable from any location in this cell regardless of the player's position.

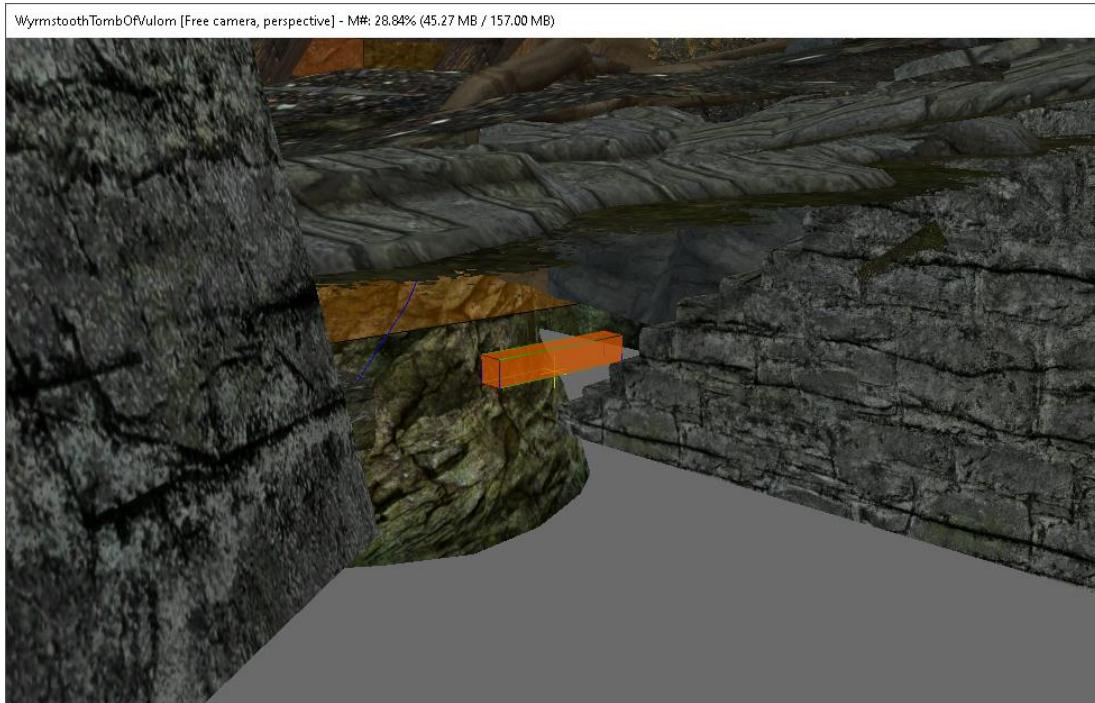
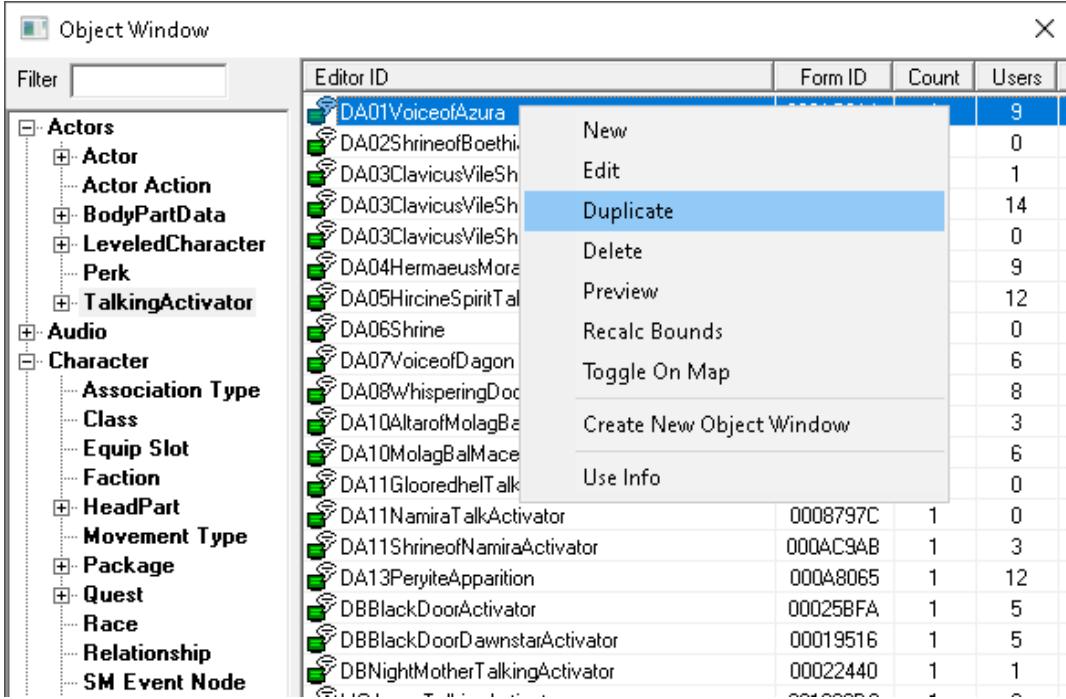


Figure 920 - Placing the first talking activator.

Next, let's create the second talking activator we'll need for this example; the skull that players can interact with directly.

Go back to the Object Window, right-click on one of the existing talking activators and select Duplicate.



Editor ID	Form ID	Count	Users
DA01VoiceofAzura	New	9	0
DA02ShrineofBoethi	Edit	0	1
DA03ClavicusVileSh	Duplicate	14	0
DA03ClavicusVileSh	Delete	0	9
DA03ClavicusVileSh	Preview	12	0
DA04HermaeusMora	Recalc Bounds	6	0
DA05HircineSpiritTal	Toggle On Map	8	0
DA06Shrine	Create New Object Window	3	0
DA07VoiceofDagon	Use Info	6	0
DA08WhisperingDoc		0	0
DA10AltarofMolagBa		0	3
DA10MolagBaMace		0	12
DA11GlooredhellTalk		0	5
DA11NamiraTalkActivator	0008797C	1	0
DA11ShrineofNamiraActivator	000AC9AB	1	3
DA13PeryiteApparition	000A8065	1	5
DBBlackDoorActivator	00025BFA	1	5
DBBlackDoorDawnstarActivator	00019516	1	5
DBNightMotherTalkingActivator	00022440	1	1

Figure 921 - Duplicating an existing talking activator.

For the second talking activator, I set its ID to WTVulomSkullTalkActivator.

Set the name in the Name field. Again, I set this to Vulom.

This time, I set the Model to Clutter\Bones\HumanSkullFull.nif.

The Voice Type was also set to WTVoiceType.

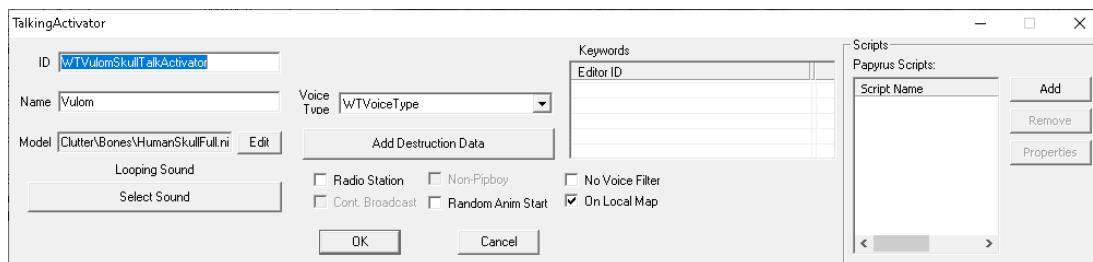


Figure 922 - Second talking activator configuration.

Note: You can retexture the .nif you set here as the talking activator's world model if you want to by clicking on Edit.

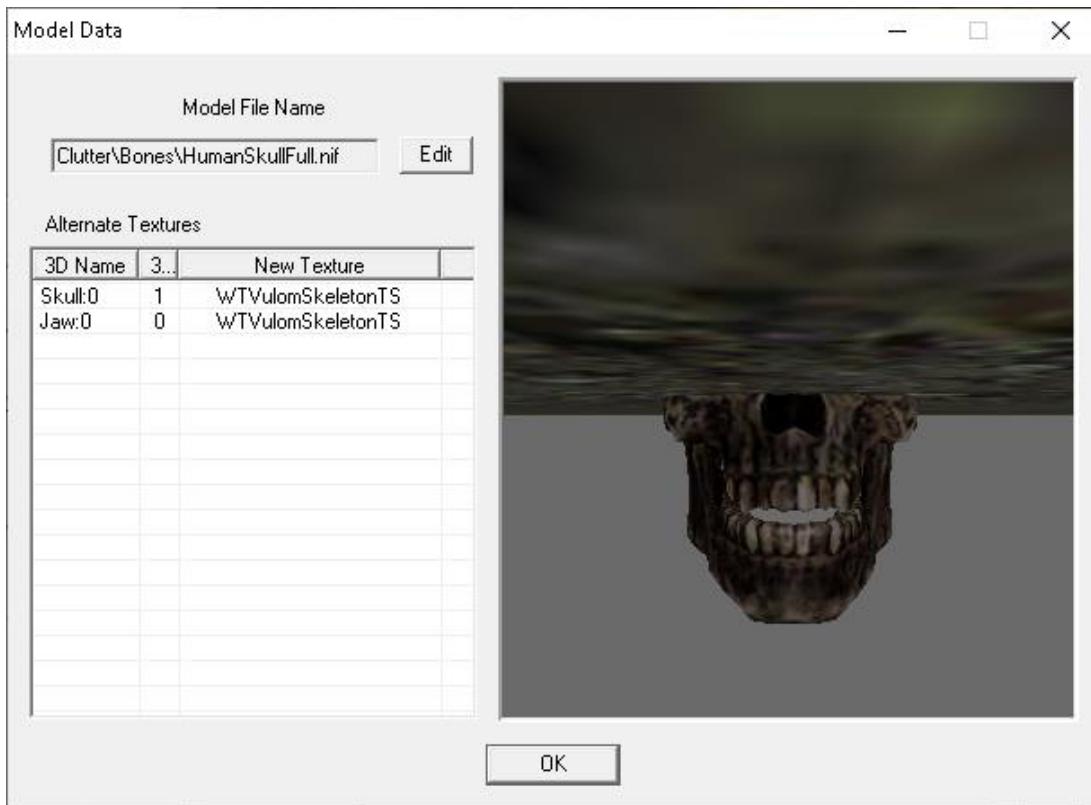


Figure 923 - Retexturing a .nif by defining a texture set.

To add a texture set to a specific part of the .nif, double-click on Skull:0 or Jaw:0.

Highlight the texture set in the form list then click OK.

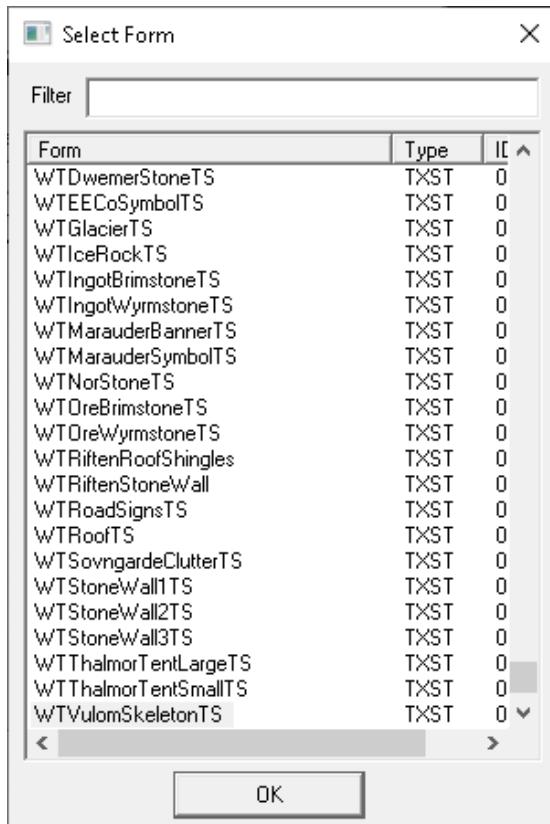


Figure 924 - Texture set form list.

Click OK to close out of Model Data properties if you changed a texture, then click OK to close out of Talking Activator properties.

Again, when prompted to create a new form, select No.

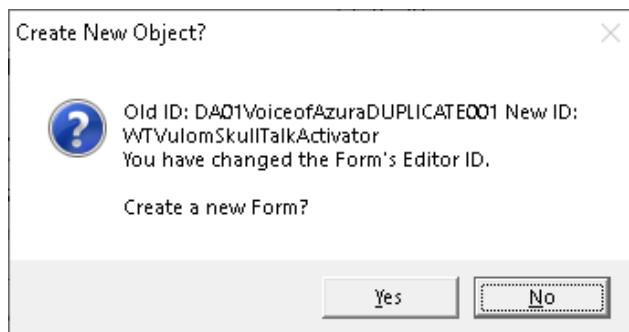


Figure 925 - Create New Object pop-up.

And when prompted to confirm rename, select Yes.

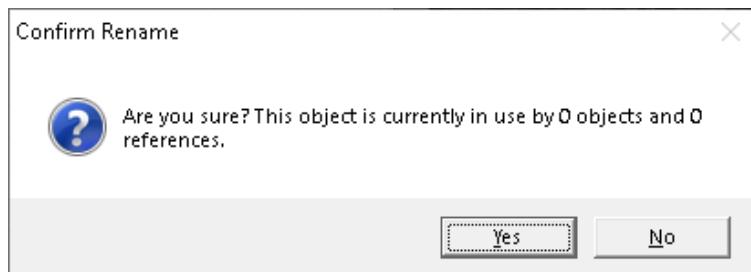


Figure 926 - Confirm Rename pop-up.

Drag and drop the second talking activator to place it. It's important to ensure that it is not clipping with the collision of any surrounding objects so the player can activate it from their position.

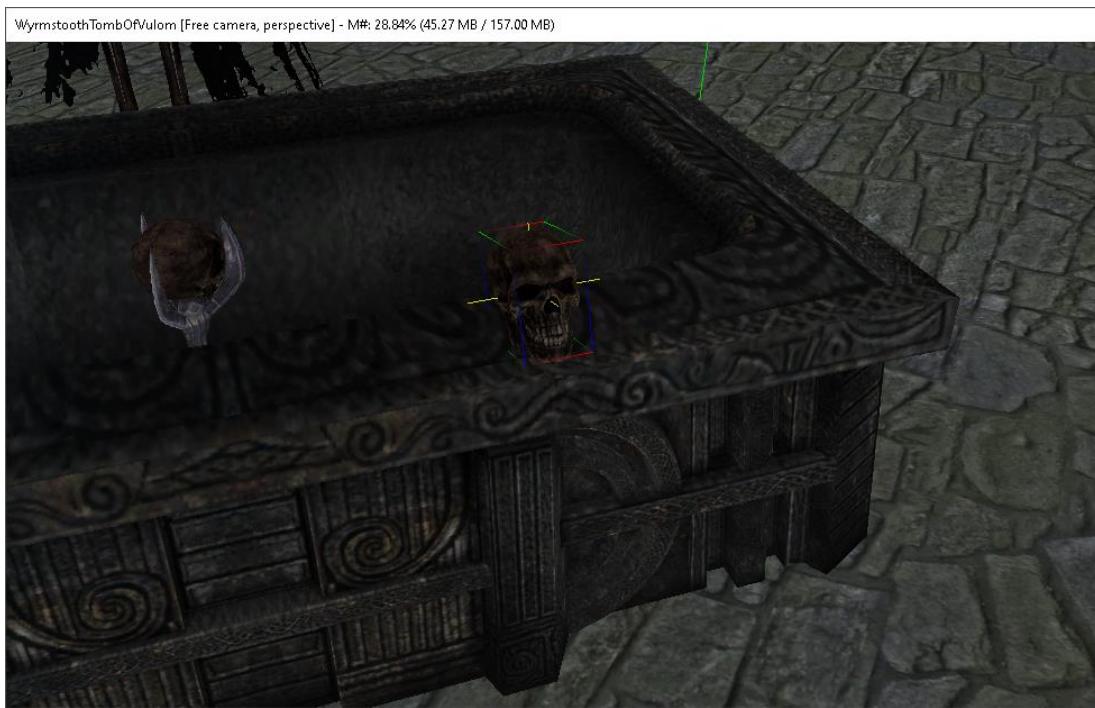


Figure 927 - Placing the second talking activator.

Next, we'll take a look at how Vulom's dialogue was set up in his associated side-quest.

If you need step-by-step instructions on setting up the quest dialogue, see the sections [Adding Dialogue to a Quest](#), [Adding Scenes](#), and [Adding Voice Acting and Lip Syncing](#).

The first thing I did was add both talking activators into Vulom's quest as quest aliases.

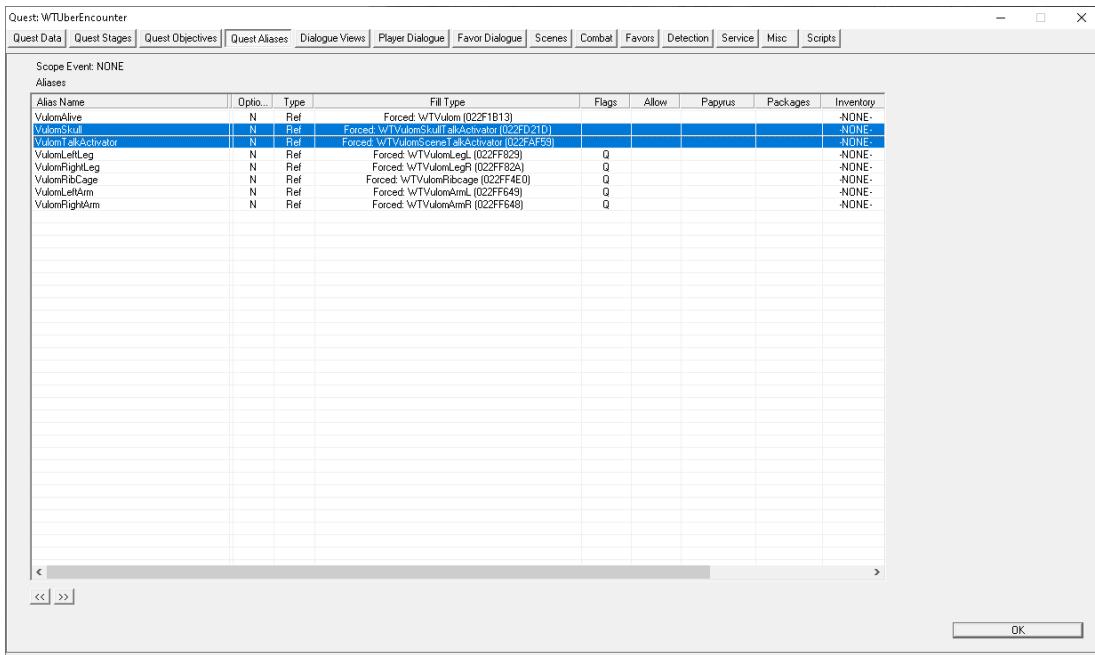


Figure 928 - Talking activators added as quest aliases.

When setting up the alias, just set the Fill Type to Specific Reference and select the reference in the render window. For talking activators, there's no need to tick any other options.

The first talking activator uses a scene which is triggered by the player when they enter Vulom's tomb for the first time.

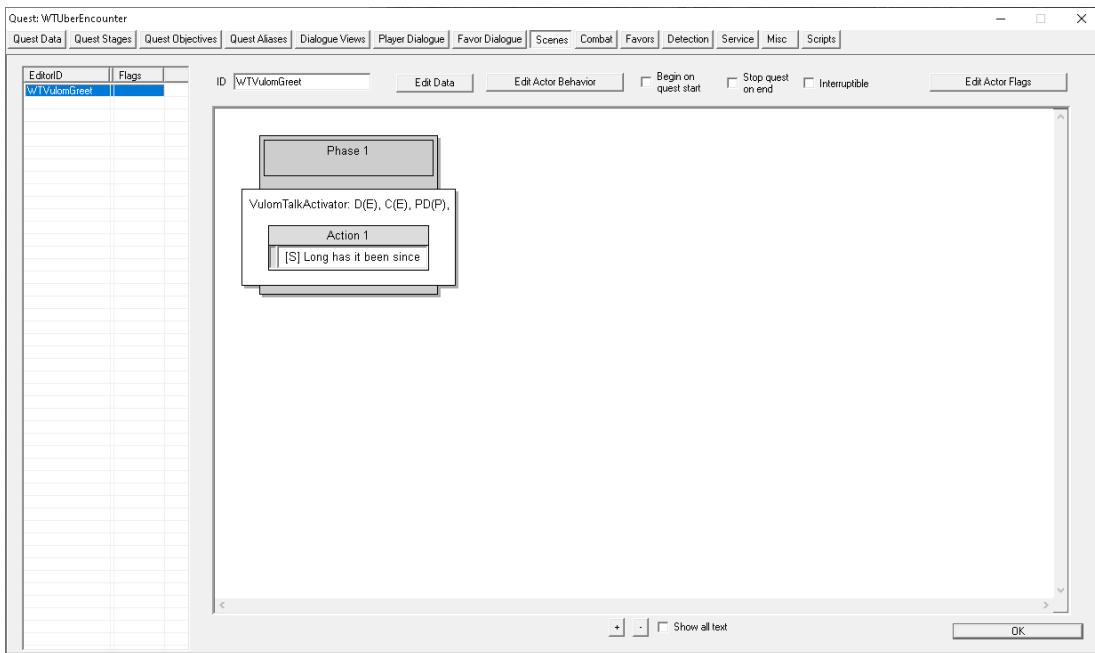


Figure 929 - Scene for the first talking activator.

This scene consists of a single phase with the first talking activator set up to deliver a single line of dialogue.

Because the talking activator itself is far away from the player when it's activated, I ticked 'Force Subtitle' to ensure subtitles are always shown regardless of proximity to the speaker.

To ensure the voice acting can be heard in the player's head, I set the Audio Override to SOMDialogue2D.

Note: There's no need to set any conditions.

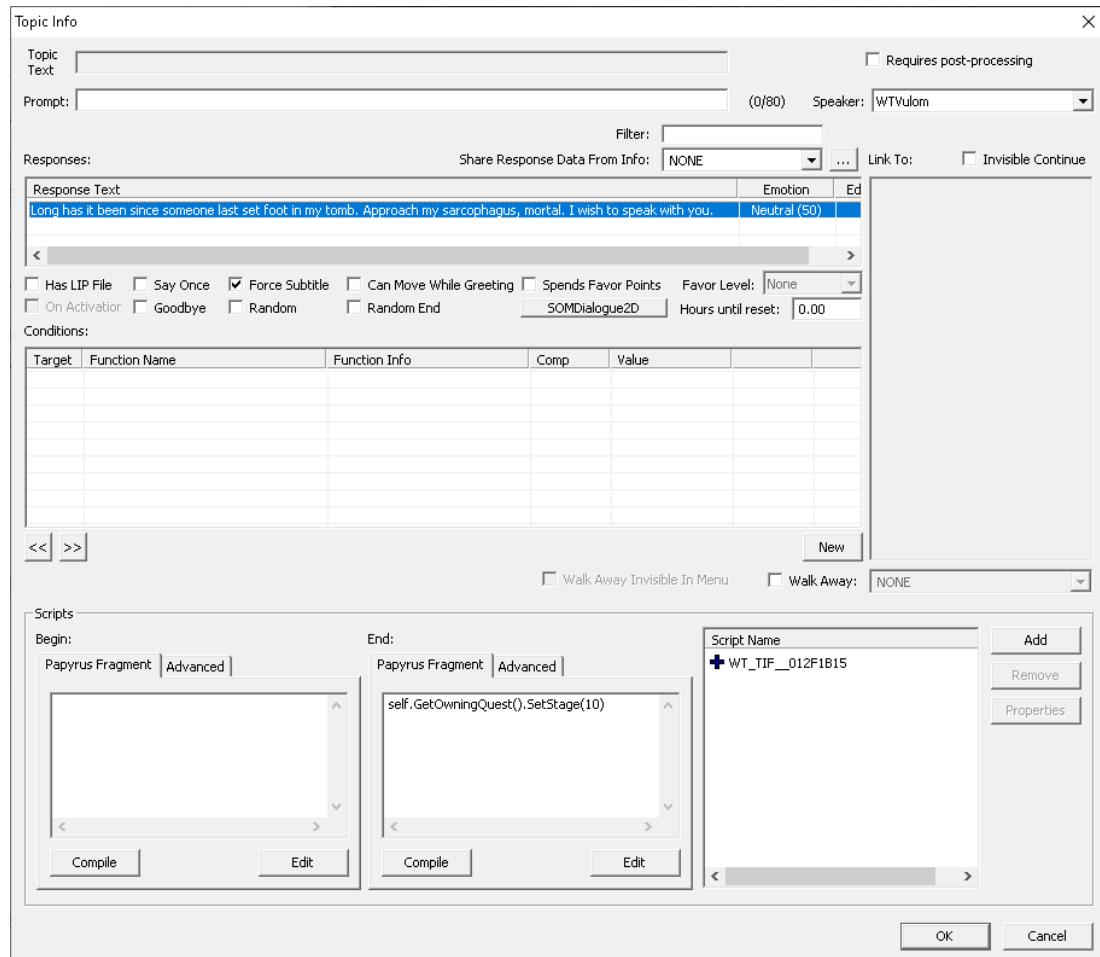


Figure 930 - Topic Info setup for the first talking activator.

Once the line of dialogue has been spoken, Vulom's quest is advanced to stage 10 which puts an objective in the player's journal.

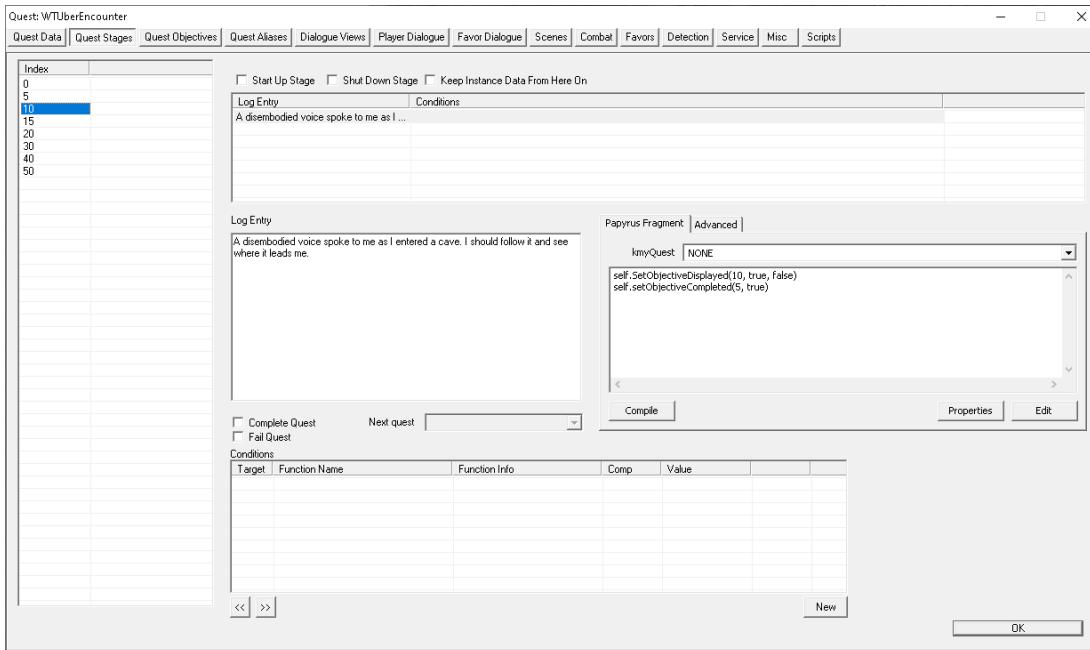


Figure 931 - Stage 10 of Vulom's quest.

A defaultStartSceneTrigger trigger is placed at the entrance to the large cavern containing Vulom's sarcophagus to start the scene.

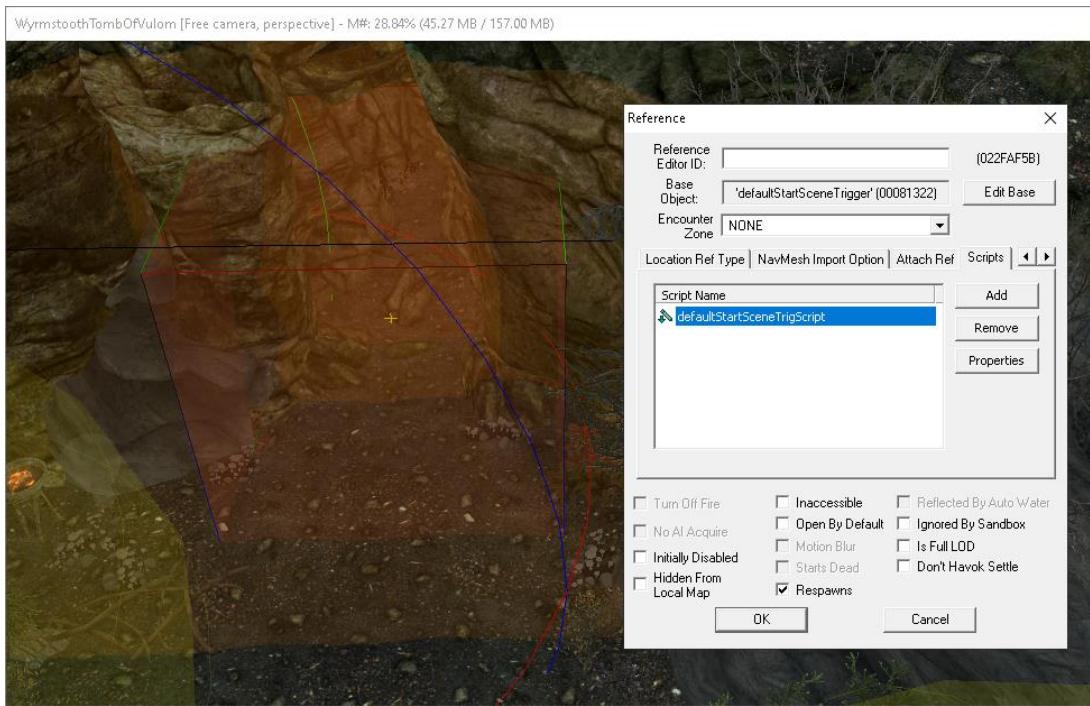


Figure 932 - defaultStartSceneTrigger used to start the first talking activator's scene.

This script is configured to run only once and to run the WTVulomGreet scene. By default, this trigger can only be activated by the player.

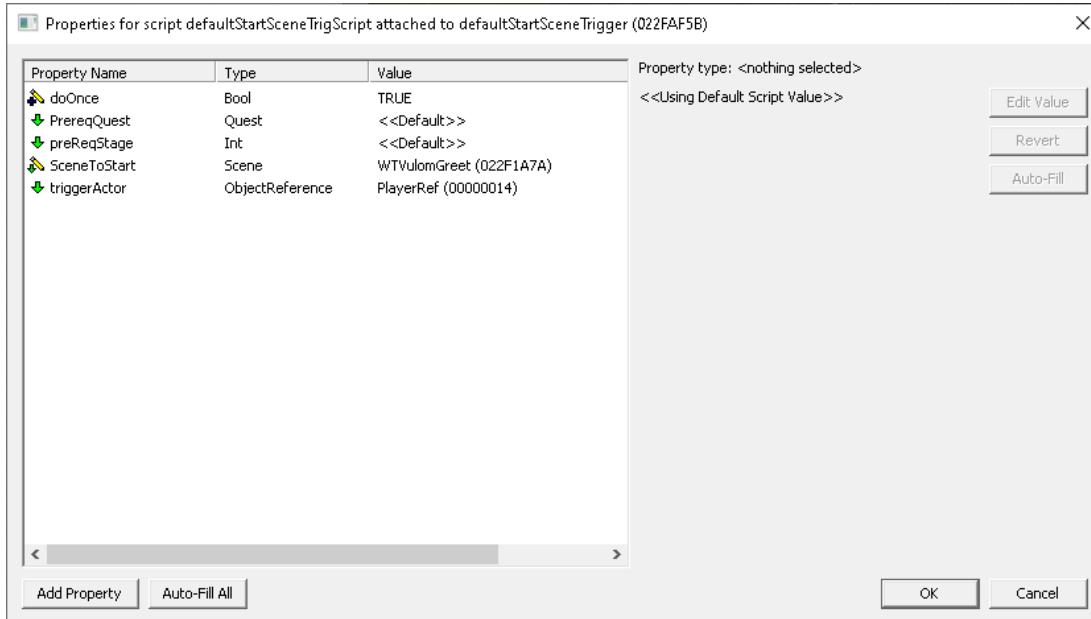


Figure 933 - Script properties on the defaultStartSceneTrigScript script.

Dialogue for the second talking activator is set up in Dialogue Views.

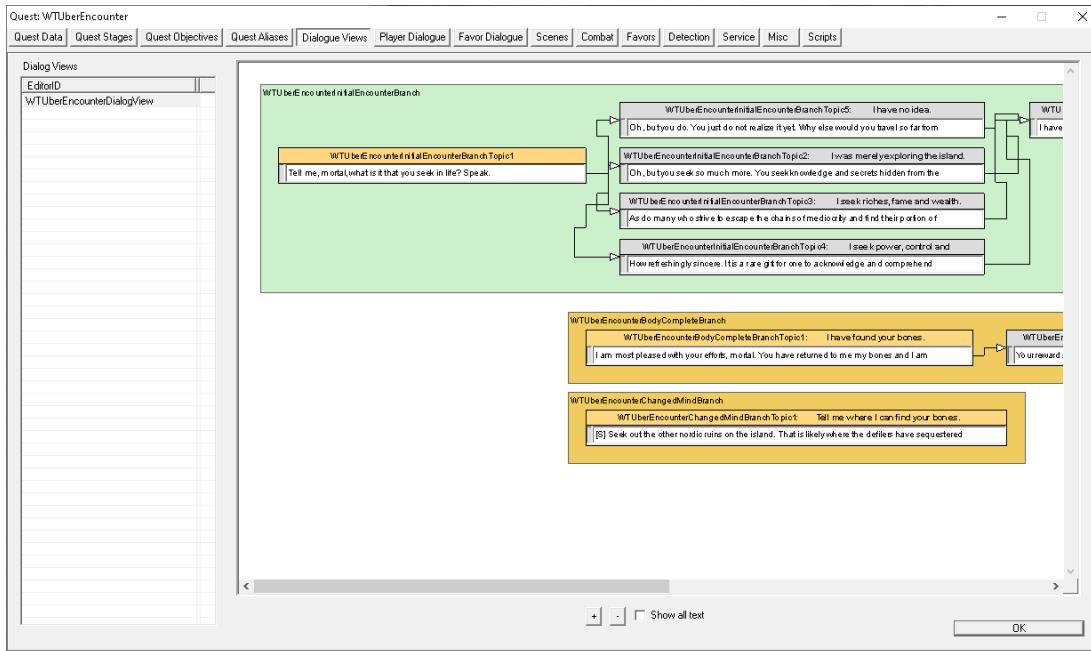


Figure 934 - The second talking activator's dialogue.

Each topic has two conditions; one to ensure the player is speaking with the second talking activator, Vulom's skull, and the other to check the current stage of Vulom's quest.

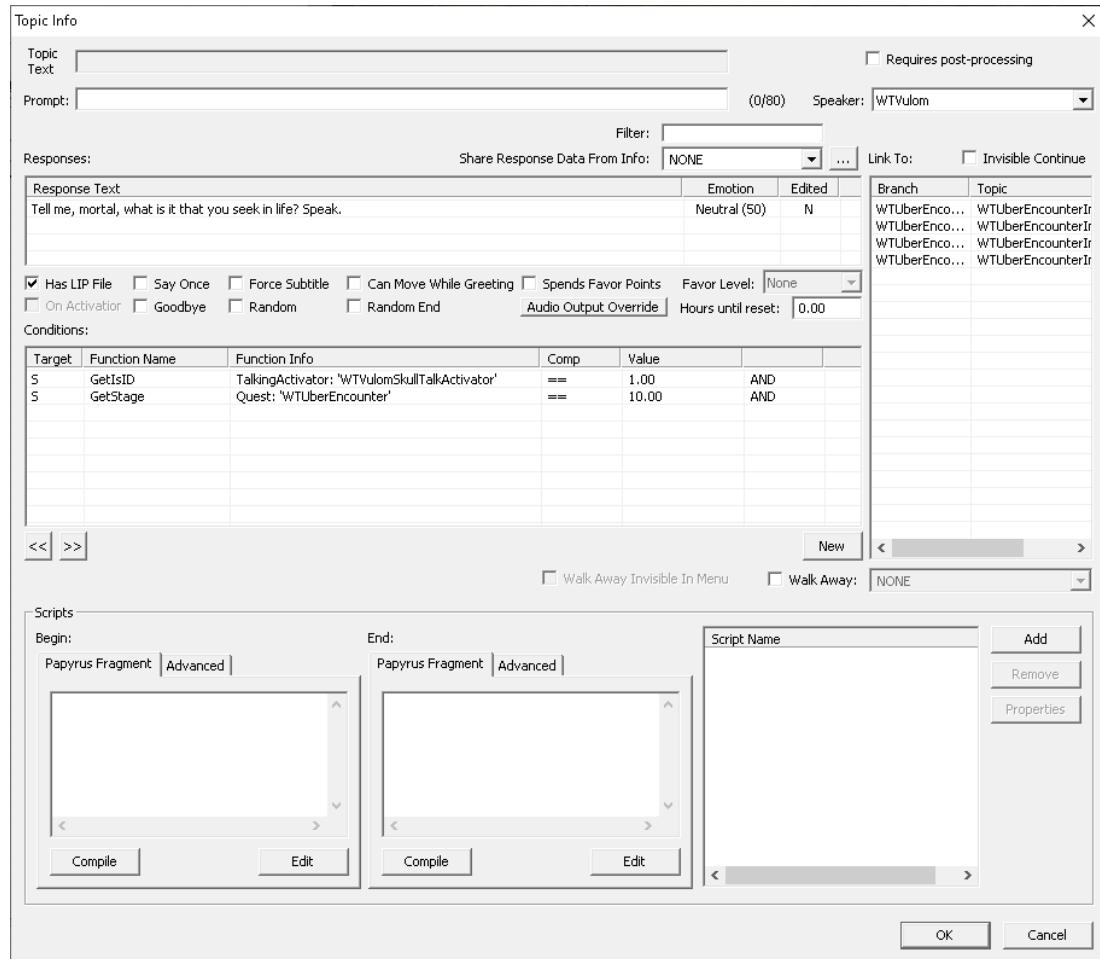


Figure 935 - Topic Info setup for the second talking activator's dialogue.

CREATING AN MCM

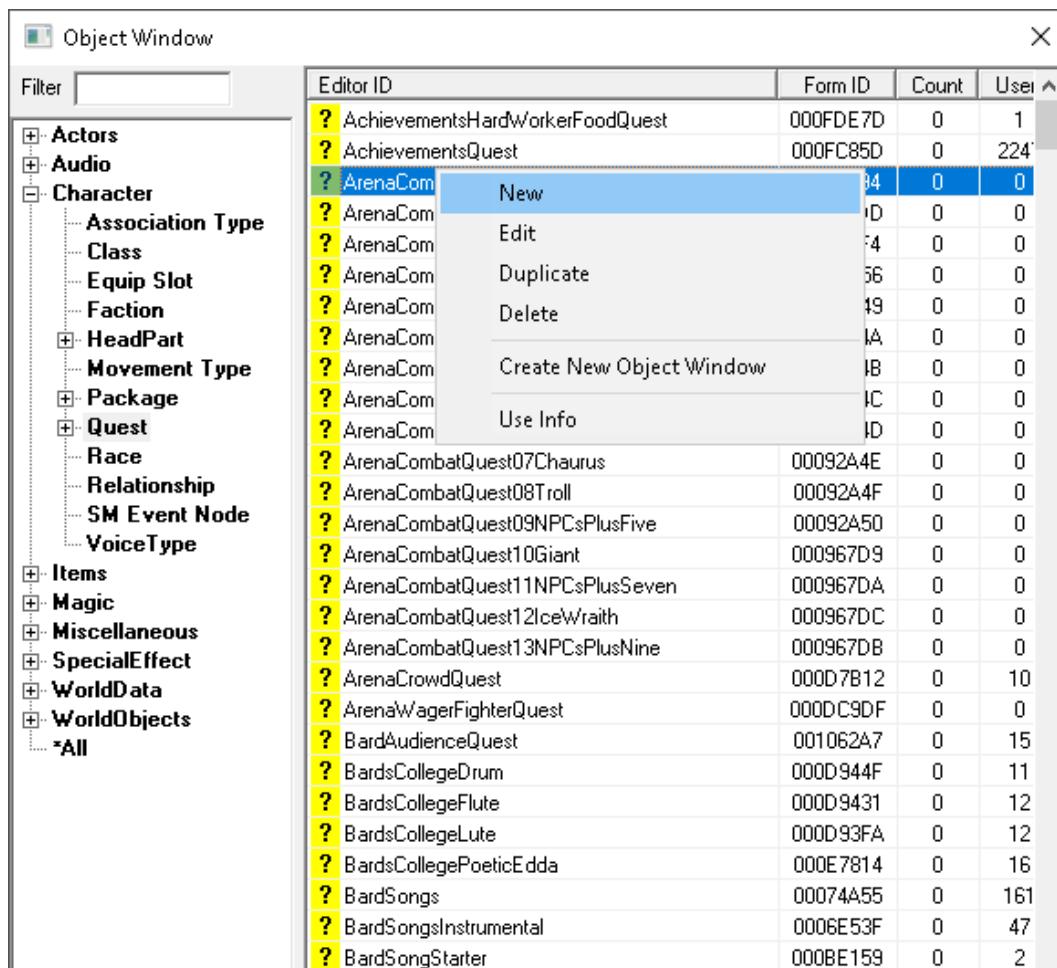
In this section I'll be going over the basics for setting up an MCM for your mod.

Note: You will need to install [SKSE](#) and SkyUI ([LE/SSE](#)).

This section will require a fair bit of papyrus scripting. If scripting makes you squeamish, you're probably not going to enjoy this. That said, I've tried to explain each line of code to make it as easy as possible to understand.

Let's start by setting up a new quest for the MCM script.

In the Object Window, go to Character > Quest, right-click on one of the existing quests and select New.



The screenshot shows the Object Window interface. On the left is a tree view of object categories: Actors, Audio, Character (with sub-options like Association Type, Class, Equip Slot, Faction, HeadPart, Movement Type, Package, Quest, Race, Relationship, SM Event Node, VoiceType), Items, Magic, Miscellaneous, SpecialEffect, WorldData, and WorldObjects. The 'Quest' category is currently selected. On the right is a table view of quest entries, with columns for Editor ID, Form ID, Count, and User. The table lists various quests such as AchievementsHardWorkerFoodQuest, AchievementsQuest, ArenaCom (which is selected and has a context menu open with 'New' highlighted), ArenaCombatQuest07Chaurus, ArenaCombatQuest08Troll, ArenaCombatQuest09NPCsPlusFive, ArenaCombatQuest10Giant, ArenaCombatQuest11NPCsPlusSeven, ArenaCombatQuest12IceWraith, ArenaCombatQuest13NPCsPlusNine, ArenaCrowdQuest, ArenaWagerFighterQuest, BardAudienceQuest, BardsCollegeDrum, BardsCollegeFlute, BardsCollegeLute, BardsCollegePoeticEdda, BardSongs, BardSongsInstrumental, and BardSongStarter. The 'ArenaCom' row has a sub-menu with options like New, Edit, Duplicate, Delete, Create New Object Window, Use Info, and ID.

Editor ID	Form ID	Count	User
? AchievementsHardWorkerFoodQuest	000FDE7D	0	1
? AchievementsQuest	000FC85D	0	224
ArenaCom	New	0	0
? ArenaCom	Edit	0	0
? ArenaCom	Duplicate	0	0
? ArenaCom	Delete	0	0
? ArenaCom	Create New Object Window	0	0
? ArenaCom	Use Info	0	0
? ArenaCombatQuest07Chaurus	00092A4E	0	0
? ArenaCombatQuest08Troll	00092A4F	0	0
? ArenaCombatQuest09NPCsPlusFive	00092A50	0	0
? ArenaCombatQuest10Giant	000967D9	0	0
? ArenaCombatQuest11NPCsPlusSeven	000967DA	0	0
? ArenaCombatQuest12IceWraith	000967DC	0	0
? ArenaCombatQuest13NPCsPlusNine	000967DB	0	0
? ArenaCrowdQuest	000D7B12	0	10
? ArenaWagerFighterQuest	000DC9DF	0	0
? BardAudienceQuest	001062A7	0	15
? BardsCollegeDrum	000D944F	0	11
? BardsCollegeFlute	000D9431	0	12
? BardsCollegeLute	000D93FA	0	12
? BardsCollegePoeticEdda	000E7814	0	16
? BardSongs	00074A55	0	161
? BardSongsInstrumental	0006E53F	0	47
? BardSongStarter	000BE159	0	2

Figure 936 - Creating a new quest for the MCM.

Enter in a quest ID. For this example I entered WITTestMCM.

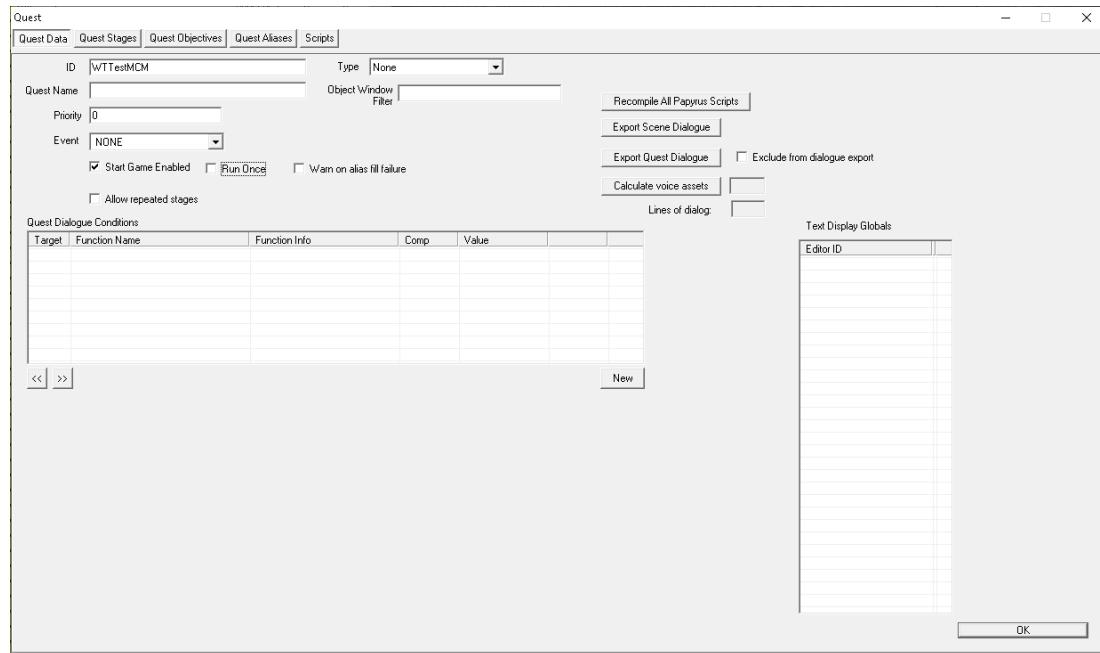


Figure 937 - Quest Data tab.

Before continuing, click OK to close out of Quest properties. Right-click on the quest we just created and select Edit to re-open its properties.

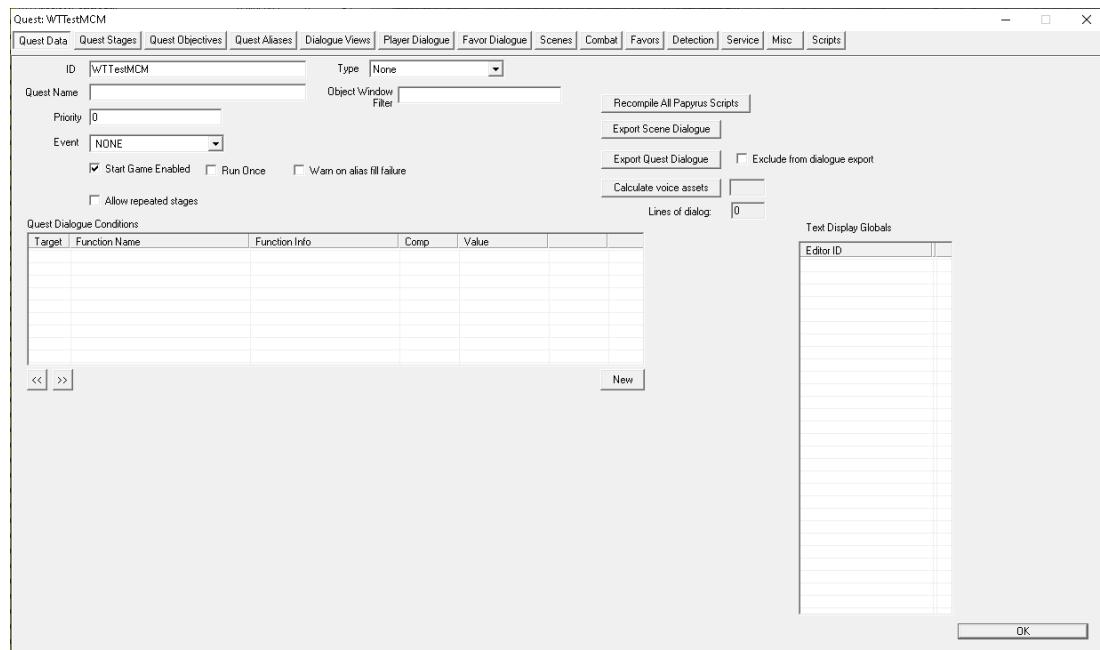


Figure 938 - Quest properties after closing and re-opening it.

We should now see more tabs than before.

Go to the Quest Aliases tab, right-click in the aliases list and select New Reference Alias.

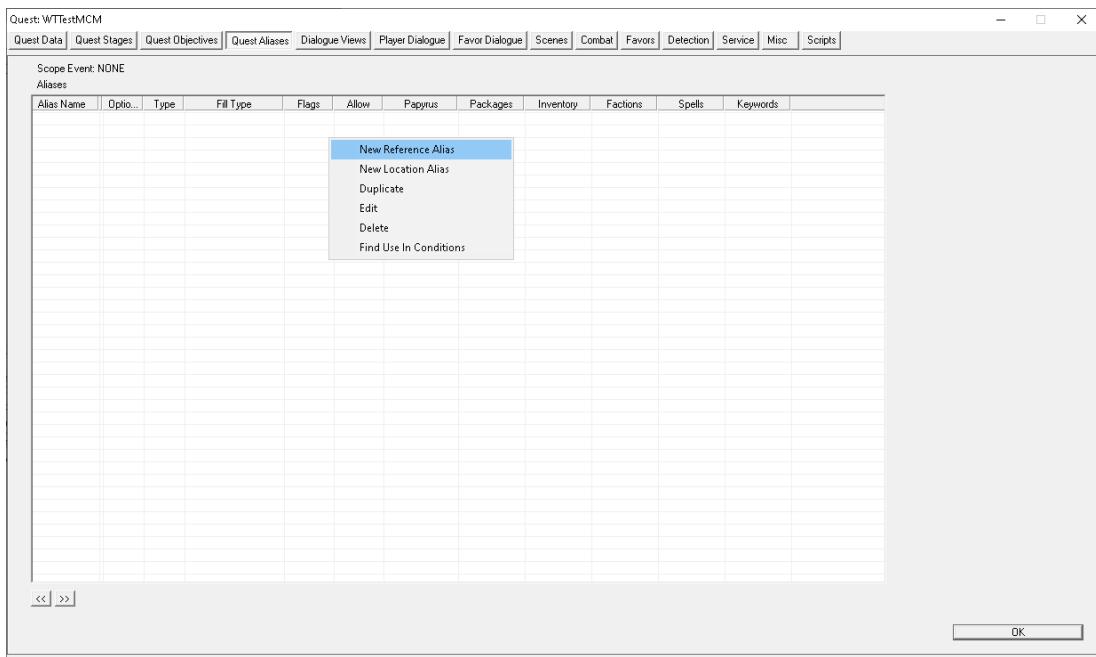


Figure 939 - Creating a new reference alias.

Set the Alias Name field to PlayerREF.

Under the Scripts section, click on the Add button.

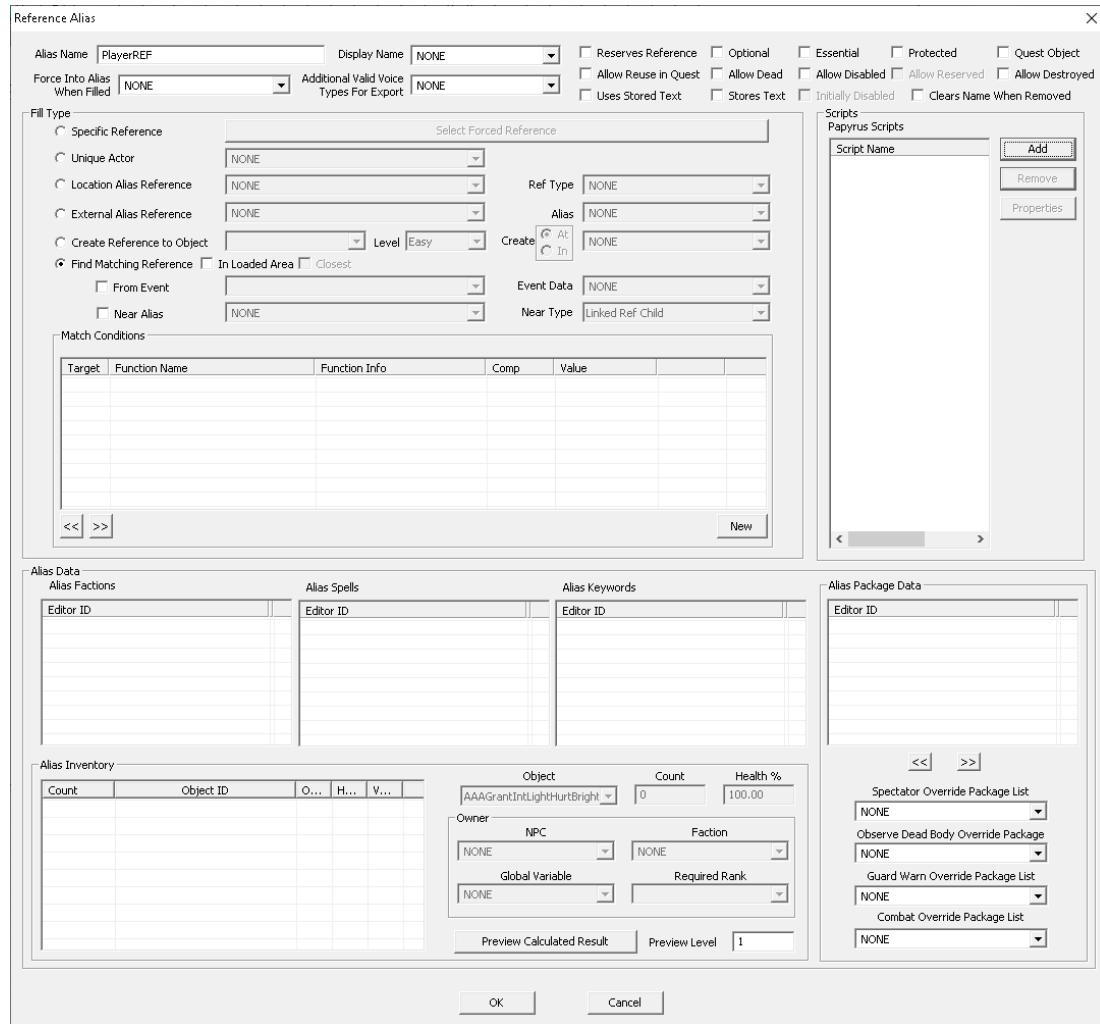


Figure 940 - Setting up the PlayerREF alias.

Filter by 'ski_player', click on SKI_PlayerLoadGameAlias in the script list to highlight it then click OK.

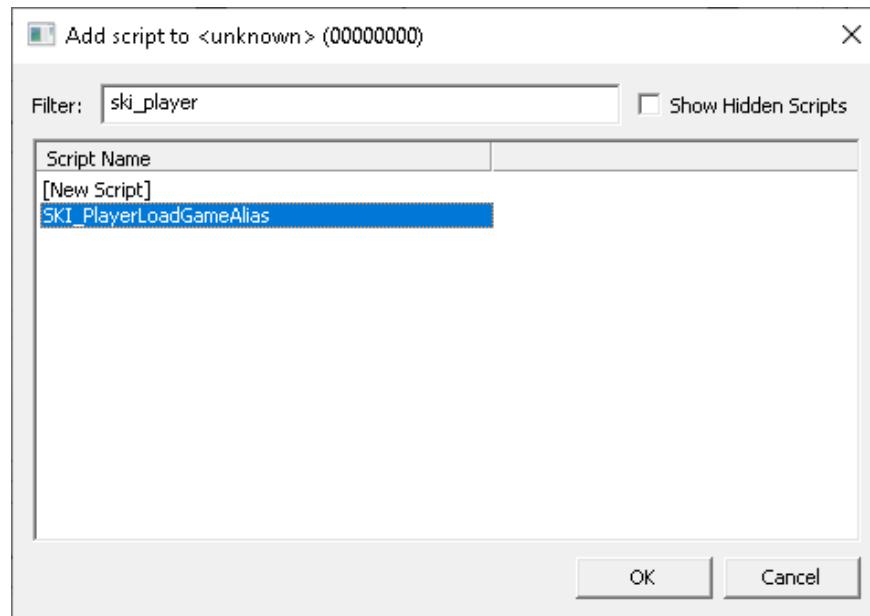


Figure 941 - Adding SKI_PlayerLoadGameAlias to the PlayerREF alias.

Click OK to close out of Reference Alias properties.

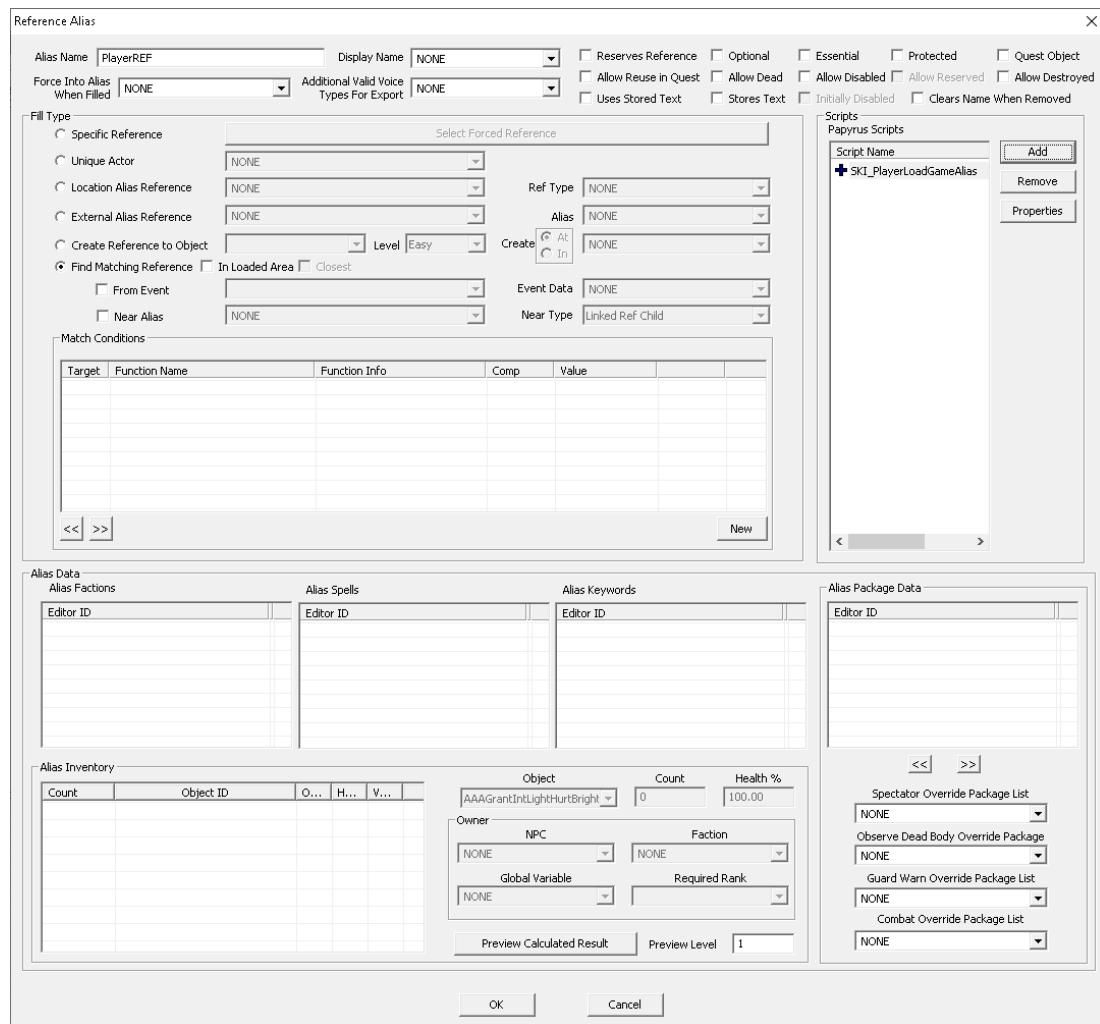


Figure 942 - PlayerREF alias with SKI_PlayerLoadGameAlias script attached.

Go to the Scripts tab and click on the Add button.

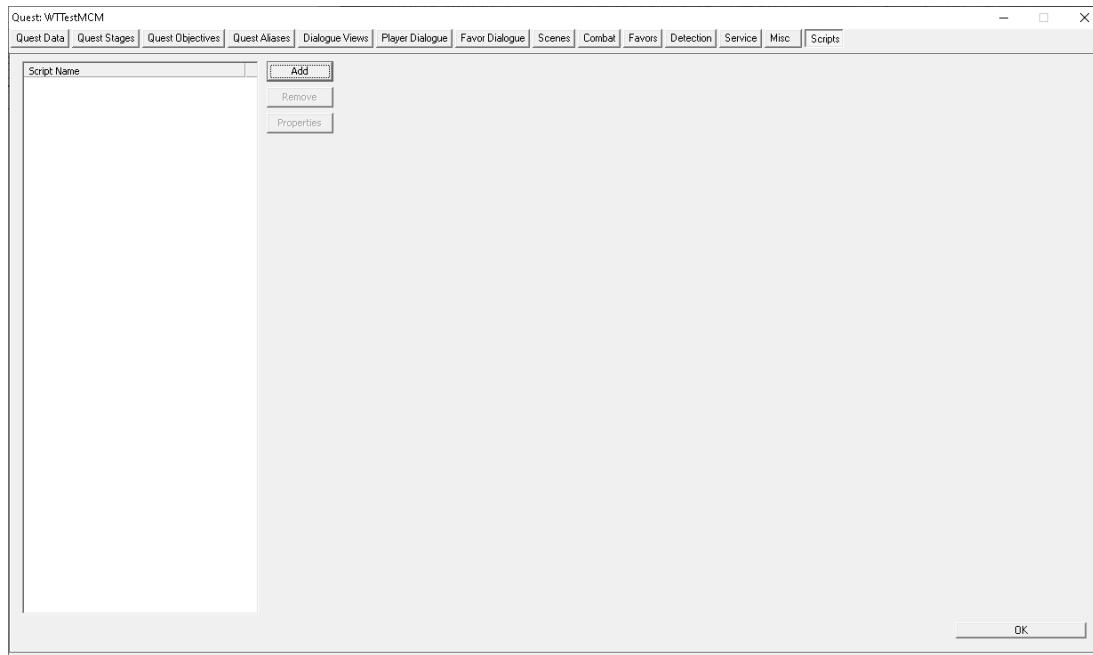


Figure 943 - Adding a new quest script.

Click on [New Script] to highlight it then click OK.

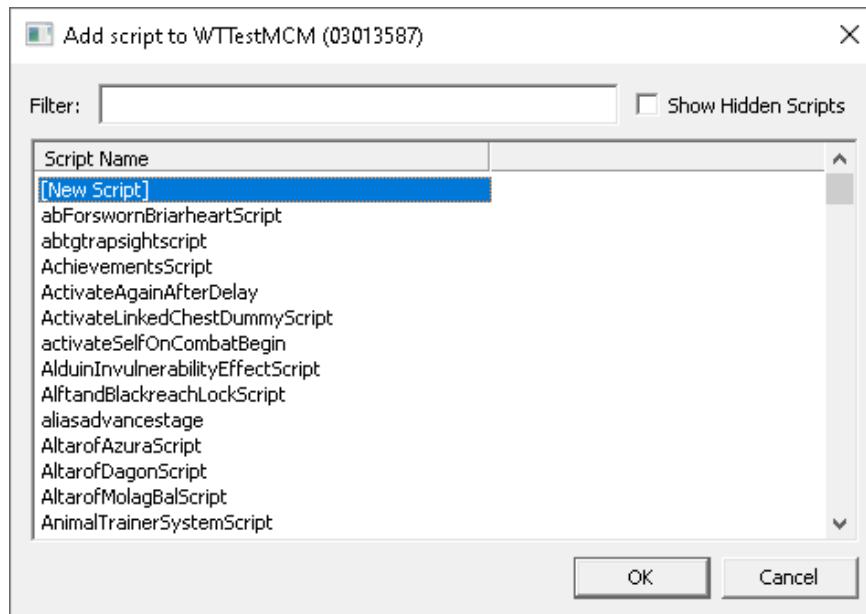


Figure 944 - Creating a new script.

Enter the name of the new script in the Name field.

In the Extends field, enter ‘ski_configbase’ then click OK.

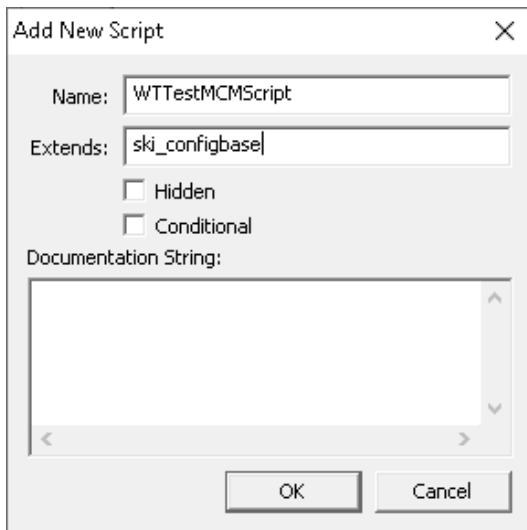


Figure 945 - New script configuration.

Right-click on the new script in the script list and select Edit Source.

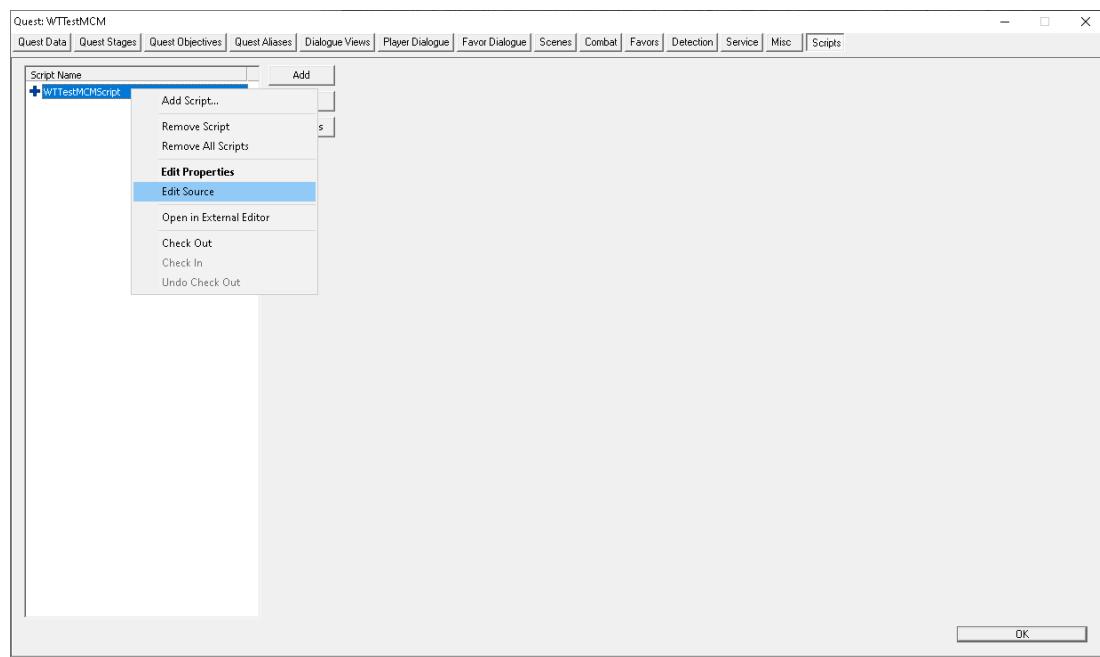


Figure 946 - Editing the script source of our new script.

Right now, our script only consists of the scriptname declaration and the script template we're extending from. `ski_configbase` is part of SKSE.



Figure 947 - Empty script.

Before we start on our script, we'll need to create some global variables. In the Object Window, go to Miscellaneous > Global.

Right-click on one of the existing global variables and select New.

Object Window				
Editor ID	Form ID	Count	User	
TAB 1stPKillCam	0010636A	0	0	
TAB AchievementSideQuestCount	000FC85C	0	0	
TAB AcolyteHealingFarmer	001075F5	0	0	
TAB AllowPlay	New	80	0	1
TAB AnimalGol	Edit	9A	0	2
TAB ArenaLarg	Duplicate	...	0	0
TAB ArenaMed	Delete	D9	0	0
TAB AtrFrgDae	B3	0	2	
TAB AtrFrgDae	Create New Object Window	B9	0	2
TAB AtrFrgRec	B2	0	2	
TAB AtrFrgRec	Use Info	3A	0	2
TAB BlackreachElevatorAltandGlobal	000F3665	0	3	
TAB BlackreachElevatorMzinchaleftGlobal	000F3663	0	3	
TAB BlackreachElevatorRaldbtharGlobal	000F3664	0	3	
TAB BladesBlessingTimer	000E6806	0	1	
TAB BookShelfGlobal	000D0568	0	59	
TAB CanReHire	00104F35	0	2	
TAB CanRehireBelrand	0010AB34	0	3	
TAB CanRehireErik	0010AB37	0	3	
TAB CanRehireJenassa	0010AB32	0	3	

Figure 948 - Creating a new global variable.

Set the ID field. The first one is going to be called WTSampleSlider1.

Leave the Variable Type set to Short and leave the Value field set to 0 then click OK.

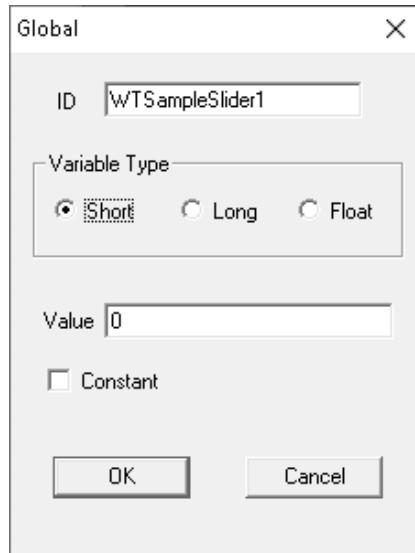


Figure 949 - Global variable setup.

Repeat these steps to create the rest of the global variables we'll need: WTSampleSlider2, WTSampleSlider3, WTSampleTickbox1, WTSampleTickbox2, WTSampleTickbox3, and WTSampleDropDown.

Object Window				
Editor ID	Form ID	Count	Users	
WTSSampleSlider1	0301566A	0 *	0	
WTSSampleSlider2	0301566B	0 *	0	
WTSSampleSlider3	0301566C	0 *	0	
WTSSampleTickbox1	0301566D	0 *	0	
WTSSampleTickbox2	0301566E	0 *	0	
WTSSampleTickbox3	0301566F	0 *	0	
WTSSampleDropDown	03015670	0 *	0	

Figure 950 - The global variables created.

Let's start adding those global variables to the script, and the option ID variables used by the interface. These variables should go directly beneath the Scriptname declaration.

```
GlobalVariable Property WTSampleSlider1 Auto
GlobalVariable Property WTSampleSlider2 Auto
GlobalVariable Property WTSampleSlider3 Auto
GlobalVariable Property WTSampleTickbox1 Auto
GlobalVariable Property WTSampleTickbox2 Auto
GlobalVariable Property WTSampleTickbox3 Auto
GlobalVariable Property WTSampleDropDownGlobal Auto

int OID_WTSampleSlider1
int OID_WTSampleSlider2
int OID_WTSampleSlider3
int OID_WTSampleTickbox1
int OID_WTSampleTickbox2
int OID_WTSampleTickbox3
int OID_WTSampleDropDownMenu
```

The screenshot shows a text editor window titled "Editing script WTTesMCMScript". The menu bar includes "File", "Edit", "View", and "Build". The script content starts with "Scriptname WTTesMCMScript extends skt_configbase". Below this, the global variable declarations are listed. In the bottom right corner of the editor window, there is a status bar with the text "Line 18/19".

```
Scriptname WTTesMCMScript extends skt_configbase

GlobalVariable Property WTSampleSlider1 Auto
GlobalVariable Property WTSampleSlider2 Auto
GlobalVariable Property WTSampleSlider3 Auto
GlobalVariable Property WTSampleTickbox1 Auto
GlobalVariable Property WTSampleTickbox2 Auto
GlobalVariable Property WTSampleTickbox3 Auto
GlobalVariable Property WTSampleDropDownGlobal Auto

int OID_WTSampleSlider1
int OID_WTSampleSlider2
int OID_WTSampleSlider3
int OID_WTSampleTickbox1
int OID_WTSampleTickbox2
int OID_WTSampleTickbox3
int OID_WTSampleDropDownMenu
```

Figure 951 - Adding in global variables.

So basically our General submenu will have three tickboxes, three sliders and one drop-down menu.

Press CTRL + S to compile the script.



```

Editing script WTTTestMCMScript
File Edit View Build

GlobalVariable Property WTSampleSlider1 Auto
GlobalVariable Property WTSampleSlider2 Auto
GlobalVariable Property WTSampleSlider3 Auto
GlobalVariable Property WTSampleSlider1 Auto
GlobalVariable Property WTSampleSlider2 Auto
GlobalVariable Property WTSampleSlider3 Auto
GlobalVariable Property WTSampleDropDownGlobal Auto

int OID_WTSampleSlider1
int OID_WTSampleSlider2
int OID_WTSampleSlider3
int OID_WTSampleSlider1
int OID_WTSampleSlider2
int OID_WTSampleSlider3
int OID_WTSampleDropDownGlobal

Compiler Output:
Starting 1 compile threads for 1 files...
Compiling "WTTTestMCMScript"...
Starting assembly of WTTTestMCMScript
0 error(s), 0 warning(s)
Assembly succeeded
Compilation succeeded.

Save Succeeded
Line 19\19

```

Figure 952 - Compiling the script.

Back in the Scripts tab, click on WTTTestMCMScript to highlight it then click on the Properties button.

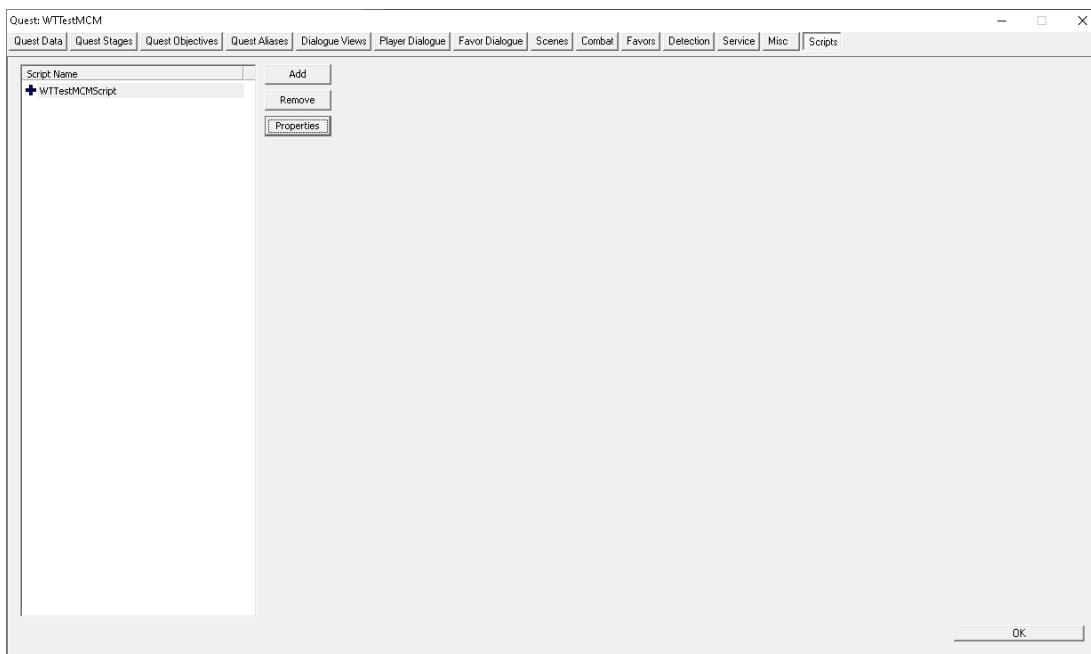


Figure 953 - Linking the script properties to the global variables.

Click on the Auto-Fill All button. This should automatically link our script properties with the global variables we just created.

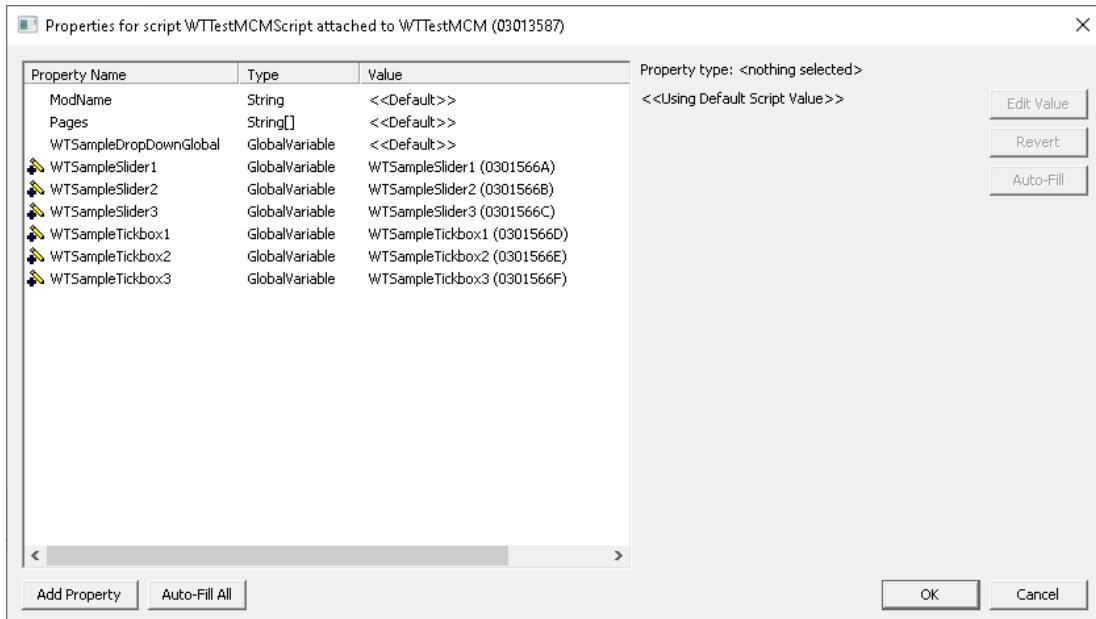


Figure 954 - Auto-Fill All.

The properties that have an identical name to their global variable ID will be linked automatically. The ones that have mismatching names can be linked manually. Set WTSSampleDropDownGlobal to WTSSampleDropDown.

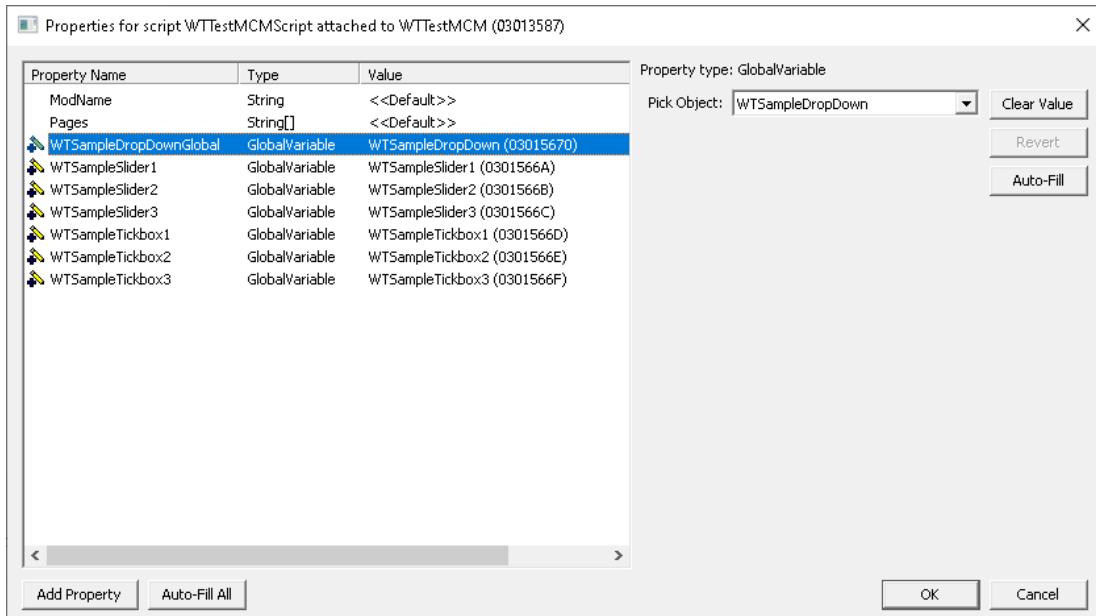


Figure 955 - Setting the global variable for WTSSampleDropDownGlobal.

Next, click on ModName in the properties list and click the Edit Value button to the right. Enter in the name of the mod as you want it to appear in the MCM menu.

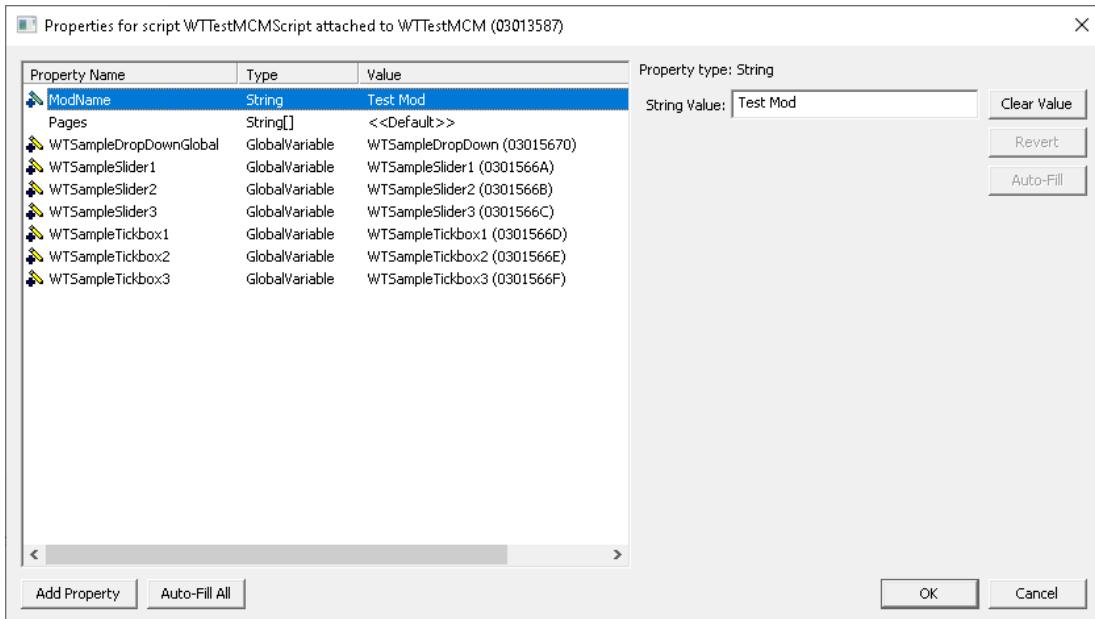


Figure 956 - Setting the mod name for the MCM.

Click on the Pages property and click on the Edit Value button.

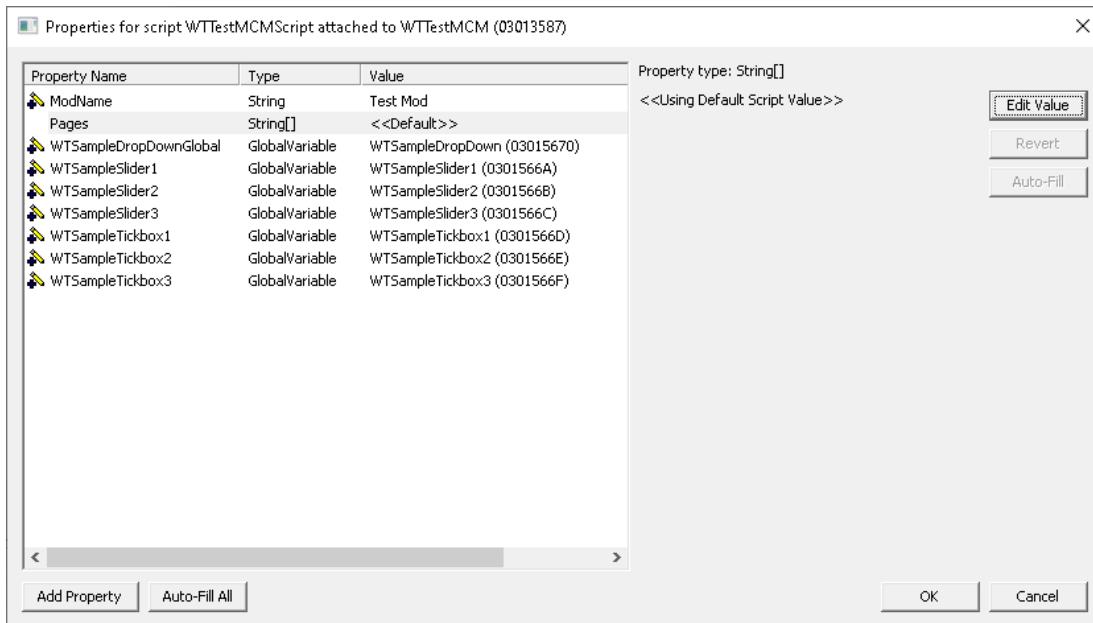


Figure 957 - Adding a submenu to our MCM.

Click on the Add button.

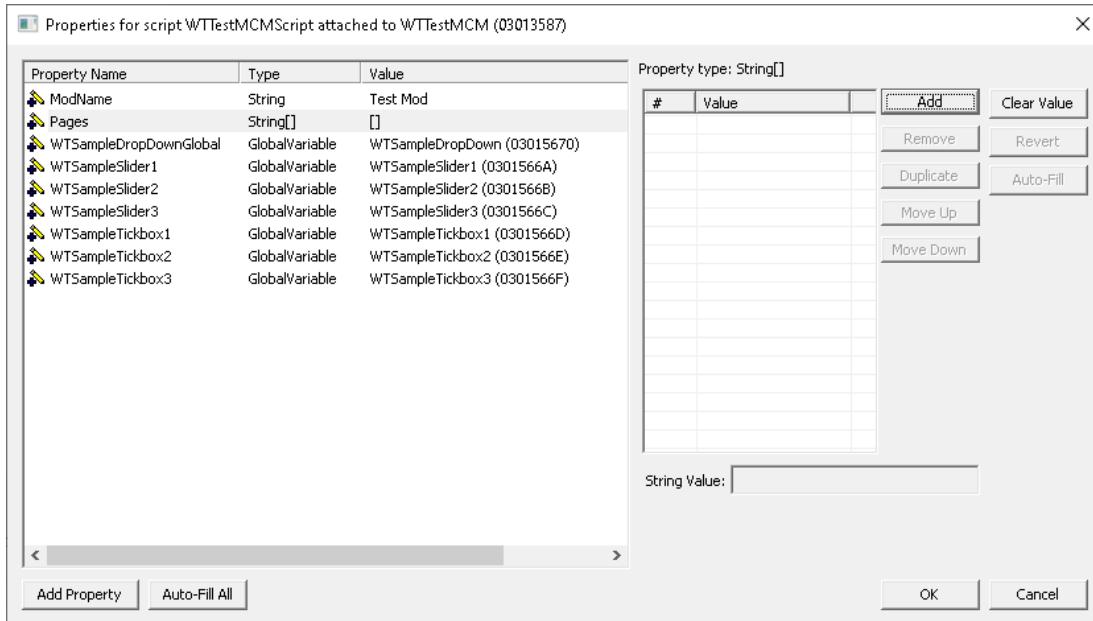


Figure 958 - Adding a new page.

For this example I'm going to add one submenu. Set the string value for the new page to General.

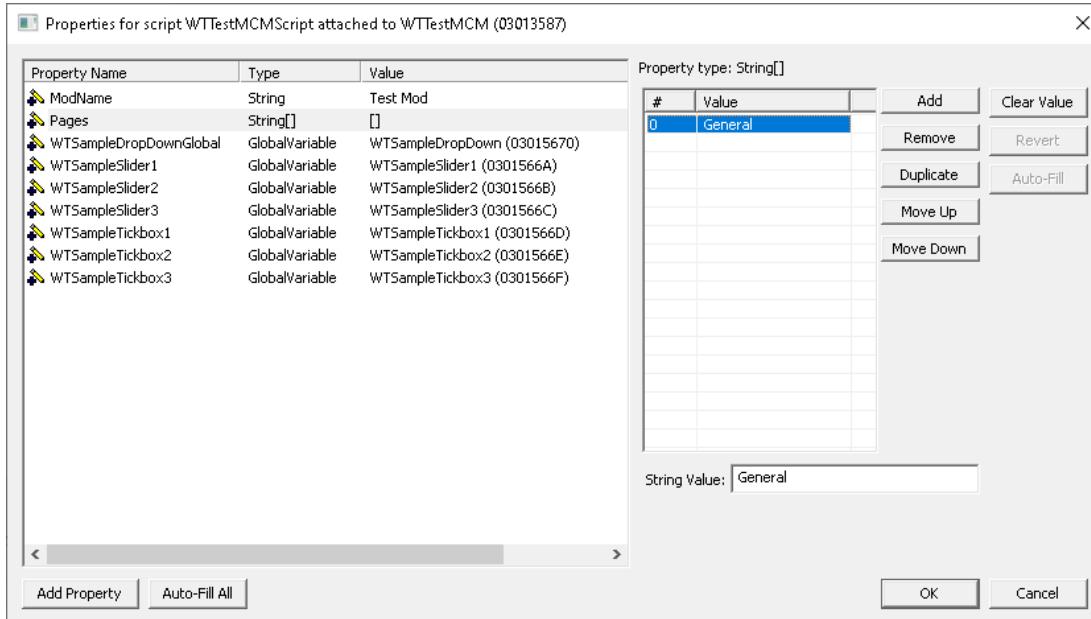


Figure 959 - Adding a General page.

Click OK to close out of script properties. Let's go back to the script source.

The first thing we need to do is initialize the script with the following commands:

```
event OnConfigInit()
parent.OnInit()
endEvent
```

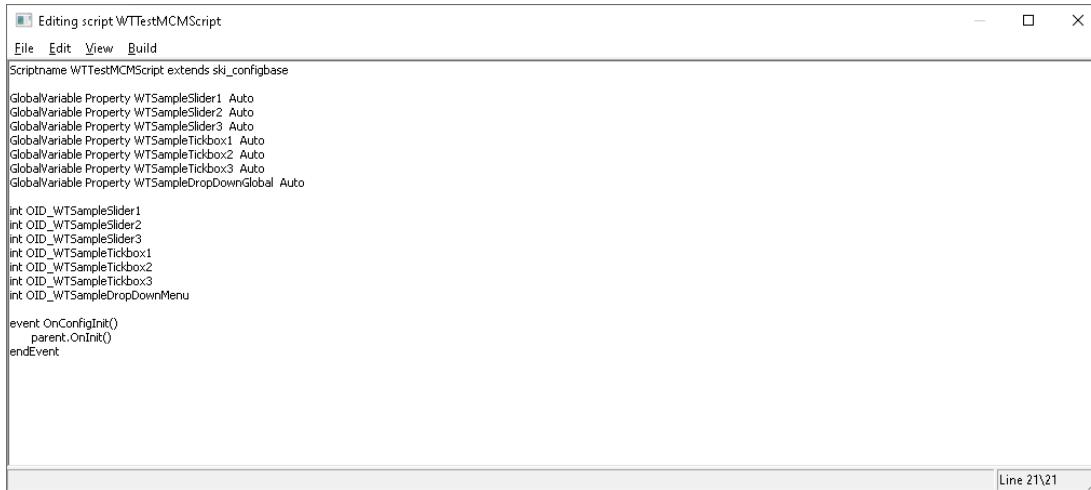


Figure 960 - Initializing the MCM script.

Most MCMs include a splash screen which is displayed when the user clicks on the mod in the MCM menu. The first page loaded by the MCM doesn't have a page name, hence “”.

```
Event OnPageReset(string page)
if page == ""
self.LoadCustomContent("testmod/testmodsplash.dds", 0.000000,
0.000000)
return
else
self.UnloadCustomContent()
endif
EndEvent
```



Figure 961 - Adding a splash screen.

The path defined in the LoadCustomContent command should exist in the Skyrim\Data\Interface or Skyrim Special Edition\Data\Interface folder.



Figure 962 - MCM splash screen for Wyrmsooth.

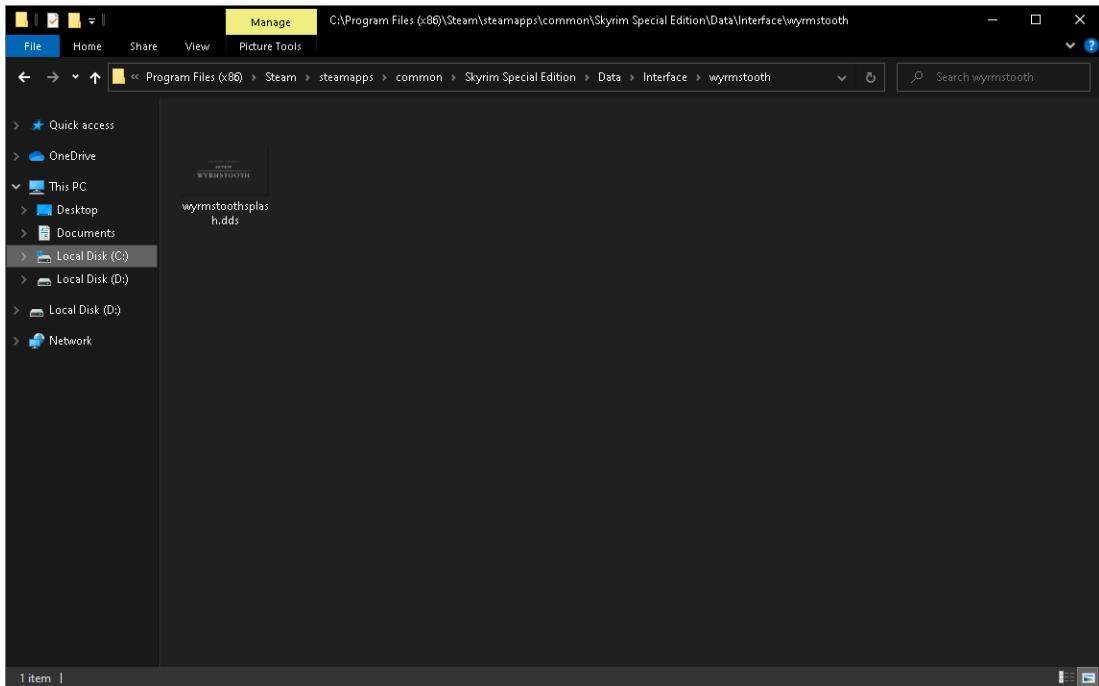


Figure 963 - Full path to splash screen .dds file.

For Wyrmsooth, the path used by LoadCustomContent for its splash screen would be:

```
self.LoadCustomContent("wyrmsooth/wyrmsoothsplash.dds", 0.000000, 0.000000)
```

The numbers after the path to the .dds file specify an X/Y offset for the image. We can leave them both set to 0.000000 for now.

Next, let's add a General tab with some sample sliders, tick-boxes and a drop-down selection.

```

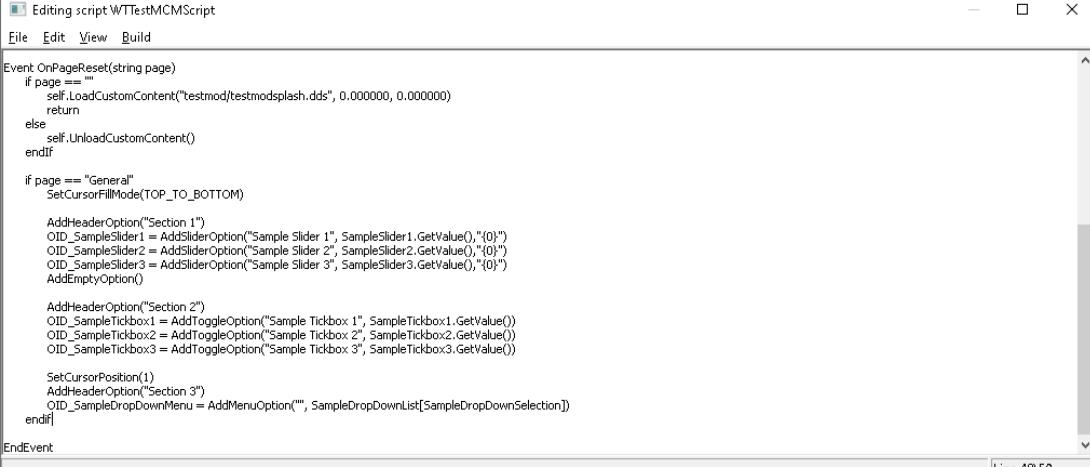
if page == "General"
SetCursorFillMode(TOP_TO_BOTTOM)

AddHeaderOption("Section 1")
OID_WTSampleSlider1 = AddSliderOption("Sample Slider 1",
WTSampleSlider1.GetValue(),"{0}")
OID_WTSampleSlider2 = AddSliderOption("Sample Slider 2",
WTSampleSlider2.GetValue(),"{0}")
OID_WTSampleSlider3 = AddSliderOption("Sample Slider 3",
WTSampleSlider3.GetValue(),"{0}")
AddEmptyOption()

AddHeaderOption("Section 2")
OID_WTSampleTickbox1 = AddToggleOption("Sample Tickbox 1",
WTSampleTickbox1.GetValue())
OID_WTSampleTickbox2 = AddToggleOption("Sample Tickbox 2",
WTSampleTickbox2.GetValue())
OID_WTSampleTickbox3 = AddToggleOption("Sample Tickbox 3",
WTSampleTickbox3.GetValue())

SetCursorPosition(1)
AddHeaderOption("Section 3")
OID_WTSampleDropDownMenu = AddMenuOption("",,
WTSampleDropDownList[WTSampleDropDownSelection])
Endif

```



The screenshot shows the MCM Editor window with the title 'Editing script WTTTestMCMScript'. The menu bar includes File, Edit, View, and Build. The main area contains the script code for the 'General' tab. The code is identical to the one provided in the text block above, including the if/else logic for different tabs and the addition of UI elements like sliders and checkboxes. The status bar at the bottom right indicates 'Line 48/50'.

```

Event:OnPageReset(string page)
if page == ""
    self.LoadCustomContent("testmod/testmodsplash.dds", 0.000000, 0.000000)
else
    self.UnloadCustomContent()
endif

if page == "General"
    SetCursorFillMode(TOP_TO_BOTTOM)

    AddHeaderOption("Section 1")
    OID_SampleSlider1 = AddSliderOption("Sample Slider 1", SampleSlider1.GetValue(),"{0}")
    OID_SampleSlider2 = AddSliderOption("Sample Slider 2", SampleSlider2.GetValue(),"{0}")
    OID_SampleSlider3 = AddSliderOption("Sample Slider 3", SampleSlider3.GetValue(),"{0}")
    AddEmptyOption()

    AddHeaderOption("Section 2")
    OID_SampleTickbox1 = AddToggleOption("Sample Tickbox 1", SampleTickbox1.GetValue())
    OID_SampleTickbox2 = AddToggleOption("Sample Tickbox 2", SampleTickbox2.GetValue())
    OID_SampleTickbox3 = AddToggleOption("Sample Tickbox 3", SampleTickbox3.GetValue())

    SetCursorPosition(1)
    AddHeaderOption("Section 3")
    OID_SampleDropDownMenu = AddMenuOption("", SampleDropDownList[SampleDropDownSelection])
endif

EndEvent

```

Figure 964 - General tab scripting added.

If you added more submenus in the script properties, you can add UI elements to them by adding elseif page == "My Second Submenu", elseif page == "My Third Submenu" before the Endif.

If the player clicks on the General tab, we want these interface elements to appear instead of the splash screen. We can separate different pages of UI elements using an if/elseif/else statement.

SetCursorFillMode(TOP_TO_BOTTOM) means the interface elements we're adding to the UI will be added vertically starting from the top-left. The alternative arrangement is LEFT_TO_RIGHT which would mean the interface elements would be added to the UI horizontally starting from the top-left. I set it to TOP_TO_BOTTOM so we can define exactly at which element we start the second column.

AddHeaderOption("Section 1") adds a header to the UI. We can use headers to group similar interface elements together.

Right now if we try and compile we'll get an error because we're missing a couple variables needed for the drop-down menu. So let's add those in next.

```

Editing script WTTestMCMScript
File Edit View Build
SetCursorFillMode(TOP_TO_BOTTOM)

AddHeaderOption("Section 1")
OID_WTSampleSlider1 = AddSliderOption("Sample Slider 1", WTSampleSlider1.GetValue(),"0")
OID_WTSampleSlider2 = AddSliderOption("Sample Slider 2", WTSampleSlider2.GetValue(),"0")
OID_WTSampleSlider3 = AddSliderOption("Sample Slider 3", WTSampleSlider3.GetValue(),"0")
AddEmptyOption()

AddHeaderOption("Section 2")
OID_WTSampleTickbox1 = AddToggleOption("Sample Tickbox 1", WTSampleTickbox1.GetValue())
OID_WTSampleTickbox2 = AddToggleOption("Sample Tickbox 2", WTSampleTickbox2.GetValue())
OID_WTSampleTickbox3 = AddToggleOption("Sample Tickbox 3", WTSampleTickbox3.GetValue())

SetCursorPosition(1)
AddHeaderOption("Section 3")
OID_WTSampleDropDownMenu = AddMenuOption("", WTSampleDropDownList[WTSampleDropDownSelection])
endif

EndEvent
Compiler Output:
Starting 1 compile threads for 1 files...
Compiling "WTTestMCMScript"...
C:\Program Files (x86)\Steam\steamapps\common\skyrim\DATA\Scripts\Source\temp\WTTestMCMScript.psc(47,47): variable WTSampleDropDownList is undefined
C:\Program Files (x86)\Steam\steamapps\common\skyrim\DATA\Scripts\Source\temp\WTTestMCMScript.psc(47,68): variable WTSampleDropDownSelection is undefined
C:\Program Files (x86)\Steam\steamapps\common\skyrim\DATA\Scripts\Source\temp\WTTestMCMScript.psc(47,47): arrays can only be indexed by integers
C:\Program Files (x86)\Steam\steamapps\common\skyrim\DATA\Scripts\Source\temp\WTTestMCMScript.psc(47,47): only arrays can be indexed
No output generated for WTTestMCMScript, compilation failed.

Save Succeeded
Line 30/50

```

Figure 965 - Compilation error.

Enter the following variable declarations beneath the int OID variables:

```

string[] WTSampleDropDownList
int WTSampleDropDownSelection

```

```

Editing script WTTestMCMScript
File Edit View Build
GlobalVariable Property WTSampleTickbox3 Auto
GlobalVariable Property WTSampleDropDownGlobal Auto

int OID_WTSampleSlider1
int OID_WTSampleSlider2
int OID_WTSampleSlider3
int OID_WTSampleTickbox1
int OID_WTSampleTickbox2
int OID_WTSampleTickbox3
int OID_WTSampleDropDownMenu

string[] WTSampleDropDownList
int WTSampleDropDownSelection

event OnConfigInit()
    parent.OnInit()
endEvent

Event onPageReset(string page)
Compiler Output:
Starting 1 compile threads for 1 files...
Compiling "WTTestMCMScript"...
Starting assembly of WTTestMCMScript
0 error(s), 0 warning(s)
Assembly succeeded

Compilation succeeded.

Save Succeeded
Line 19/53

```

Figure 966 - Adding the string and int variables for the drop-down list.

If you compile the script it should now compile successfully.

Next, let's add in an array with some selectable options for the drop-down menu. These options will appear when the player clicks on the drop-down menu in the MCM.

In the OnConfigInit() event, add the following lines beneath parent.OnInit():

```
WTSSampleDropDownList = new string[15]
WTSSampleDropDownList[0] = "Option 1"
WTSSampleDropDownList[1] = "Option 2"
WTSSampleDropDownList[2] = "Option 3"
WTSSampleDropDownList[3] = "Option 4"
WTSSampleDropDownList[4] = "Option 5"
```



The screenshot shows the WTS Editor interface with the title bar 'Editing script WTTestMCMScript'. The menu bar includes File, Edit, View, and Build. The main code area contains C# script code. The code defines a string array `WTSSampleDropDownList` with five elements, each containing a string option. It also includes an `OnConfigInit()` event that calls `parent.OnInit()`. Below the code is a 'Compiler Output:' section showing the compilation process and success message. At the bottom, a status bar indicates 'Save Succeeded' and 'Line 24\59'.

```
WTSSampleDropDownList = new string[15]
WTSSampleDropDownList[0] = "Option 1"
WTSSampleDropDownList[1] = "Option 2"
WTSSampleDropDownList[2] = "Option 3"
WTSSampleDropDownList[3] = "Option 4"
WTSSampleDropDownList[4] = "Option 5"

event OnConfigInit()
    parent.OnInit()

Event onPageReset(string page)
    if page ===
        self.LoadCustomContent("testmod/testmod\dslash.dds", 0.000000, 0.000000)

Compiler Output:
Starting 1 compile threads for 1 files...
Compiling "WTTestMCMScript"...
Starting assembly of WTTestMCMScript
0 error(s), 0 warning(s)
Assembly succeeded
Compilation succeeded.

Save Succeeded
Line 24\59
```

Figure 967 - Adding drop-down list options.

To show the options in a drop-down list when the player clicks on it, we'll need to add an OnOptionMenuOpen() event.

```
Event OnOptionMenuOpen(int option)
if (option == OID_WTSampleDropDownMenu)
SetMenuDialogOptions(WTSampleDropDownList)
SetMenuDialogStartIndex(WTSampleDropDownSelection)
SetMenuDialogDefaultIndex(0)
endif
EndEvent
```



Figure 968 - Event to handle opening a drop-down menu.

The value defined in SetMenuDialogueDefaultIndex() specifies the default value for this drop-down menu. For this example I'm just going to leave it set to '0'.

We need to add an OnOptionMenuAccept() event to handle the player making a selection from a drop-down menu.

```
Event OnOptionMenuAccept(int option, int index)
if (option == OID_WTSampleDropDownMenu)
WTSampleDropDownSelection = index
SetMenuOptionValue(OID_WTSampleDropDownMenu,
WTSampleDropDownList[WTSampleDropDownSelection])
WTSampleDropDownGlobal.SetValue(WTSampleDropDownSelection)
endif
EndEvent
```

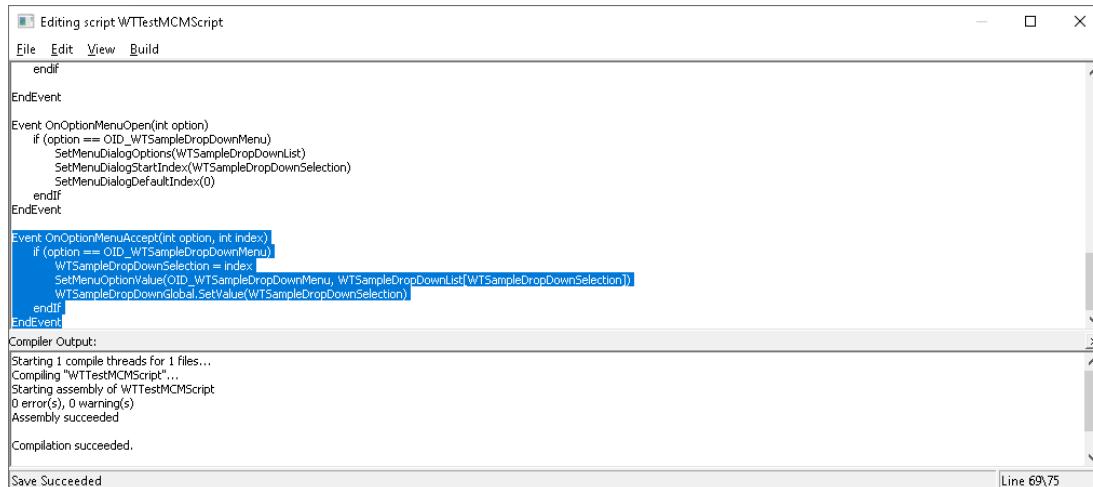


Figure 969 - Event to handle making a selection from a drop-down menu.

The option we choose here will change the value of the WTSampleDropDownGlobal global variable which we can reference in other scripts.

Like drop-down menus, we need to set up two events sliders. The OnOptionSliderOpen() event handles the opening of the slider menu when the player clicks on it in the MCM.

```
Event OnOptionSliderOpen(Int Option)
if(Option == OID_WTSampleSlider1)
SetSliderDialogStartValue(WTSampleSlider1.GetValue())
SetSliderDialogDefaultValue(0)
SetSliderDialogRange(0, 99)
SetSliderDialogInterval(1)
elseif(Option == OID_WTSampleSlider2)
SetSliderDialogStartValue(WTSampleSlider2.GetValue())
SetSliderDialogDefaultValue(0)
SetSliderDialogRange(0, 99)
SetSliderDialogInterval(1)
elseif(Option == OID_WTSampleSlider3)
SetSliderDialogStartValue(WTSampleSlider3.GetValue())
SetSliderDialogDefaultValue(0)
SetSliderDialogRange(0, 99)
SetSliderDialogInterval(1)
endif
EndEvent
```

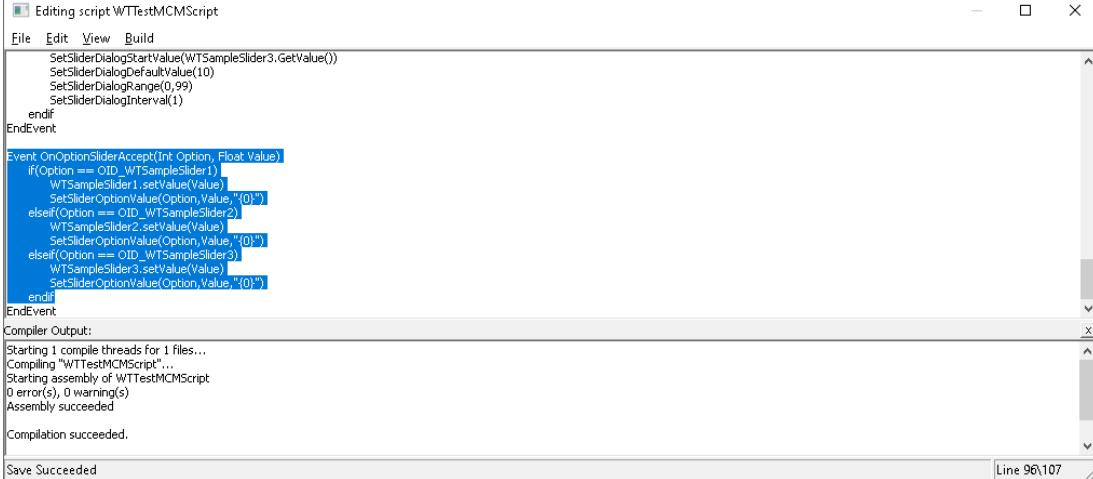


Figure 970 - Event to handle opening a slider.

SetSliderDialogueStartValue() sets the initial value of the slider. In the example above that value is read from the global variable. SetSliderDialogDefaultValue() sets the default value of the slider. SetSliderDialogRange() sets the minimum and maximum values for the slider. SetSliderDialogInterval() sets the amount by which the player can increase or decrease the slider value. In the example above I set this to '1' so the player can increase or decrease the slider value by a full integer.

To save the slider value set by the player we need to use the OnOptionSliderAccept() event.

```
Event OnOptionSliderAccept(Int Option, Float Value)
if(Option == OID_WTSampleSlider1)
    WTSampleSlider1.setValue(Value)
    SetSliderOptionValue(Option,Value,"{0}")
elseif(Option == OID_WTSampleSlider2)
    WTSampleSlider2.setValue(Value)
    SetSliderOptionValue(Option,Value,"{0}")
elseif(Option == OID_WTSampleSlider3)
    WTSampleSlider3.setValue(Value)
    SetSliderOptionValue(Option,Value,"{0}")
endif
EndEvent
```



The screenshot shows the MCM Editor interface with the following details:

- Title Bar:** Editing script WTTTestMCMScript
- Menu Bar:** File Edit View Build
- Script Area:**

```
SetSliderDialogStartValue(WTSampleSlider3.GetValue())
SetSliderDialogDefaultValue(10)
SetSliderDialogRange(0,99)
SetSliderDialogInterval(1)
endif
EndEvent

Event OnOptionSliderAccept(Int Option, Float Value)
if(Option == OID_WTSampleSlider1)
    WTSampleSlider1.setValue(Value)
    SetSliderOptionValue(Option,Value,"{0}")
elseif(Option == OID_WTSampleSlider2)
    WTSampleSlider2.setValue(Value)
    SetSliderOptionValue(Option,Value,"{0}")
elseif(Option == OID_WTSampleSlider3)
    WTSampleSlider3.setValue(Value)
    SetSliderOptionValue(Option,Value,"{0}")
endif
EndEvent
```
- Compiler Output:**

```
Starting 1 compile threads for 1 files...
Compiling "WTTTestMCMScript".
Starting assembly of WTTTestMCMScript
0 error(s), 0 warning(s)
Assembly succeeded
```
- Status Bar:** Save Succeeded Line 96/107

Figure 971 - Event to handle setting a new value based on the slider selection.

We only need the OnOptionSelect() event to handle tickboxes.

```

event OnOptionSelect(int Option)
    if (Option == OID_WTSampleTickbox1)
        if WTSampleTickbox1.getValue() as int == 0
            WTSampleTickbox1.setValue(1)
            SetToggleOptionValue(Option, 1)
        else
            WTSampleTickbox1.setValue(0)
            SetToggleOptionValue(Option, 0)
        endif

    elseif (Option == OID_WTSampleTickbox2)
        if WTSampleTickbox2.getValue() as int == 0
            WTSampleTickbox2.setValue(1)
            SetToggleOptionValue(Option, 1)
        else
            WTSampleTickbox2.setValue(0)
            SetToggleOptionValue(Option, 0)
        endif

    elseif (Option == OID_WTSampleTickbox3)
        if WTSampleTickbox3.getValue() as int == 0
            WTSampleTickbox3.setValue(1)
            SetToggleOptionValue(Option, 1)
        else
            WTSampleTickbox3.setValue(0)
            SetToggleOptionValue(Option, 0)
        endif
    endif
EndEvent

```



Figure 972 - Event to handle ticking or unticking a tickbox.

Basically, if the player ticks a tickbox and the tickbox isn't ticked, we flag it as ticked and set its global variable value to 1. Otherwise, untick it and set its global variable value to 0. Use tickboxes for setting Boolean values. i.e.: when an option either needs to be on or off/yes or no.

Lastly, we need to add an OnoptionHighlight() event to handle text tips that appear at the bottom of the MCM screen when we hover over UI elements.

```
Event OnOptionHighlight(Int Option)
If (Option == OID_WTSampleSlider1)
SetInfoText("Sample slider 1 highlighted.")
ElseIf (Option == OID_WTSampleSlider2)
SetInfoText("Sample slider 2 highlighted.")
ElseIf (Option == OID_WTSampleSlider3)
SetInfoText("Sample slider 3 highlighted.")
ElseIf (Option == OID_WTSampleTickbox1)
SetInfoText("Sample tickbox 1 highlighted.")
ElseIf (Option == OID_WTSampleTickbox2)
SetInfoText("Sample tickbox 2 highlighted.")
ElseIf (Option == OID_WTSampleTickbox3)
SetInfoText("Sample tickbox 3 highlighted.")
ElseIf (Option == OID_WTSampleDropDownMenu)
SetInfoText("Sample drop down menu highlighted.")
EndIf
EndEvent
```

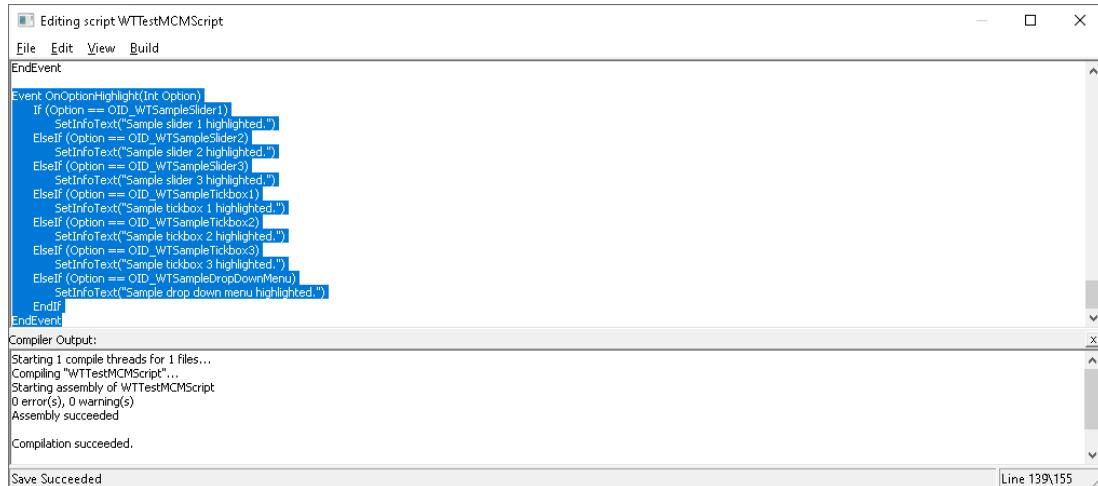


Figure 973 - Event to handle text tips.

The text to display at the bottom of the screen is defined by SetInfoText().

If we go in-game, our new MCM should register itself after a few seconds.



Figure 974 - Test Mod MCM.

Take a few moments to confirm all the options are present and the text tips for each UI element are correct.

Change a few settings then close and re-open the MCM to confirm that the settings were successfully saved.



Figure 975 - Options saved after closing the menu.

For more information I'd recommend Schlangster's [MCM Quickstart](#) guide on Github.