

Celtx Import Guide for Dialogue System for Unity

Updated: February 4, 2022

Table of Contents

Introduction.....	2
Celtx Importer Setup.....	2
Quick Start.....	4
Preparing CeltX Content.....	5
Terminology.....	5
Tags.....	5
Conversations and Sequence Maps.....	5
Conversation Root.....	6
CeltX Sequence Nodes.....	6
Interactive Dialogue Nodes.....	7
CeltX Script Formatting.....	8
Variables and Conditions.....	17
Localization.....	18
How To Get Help.....	19

Introduction

This document explains how to import content from Celtx into the Dialogue System for Unity. It assumes basic familiarity with Celtx and the Dialogue System for Unity.

Celtx Importer Setup

This section explains how to set up the Dialogue System’s Celtx importer.

1. Import the latest version of the [Dialogue System for Unity](#) into your Unity project.
2. If Newtonsoft.Json.NET is already installed in your project, setup is complete.

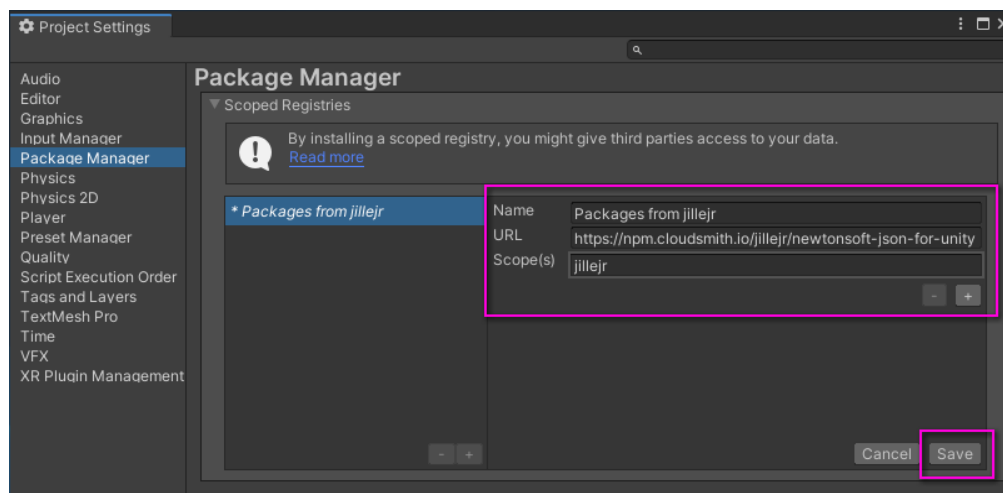
Otherwise, select menu item **Edit** → **Project Settings**. On the Package Manager page, click “+” if necessary to create a new, empty form. If nothing is registered yet in the Scoped Registries section, you don’t have to click “+”. Enter the following information into the form:

Name: Packages from jillejr

URL: <https://npm.cloudsmith.io/jillejr/newtonsoft-json-for-unity/>

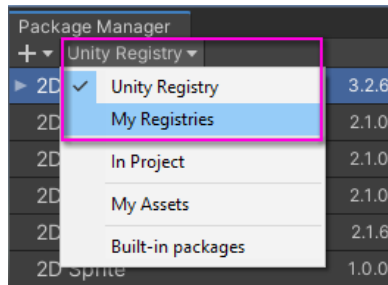
Scope(s): jillejr

Then click **Save**.

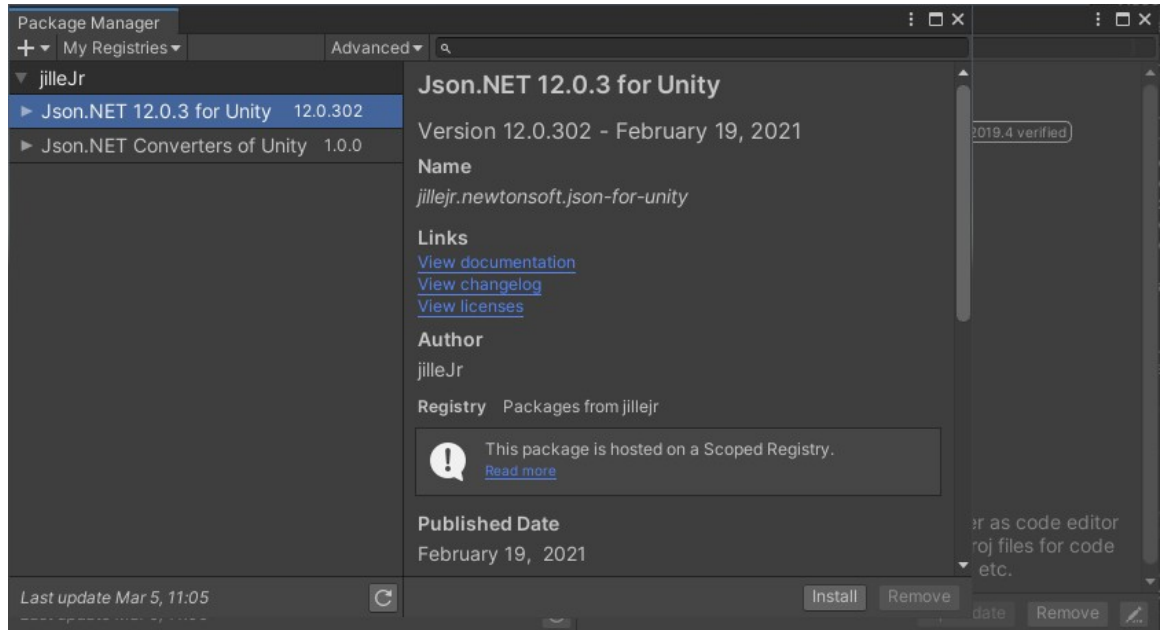


3. Select menu item **Window** → **Package Manager** to open the Package Manager window.

- Select **Unity Registry** → **My Registries**:



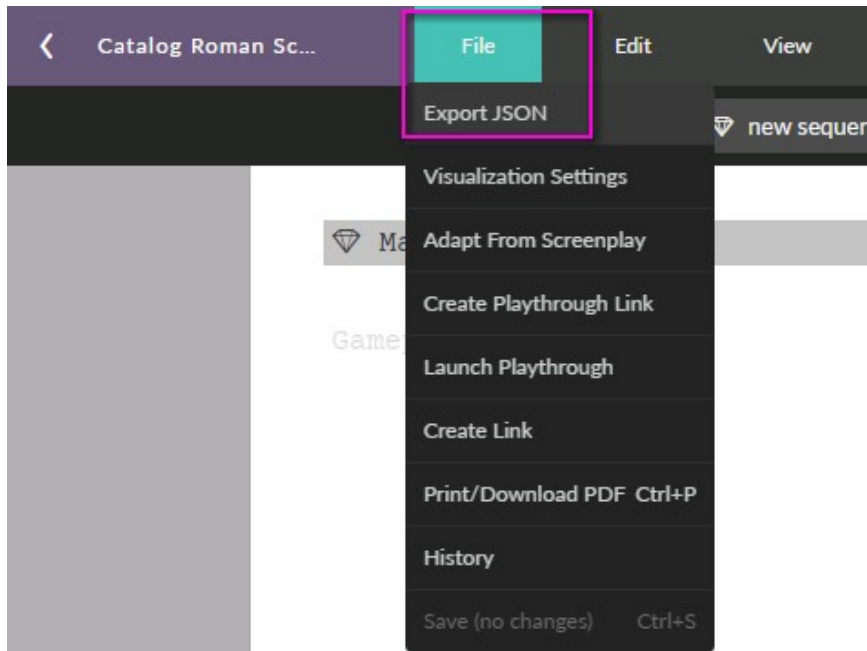
- Select **Json.NET** and click **Install**:



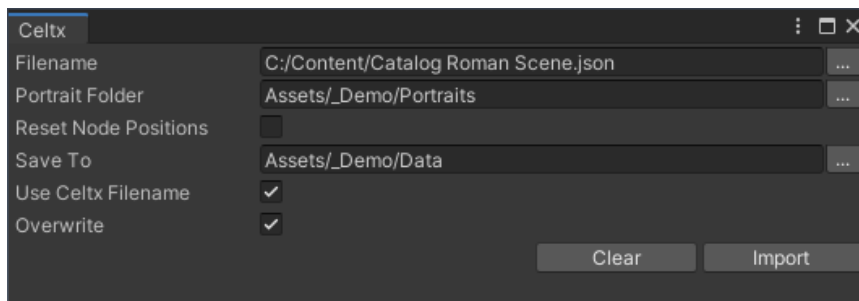
Quick Start

After the Dialogue System's Celtx importer has been set up using the steps above, use these steps to import your Celtx content into the Dialogue System.

1. In Celtx, export your Game & VR Script to a JSON file:



2. In Unity, select menu item **Tools** → **Pixel Crushers** → **Dialogue System** → **Import** → **Celtx** to open the Celtx Import window. The first time you select this menu item, it will ask if you want to enable Celtx import capability. Click **Enable**, and select the menu item again to open the window.
3. In the **Filename** field, select your JSON export file. Set the **Save To** field to the folder in which you want to create the dialogue database asset. The other fields are optional. Mouse hover tooltips describe their uses.



4. Click **Import**. This will create (or overwrite) a Dialogue System dialogue database asset.

Preparing Celtx Content

This section explains how to prepare Celtx content for use with the Dialogue System for Unity.

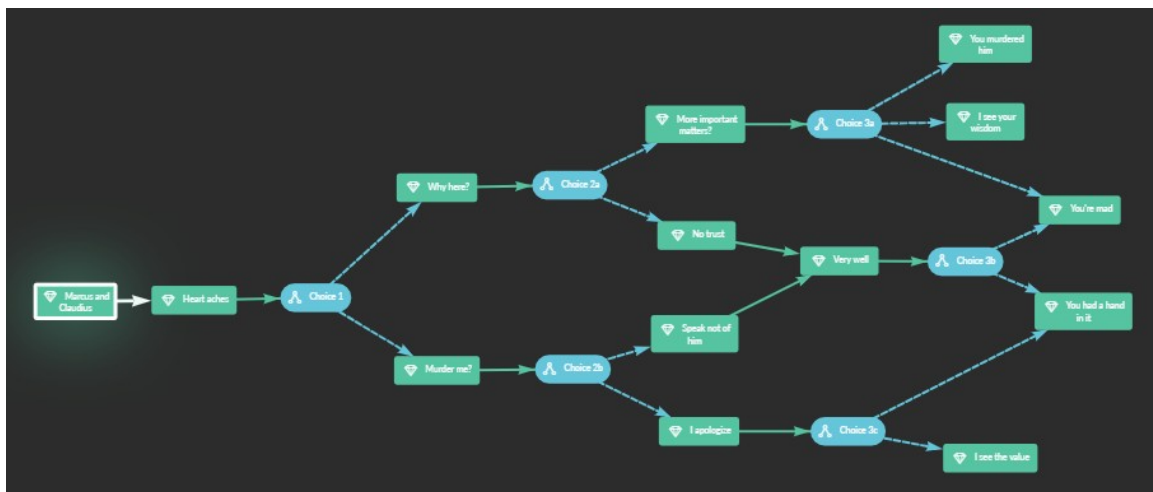
Terminology

Tags

The Dialogue System's Celtx importer makes use of special instructions contained within square brackets, such as [C]. This manual refers to these instructions as *tags*. When a tag is required, it must be located before any other text in its section.

Conversations and Sequence Maps

Conversations in the Dialogue System are generated from a network (or map) of Celtx *Sequence* and *Branch* nodes.



Conversation Root

A Sequence node without any incoming links denotes the beginning (root) of a conversation tree.

- This node must contain two Character objects, one per line:

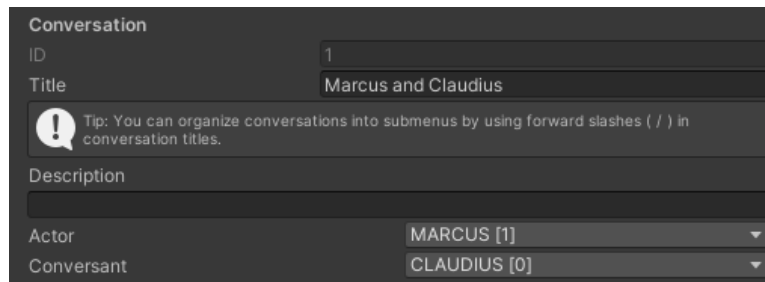
```
◆ Marcus and Claudius
```

MARCUS

CLAUDIUS

Dialogue

- The first character line indicates that the character with this name will be the Conversation Actor in the Dialogue System conversation generated from this sequence map.
- Likewise, the second character line indicates that the character with this name will be the Conversation Conversant in the Dialogue System conversation.
- The example node above will be represented by these settings in the Dialogue System:



The screenshot shows a configuration window for a conversation. It includes the following fields and values:

- Conversation ID:** 1
- Title:** Marcus and Claudius
- Description:** (empty)
- Actor:** MARCUS [1]
- Conversant:** CLAUDIUS [0]

A tip message is displayed: "Tip: You can organize conversations into submenus by using forward slashes (/) in conversation titles."

- Note: The importer will only look for the first two Character objects in this node. Any other data will be ignored. Additionally, if the node does not have 2 Character objects, that node's sequence map will be skipped and assumed to not represent a conversation.

CeltX Sequence Nodes

Subsequent Sequence nodes in the sequence map contain the conversation's content. The script page for each node will be used to generate dialogue entries (nodes in the conversation tree) in the Dialogue System. Any data that precedes a dialogue object in the CeltX sequence script will be added to the Dialogue System dialogue entry with that dialogue text. Once a dialogue object is found and processed, any data on the script that follows it will be added to a different dialogue entry linked from the previous one, until the next dialogue object is found. Subsequent dialogue entries will have the sequence name followed by "-NUM", where *NUM* corresponds to the number of entries that have been generated from that sequence. The following is an example of how content would be grouped.

```
Marcus, how my heart aches for your  
loss.
```

```
CLAUDIUS  
([C]Agatha)  
Marcus, how my heart aches for your  
loss.
```

```
CLAUDIUS  
(This would be your menu text)  
([VO]Claudius_Why_Here_1)  
Marcus, how my heart aches for your  
loss.
```

```
[SEQ]SALSA(entrytag); LiveCamera(Claudius_Closeup,,2);  
LookAt(listener,,1)
```

```
CLAUDIUS  
Marcus, how my heart aches for your  
loss.
```

```
[SEQ]SALSA(entrytag); LiveCamera(Claudius_Closeup,,2);  
LookAt(listener,,1)
```

```
([C]Agatha)  
([VO]Claudius_Why_Here_1)  
This second Dialogue text will  
generate a new entry with Actor  
Claudius and Conversant Marcus
```

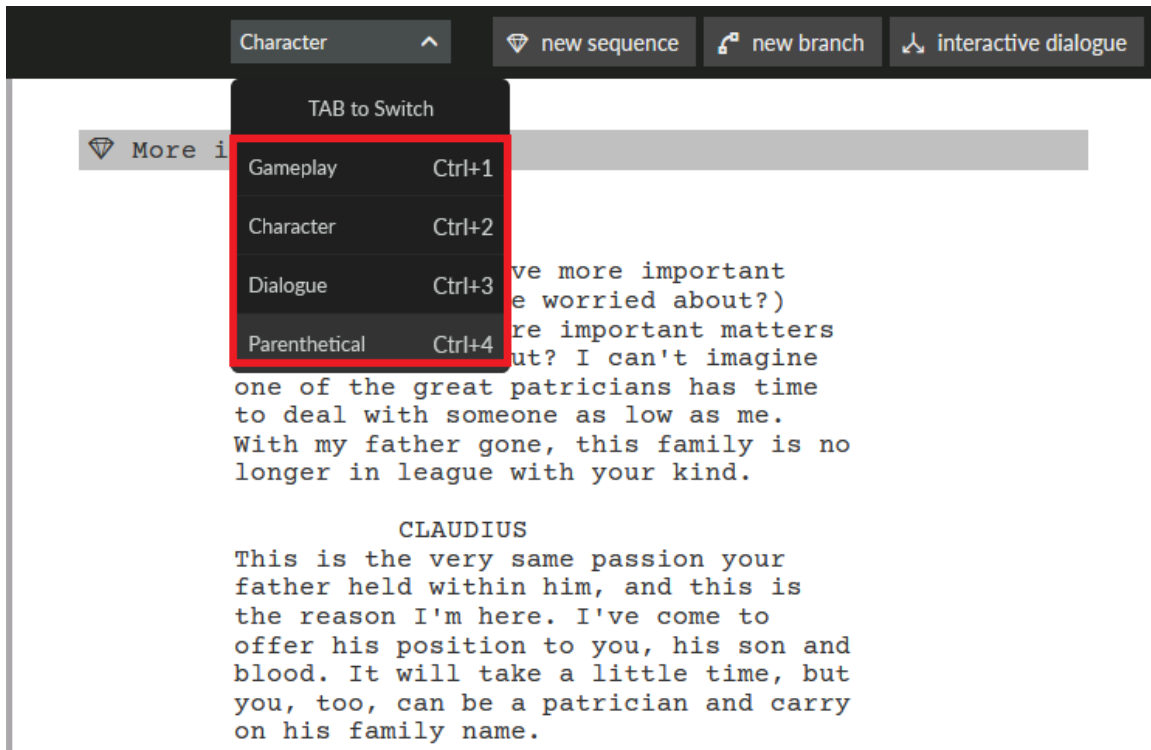
Interactive Dialogue Nodes

The Dialogue System importer does not use Interactive Dialogue nodes as they do not support conditional logic, nor do they provide an ability for conversation tree branches to converge.

CeltX Script Formatting

Follow the formatting guidelines outlined in this section to ensure your Dialogue System conversations are generated correctly.

Note: The terms *Dialogue*, *Character*, *Gameplay*, and *Parenthetical objects* refer to the following:



Dialogue

The most basic option is to include only a Dialogue object on the script page. The resulting dialogue entry will have the name of the sequence, the default Conversation Actor and Conversant (set on the conversation root node), and the script's dialogue text.

◆ Heart aches

Marcus, how my heart **aches** for your
loss.

The object above will be represented by this dialogue entry node in the Dialogue System:

Dialogue Entry

ID: 1

Title: Heart aches

Description:

Actor: MARCUS [1]

Conversant: CLAUDIUS [0]

Group:

Menu Text:

Dialogue Text (62 chars): `Marcus, how my heart aches for your <u>loss</u>.`

Dialogue Entry Actor (Character)

Adding a Character object to the script will override the Dialogue System Actor on the dialogue entry that is currently being processed. If this Character corresponds to the conversation's default Conversant, then the dialogue entry's Conversant field will be set to the conversation's default Actor.

CLAUDIUS
Marcus, how my heart **aches** for your
loss.

The object above will be represented by a dialogue entry spoken by CLAUDIUS in the Dialogue System:

Dialogue Entry

ID: 2

Title: Heart aches-2

Description:

Actor: CLAUDIUS [0]

Conversant: MARCUS [1]

Group:

Menu Text:

Dialogue Text (62 chars): `Marcus, how my heart aches for your <u>loss</u>.`

Dialogue Entry Conversant (Parenthetical)

To set a dialogue entry's Conversant (the character being addressed by the speaker) to a different character than the defaults explained above, add a Parenthetical object with the [C] tag and the name of the intended Character:

```
                CLAUDIUS
            ([C]Agatha)
Marcus, how my heart aches for your
loss.
```

The object above will result in a dialogue entry whose Conversant is AGATHA:

Dialogue Entry

ID: 3

Title: Heart aches-3

Description:

Actor: CLAUDIUS [0]

Conversant: AGATHA [3]

Group:

Menu Text:

Dialogue Text (62 chars): Marcus, how my heart aches for your <u>loss</u>.

Menu Text (Parenthetical)

Dialogue System dialogue entries may have Dialogue Text and optional Menu Text. Menu Text is typically a short paraphrase of the full Dialogue Text. To add Menu Text to a dialogue entry, include a Parenthetical object with no tags:

```
                CLAUDIUS
            (This would be your menu text)
Marcus, how my heart aches for your
loss.
```

The object above would result in this menu text in the corresponding dialogue entry:

Dialogue Entry

ID: 4

Title: Heart aches-4

Description:

Actor: CLAUDIUS [0]

Conversant: MARCUS [1]

Group:

Menu Text (28 chars): This would be your menu text

Dialogue Text (62 chars): Marcus, how my heart aches for your <u>loss</u>.

Voiceover Files (Parenthetical)

If your dialogue has a voiceover, you may specify the value of the Dialogue System dialogue entry's VoiceOverFile field by including a Parenthetical object containing the [VO] tag, followed by the name of the voiceover file.

```

CLAUDIUS
([VO]Claudius_Why_Here_1)
Marcus, how my heart aches for your
loss.

```

The object above will set the dialogue entry's VoiceOverFile field to the value shown in the image on the next page.

Dialogue Entry

ID: 5
 Title: Heart aches-5
 Description:

Actor: CLAUDIUS [0]
 Conversant: MARCUS [1]
 Group:

Menu Text:
 Dialogue Text (62 chars): Marcus, how my heart aches for your <u>loss</u>.

Sequence: [Check] +
 Add Response Menu Sequence

Conditions: ...
 False Condition Action: Block

Script: ...

▶ Events

▼ All Fields

	Value	Template	Copy	Paste	
Title	Heart aches-5	Text	↑	↓	⊖
Pictures	☐	Files	↑	↓	⊖
Description		Text	↑	↓	⊖
Actor	CLAUDIUS [0]	Actor	↑	↓	⊖
Conversant	MARCUS [1]	Actor	↑	↓	⊖
Menu Text		Text	↑	↓	⊖
Dialogue Text	Marcus, how	Text	↑	↓	⊖
Parenthetical		Text	↑	↓	⊖
Audio Files	☐	Files	↑	↓	⊖
Video File		Text	↑	↓	⊖
Sequence		Text	↑	↓	⊖
Celtx ID	mkdUUMKO-5	Text	↑	↓	⊖
VoiceOverFile	Claudius_Why_Here_1	Files	↑	↓	⊖

Links To: (Link To)
 Marcus, how my heart aches for your <u>loss</u>

Dialogue Entry 'Sequence' Field (Gameplay)

To add Dialogue System cutscene sequencer commands to a dialogue entry, add a Gameplay object beginning with the tag [SEQ]. All text following the tag being added to the dialogue entry's Sequence field.

```
[SEQ]SALSA(entrytag); LiveCamera(Claudius_Closeup,,2);  
LookAt(listener,,1)
```

```
                CLAUDIUS  
    Marcus, how my heart aches for your  
    loss.
```

The [SEQ] object above will add these cutscene sequencer commands to the dialogue entry's Sequence field:

The screenshot shows a configuration panel for a Dialogue Entry. The fields are as follows:

- ID: 6
- Title: Heart aches-6
- Description: (empty)
- Actor: CLAUDIUS [0]
- Conversant: MARCUS [1]
- Group: (empty)
- Menu Text: (empty)
- Dialogue Text (62 chars): Marcus, how my heart aches for your <u>loss</u>.
- Sequence: SALSA(entrytag); LiveCamera(Claudius_Closeup,,2); LookAt(listener,,1). Includes a 'Check' button and a '+' icon.
- Add Response Menu Sequence: (checkbox is unchecked)
- Conditions: (empty)
- False Condition Action: Block
- Script: (empty)

Dialogue Entry 'Conditions' Field (Gameplay)

To add content to a dialogue entry's Conditions field, add a Gameplay object beginning with the tag [COND]. All text following the tag being added to the dialogue entry's Conditions field.

```
[COND] GetInventoryQuantity("Apples") > 3
```

Dialogue Entry 'Script' Field (Gameplay)

To add content to a dialogue entry's Script field, add a Gameplay object beginning with the tag [SCRIPT]. All text following the tag being added to the dialogue entry's Script field.

```
[SCRIPT] ShowAlert("Claudius!")
```

See also the [Variables and Conditions](#) section for an explanation of how Celtx native logic also translates into Dialogue System Conditions and Script fields.

Split Dialogue into Multiple Dialogue Entries

If no Character field is added between two dialogue objects, the latter of the two entries generated will contain the same Actor and Conversant as the previous entry. Any other data may be specified normally for these subsequent entries, such as Dialogue System Sequence, Conversant, or VoiceOverFile field data.

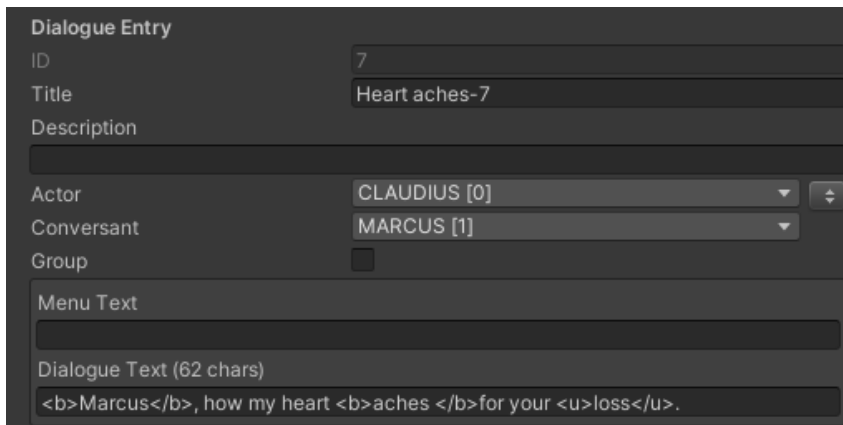
```
                CLAUDIUS
    Marcus, how my heart aches for your
    loss.

[SEQ]SALSA(entrytag); LiveCamera(Claudius_Closeup,,2);
LookAt(listener,,1)

    ([C]Agatha)
    ([VO]Claudius_Why_Here_1)
    This second Dialogue text will
    generate a new entry with Actor
    Claudius and Conversant Marcus
```

The content above will generate the following two dialogue entries in the Dialogue System:

First entry:



The screenshot shows a 'Dialogue Entry' configuration window with the following fields:

- ID: 7
- Title: Heart aches-7
- Description: (empty)
- Actor: CLAUDIUS [0]
- Conversant: MARCUS [1]
- Group: (checkbox unchecked)
- Menu Text: (empty)
- Dialogue Text (62 chars): Marcus, how my heart aches for your <u>loss</u>.

Second entry:

Dialogue Entry

ID: 8

Title: Heart aches-8

Description:

Actor: CLAUDIUS [0]

Conversant: AGATHA [3]

Group:

Menu Text:

Dialogue Text (93 chars): This second Dialogue text will generate a new entry with Actor Claudius and Conversant Marcus

Sequence: +

SALSA(entrytag); LiveCamera(Claudius_Closeup,,2); LookAt(listener,,1)

Add Response Menu Sequence

Conditions: ...

False Condition Action: Block

Script: ...

Events

▼ All Fields

Title	Value	Type	↑	↓	⊖
Title	Heart aches-8	Text	↑	↓	⊖
Pictures	[]	Files	↑	↓	⊖
Description		Text	↑	↓	⊖
Actor	CLAUDIUS [0]	Actor	↑	↓	⊖
Conversant	AGATHA [3]	Actor	↑	↓	⊖
Menu Text		Text	↑	↓	⊖
Dialogue Text	This second Dialogue	Text	↑	↓	⊖
Parenthetical		Text	↑	↓	⊖
Audio Files	[]	Files	↑	↓	⊖
Video File		Text	↑	↓	⊖
Sequence	SALSA(entrytag);	Text	↑	↓	⊖
Celtx ID	mkdUUMKO-8	Text	↑	↓	⊖
VoiceOverFile	Claudius_Why_Here_1	Files	↑	↓	⊖

Cross-Conversation Links (Jump)

To link from one conversation to another, use a Jump in Celtx. This will result in a cross-conversation link in the Dialogue System.

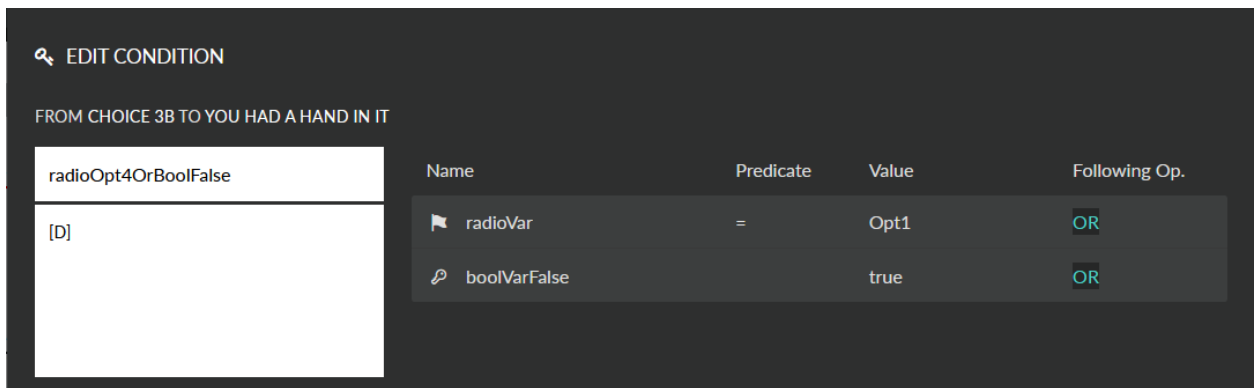
Variables and Conditions

Variable values can be set on any CeltX node. In the Dialogue System, the variable will be set in the Script field of the first dialogue entry generated for the corresponding node.

The Dialogue System will process Variable and Condition literals when generating a Lua expression for a dialogue entry's Conditions field. Asset literals will be ignored. Additionally, "Exists" and "Does Not Exist" operations will default to "Equals" and "Does Not Equal" in all cases.

If a condition is set on the transition between two nodes in CeltX, the Dialogue System will generate a new group dialogue entry containing the condition's corresponding lua code between the two linked nodes.

Note: Please keep in mind that the [Dialogue System evaluates conditions one extra level ahead](#) to accommodate certain continue button modes. To generate the "intermediate node" for a condition's evaluation to be delayed, simply add a [D] tag to the condition's description in CeltX. This delay entry will be added as an extra group dialogue entry between the original source node and the generated condition node:

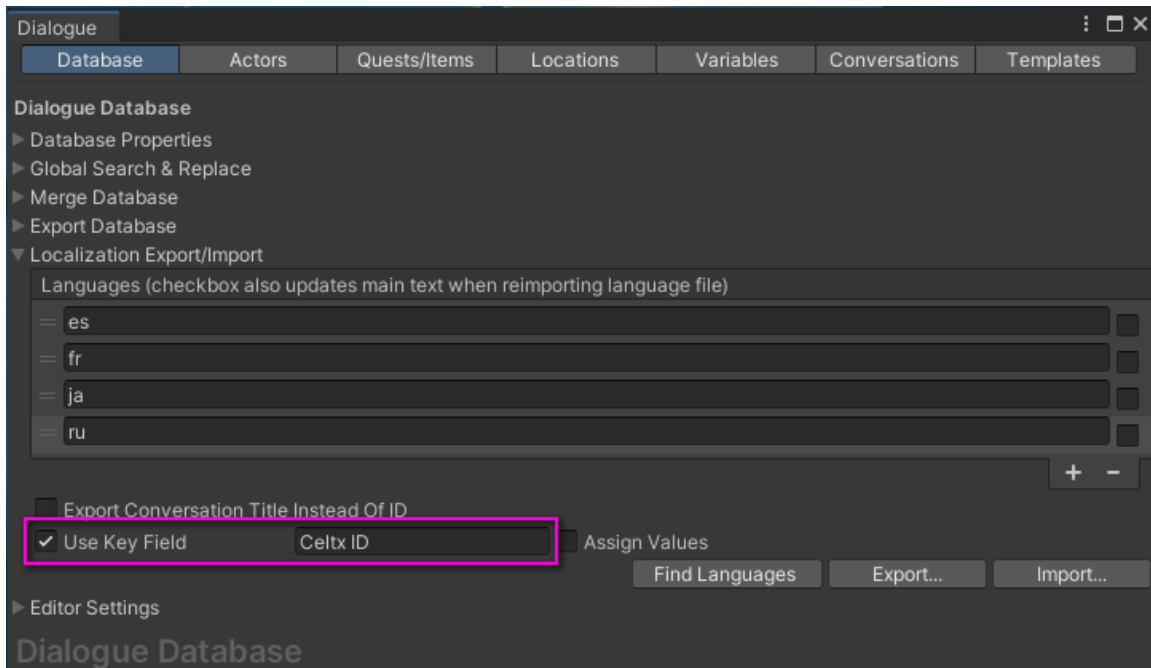


You can also create Gameplay nodes that start with [COND] or [SCRIPT] to add Conditions and Script content, respectively, to Dialogue System dialogue entries.

Localization

To localize your Celtx content, import your Celtx JSON file into a dialogue database. Then open the dialogue database in the Dialogue System's Dialogue Editor window.

1. In the **Database** section, expand **Localization Export/Import**.
2. Add the language you want to localize to.
3. Tick **Use Key Field**. Enter `Celtx ID` (exact spelling and capitalization, with space):



4. Click **Export....** Specify a folder on your computer. This will create a CSV spreadsheet for each language. Enter the translations in the spreadsheet.
5. When the CSV spreadsheets have been translated, click **Import...** to import them back into your dialogue database.

For more information on language localization in the Dialogue System, please see the [Localization](#) section of the online manual.

How To Get Help

Pixel Crushers maintains and supports the Dialogue System's Celtx importer. If you have questions, bug reports, or feature suggestions, please contact us:

- Email: support@pixelcrushers.com
- Forum: <https://pixelcrushers.com/phpbb/>
- Discord: <https://discord.gg/FwUaCNt>
- Bug Report Form: Tools → Pixel Crushers → Dialogue System → Help → Report a Bug