Page contents

- Plan the conversation before opening the editor
- Bind the speaker through a quest alias
- Add a greeting and a choice phase
- Optional: more than one reply to a choice
- Inspect the saved records and run the test
- Sources and limits

Offline copy of Dialogue scene and player choices .

Fallout 4: build and test a first dialogue scene

Documentation status: In progress. **Sources checked:** 2026-11-11 against two Fallout 4 Creation Kit tutorials by Seddon4494 and the Fallout 4 Papyrus Scene/TopicInfo API index. **Technical validation:** the author demonstrates parts of this workflow in a 2016 game build; ModDocs has not made the example plugin or repeated the test on a current game and Creation Kit build. Editor labels and voice behavior may differ. Record your versions and inspect a current vanilla scene before relying on an older click path.

This chapter creates a small conversation with one NPC: a greeting, player choices, an optional second phase, and a deliberate exit. It follows Basic Dialogue and Longer Conversation and Random Responses, which are direct tutorials by the person demonstrating the editor. The wording and example structure here are new. It does not require a dialogue-extension mod. The quest stages and aliases chapter covers the separate problem of quest progression.

Plan the conversation before opening the editor

Make a new plugin in an isolated test profile and preserve a pre-test save. Prepare one **unique NPC** already placed where the player can reach it. Use a distinctive editor ID, voice type, quest, alias, scene and topic names so you can inspect them again after

saving. In the tutorial, a unique voice type helps separate the NPC from generic lines. Do not remove an NPC's factions as a general fix: factions can control other behavior. If you are editing an existing NPC, compare its current voice and faction data before changing either.

Use an explicit flow map. The following is a **design example**, not a captured plugin or tested Fallout 4 scene:

Step	Player sees or does	NPC response	Next state
Greeting	Activates the test NPC	Brief introduction	Choice phase
Choice A	Asks about the location	Gives one answer	Choice phase again
Choice B	Asks for a task	Gives a short request	Second phase
Choice C	Says goodbye	Acknowledges departure	End running scene
Second phase	Accepts or declines	Confirms the choice	End running scene

Write the *prompt*, the player's actual response text, and the NPC's response separately for each choice. In Fallout 4 the short choice prompt and the line spoken by the voiced protagonist are distinct editor fields in the tutorial. If you have not authored or licensed voice files, expect silence for new lines; an on-screen prompt is not evidence that spoken audio exists. Plan an exit from every reachable phase. A loop is useful for repeated questions only when another choice can leave it.

Bind the speaker through a quest alias

In the Creation Kit, load the required masters and set **your plugin** as active. The tutorial opens **Character > Voice Type**, creates a unique voice type and assigns it to the new NPC. Save the plugin, then create a new quest under **Character > Quest**. The dialogue quest is the owner of the alias and scene; it need not be presented as a journal quest simply because it holds dialogue. Reopen the quest after its first save if the editor has not exposed all tabs yet, as the video demonstrates.

On the quest's **Aliases** tab, create a **Reference Alias**, give it an identifiable name, and fill it from the unique actor. Verify the chosen actor rather than accepting the first

similar name in the picker. The alias is the role the scene uses for its speaker; an empty or wrong alias can make a carefully authored scene appear to do nothing or use the wrong actor. Reopen the saved quest and inspect the alias fill before working on responses.

Create a new scene on the quest's **Scenes** tab and mark it as **Player Dialogue**. Add the NPC alias as the scene actor. Give the scene and each phase a distinctive name. The tutorial's editor sometimes did not show a newly named phase until the author closed and reopened its dialog. If a phase is absent from a later phase picker, save and reopen before creating duplicate phases.

Add a greeting and a choice phase

Create a **Start Greeting** topic for the actor and add a short NPC response. Do not assume that choosing the actor alias on the scene automatically confines every greeting response. In the basic tutorial, the first test let an unintended actor say the greeting; the author returned to the greeting's topic info and assigned its **Forced Alias** to the intended NPC alias. Reinspect the saved greeting topic info and test with the intended NPC *and* a nearby unrelated NPC. The second test is the one that catches a global greeting leak.

Add a phase with a **Player Dialogue** action. Enter each player's choice prompt and response text, then the NPC response in the corresponding response area. The basic lesson uses the positive, negative, neutral and question slots to demonstrate Fallout 4's choice layout. Those slots are a UI arrangement, not proof that every conversation must have four meaningful branches. Use only branches you can write and test; keep the prompt truthful about the longer response text.

For Choice A in the example map, direct its response back to the choice phase with a **Start Scene** action targeting the same scene and named phase. For Choice B, add a second phase and target that phase. In the follow-up tutorial, an action was used to advance to another phase and a selected response was routed back to a named phase. After editing each destination, close and reopen the scene, then confirm that the action still names the intended scene and phase. A visual arrow or an action label alone is insufficient if it points to a stale name.

For Choice C and every terminal branch in the second phase, set **End Running Scene** on the final response. The first tutorial's in-game test showed the choices working but trapped the player in a loop because no exit had been authored. The author then added

an end flag but did not demonstrate a second in-game run for that correction. Treat the exit as a required test, not as proven by the video's first run.

Optional: more than one reply to a choice

The follow-up lesson adds multiple response entries under one NPC answer and marks them as random, ending the random group at its final entry. If you use this, give **every** possible reply the same intended next state or an intentionally different state. The author initially forgot to return each random reply to the choice phase and corrected that before the game test. A branch that works with one random line but ends unexpectedly with another is an incomplete scene.

Random responses should be tested as a set. Re-enter the conversation until each reply has appeared, record the reply and destination, and confirm that the player can still exit. The tutorial's later game segment demonstrates more than one random response, a longer second phase and an ending. It does not establish current-build behavior for this site's own plugin.

Inspect the saved records and run the test

Save and close the Creation Kit, reopen the plugin as active, then inspect the quest, alias, scene, topic infos, greeting restriction, response actions and end flags. In FO4Edit, confirm the plugin has only expected masters and records; check for accidental NPC or quest overrides. Do not infer that a clean xEdit record tree proves the dialogue will play.

Run a fresh disposable save with only required mods active. Talk to the intended NPC, then to a different NPC nearby. For each choice, record the exact visible prompt, actual subtitle or audio behavior, NPC reply, next phase and whether the player can leave dialogue. Repeat on a reload. Check the loop path twice, both choices in the second phase, and every random reply you authored. A missing greeting can come from the wrong alias, conditions, quest startup or voice setup; a greeting spoken by another NPC points first to topic-info restrictions. A choice that never exits points to a missing end flag or a loop target. Change one condition or action at a time and keep the failing test save.

The Fallout 4 Scene API lists `OnBegin`, `OnEnd`, `OnPhaseBegin` and `OnPhaseEnd`; the TopicInfo API lists `OnBegin` and `OnEnd`. Those are possible diagnostics for a later

scripted scene. This chapter does not add a Papyrus script or claim that those events fired in this example. A quest dialogue release still needs a tested start condition, voice-asset and subtitle review, stage/objective integration, localization plan, and save-state tests.

Sources and limits

- Fallout 4 Creation Kit Tutorial — Basic Dialogue, Seddon4494, published 2016-04-28. Its demonstrated game run covers the initial greeting and choices; the exit correction was described after that run.
- Fallout 4 Creation Kit Tutorial — Longer Conversation and Random Responses, Seddon4494, published 2016-04-29. Its game run covers a longer conversation, alternate replies and an ending. These videos were reviewed through auto-generated English captions; exact spoken words may have transcription errors.
- Fallout 4 Scene API and TopicInfo API, community-maintained Papyrus Index. Used only for the named event interfaces, not to claim a tested scene script.