

# Important Information

## Support and Full Documentation

- Full documentation and examples can be found at [our website](#)
- Support for this asset can be found at [our Github](#)
- Contact us at [support@quantumtekhub.com](mailto:support@quantumtekhub.com) for problems with the asset

## Installation

### 1. Package Requirements

Before installation of Quantum Dialogue, you need to have the Text Mesh Pro package from the Package Manager (Under Window > Package Manager) installed. After installation of Text Mesh Pro, you need to import the files necessary for Text Mesh Pro from Window > Text Mesh Pro > Import TMP Essential Resources.

**However, you don't need Text Mesh Pro if you don't import the demo folder.**

### 2. Install

After you meet the package requirements you can install Quantum Dialogue.

### 3. Demo Scenes

There is one demo scene: the Dialogue Demo, located under the Demo folder. The Dialogue Demo showcases the dialogue system, with messages and choices.

## Getting Started

### Basics

Quantum Dialogue is a dialogue system, which allows for conversations with speakers (NPCs or the player), different message texts, and player choices. It uses a node editor to make it easier to create a dialogue tree.

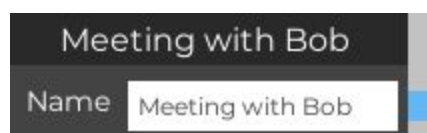
---

Dialogues are just an over arching thing, made of many conversations. Each conversation has a starting message and likely many speakers. Player choices are also parts of conversations. Speakers are NPCs or the player, that have a name and optional icon. Messages are just text spoken by someone, and can even have audio clips. Player choices are a list of choices that lead to different messages. At most one choice node can have 6 options.

To create a dialogue, go to Create > Quantum Tek > Quantum Dialogue > Dialogue. To edit a dialogue, first Create > Quantum Tek > Quantum Dialogue > Node Database. This file should be saved under an Editor folder, as it is only needed in the Unity editor. Drag the dialogue into the Data DB field of the node database. Select the node database, then go to Window > Quantum Dialogue > Dialogue Editor.

Conversations have a starting message that can lead to another message or a player choice. Each message can lead to one message or one choice node. Choice nodes can only lead to message nodes. Below is an explanation of each node.

### Conversation Node



Conversations only need a conversation name (“Meeting with Bob”), and a first message (the output knob on the right). This output knob can only connect to the second input knob (bottom left) on a message node.

## Speaker Node



Speakers only need a name (“Bob”), but can also have an icon. Their output knob can only connect to the first input (top left) of a message node.

## Message Node



Messages have a message text, and can have an audio clip (which would have audio programmed in by the asset user. There is no pre-programmed functionality with it). The first input knob connects to the output knob of a speaker, or the first output knob of a message. Connect the first output knob of a message to the first input knob of another message means the two messages have the same speaker, so the first message in the chain will connect to a speaker node. The second input knob would be

connected to the output knob of a conversation, or the second output knob of a message. The second output knob of a message can also connect to nothing, signifying the end of a conversation.

## Choice Node



Choice nodes just require a list of up to 6 choices. Add a choice with the (+) button at the bottom. Remove the last choice using the (-) button next to the last choice. The input knob connects to the second output knob of a message. There is one output knob per choice, meaning each output knob goes to the second input knob of a message.

## Using the Editor

Drag nodes by clicking and dragging the mouse. Pan the editor using the bottom and right scrollbars. Right click outside of a node to add a new node. Right click on a node to detach all connected nodes, or delete the node. Right click on a knob (the blue squares on either side of the node) to attach a node, or detach nodes connected to that knob. When attaching nodes together, the first selected node will have a connector between the node's knob and the mouse. Left clicking on a knob on another node will connect the two, as long as it is an allowed connection, as highlighted in the paragraph below.

## Connection Rules

Conversation - Inputs: None. Outputs: First - Second on Message.

Speaker - Inputs: None. Outputs: First - First on Message.

Message - Inputs: First - First on Speaker, First on Message, X on Choice. Outputs: First - First on Message, Second - Second on Message, First on Choice.

Choice - Inputs: Second on Message. Outputs: X - Second on Message.

---

Dialogue handlers require a dialogue object to work properly. Additionally, they have a current conversation index, current conversation, and current message info. The current conversation can be set using the SetConversation(name) function. Input the name of the conversation, such as “Meet with Bob”, in the example. Current message info gives the current message ID, next message ID, and message type (either Message or Choice). The current message ID refers to the ID of the current message or choice node in the current conversation. The next message ID refers to the ID of the next message or choice node, or -1 if there is none.

The current message or choice node can be fetched with either the GetMessage() or GetChoice() functions. You can determine which function is needed based on currentMessageInfo.Type. NextMessage(choice) is used to go to the next message in a conversation. choice is -1 by default, meaning the current node is a message node. choice must be between 0 and the number of choices if the current node is a choice node. The demo shows an example of how to use this function.

## Script Terminology

Quantum Dialogue contains custom scripts, with each outlined in the table below.

**QD\_DialogueHandler** - Responsible for controlling a dialogue and the conversations within.

**QD\_Dialogue** - A list of conversations that contain speakers, messages, and player choices.

**QD\_Conversation** - Represents a conversation between different speakers, or just one.

**QD\_Speaker** - Represents an NPC or player that is speaking in a conversation.

**QD\_Message** - Represents one message in a conversation said by a speaker.

**QD\_Choice** - Represents a choice that the player must make in a conversation.