

DialogUtility Spruce usage guide

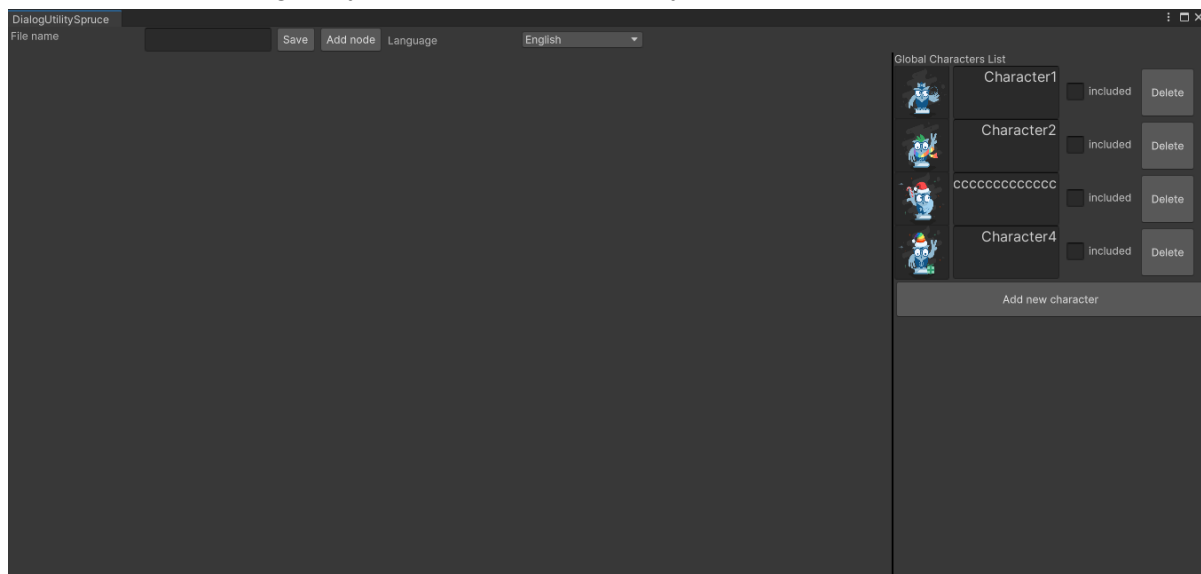
About “Spruce”:

This dialog utility provides tools to build dialog nodes, compose them into decision trees with condition scriptable objects. Aside setting an actual text and conditions it also provides features like localization and character list.

Gettings started:

Creating Dialog

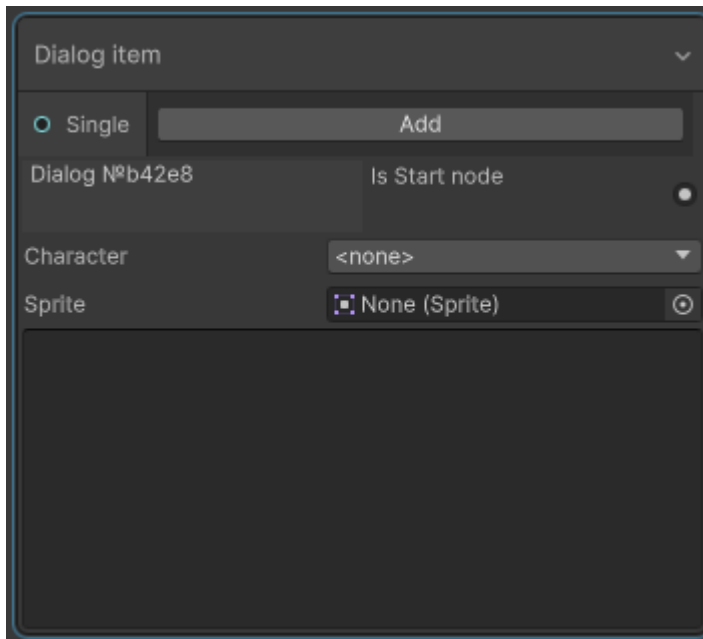
Click Window->DialogUtilitySpruce to open the utility window.




Use File name text field on the top to set name of the dialog. All related files (localisation) will have name based on this value. Changing this value and pressing Save will always rename opened container, not create new one with different name!

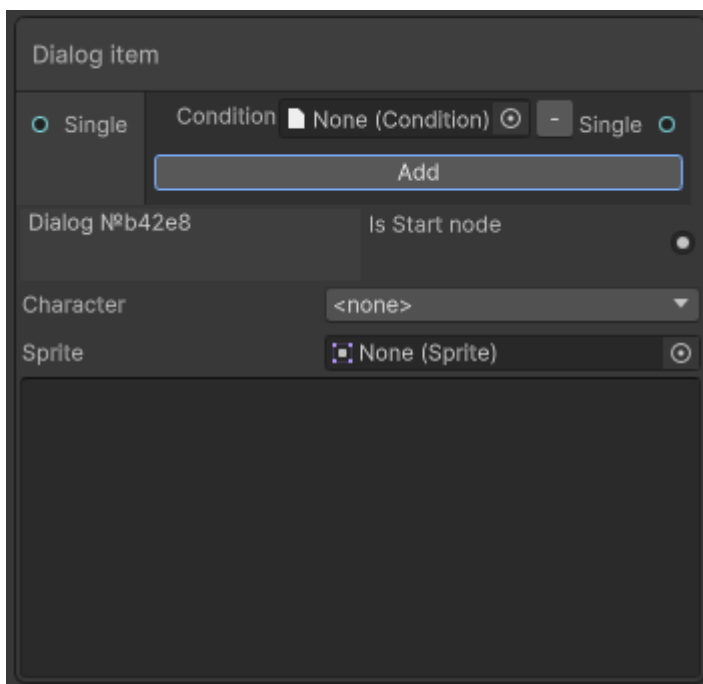
Adding nodes

Use Add Node or Right mouse button in graph area and Add in context menu to add nodes. Deleting nodes can be performed with same context menu or by pressing delete button on your keyboard.



New dialog item will appear. Text field can be used to write dialog content. Add button will add dialog branching (dialog option paths). Each branch can use condition scriptable object defining option availability. By default all options without condition set are considered

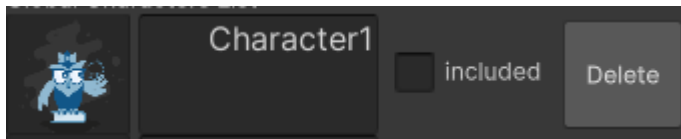
available. Use  button to delete option. Use Sprite field to assign an image to this dialog. Use Is Start node checkbox to set node as a start of the dialog.



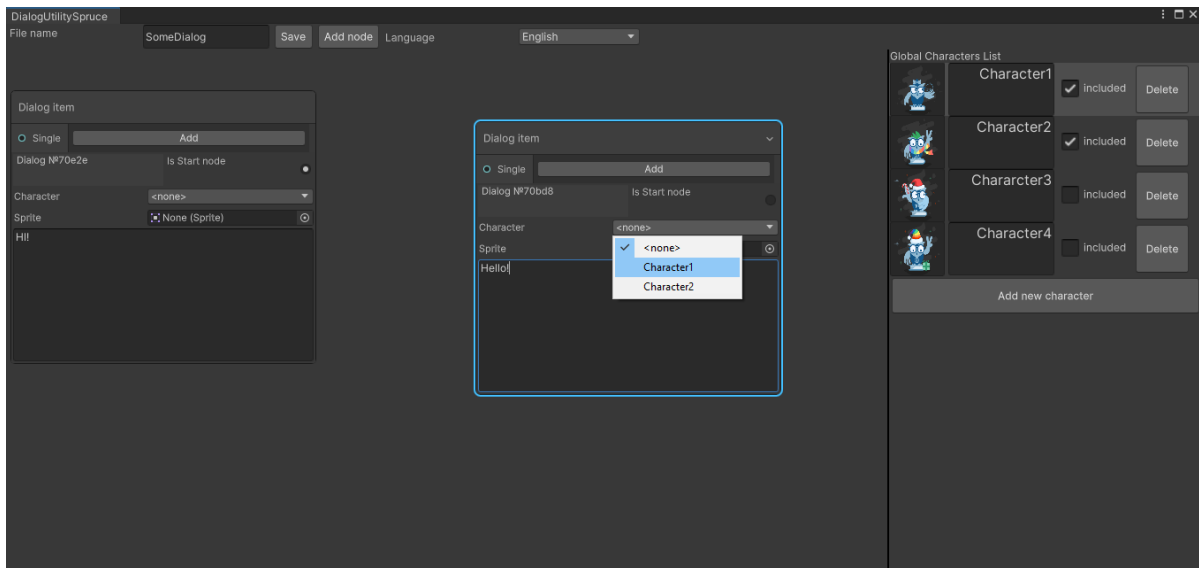
Creating and using Characters

One of the main features of Spruce is the character list. Use Add new character button to create new character. Use Sprite field to set character sprite (good for icons or any other

related things). Use Name textfield to set characters name. Use Include checkbox to include this character into the current dialog. Use the delete button to delete character.

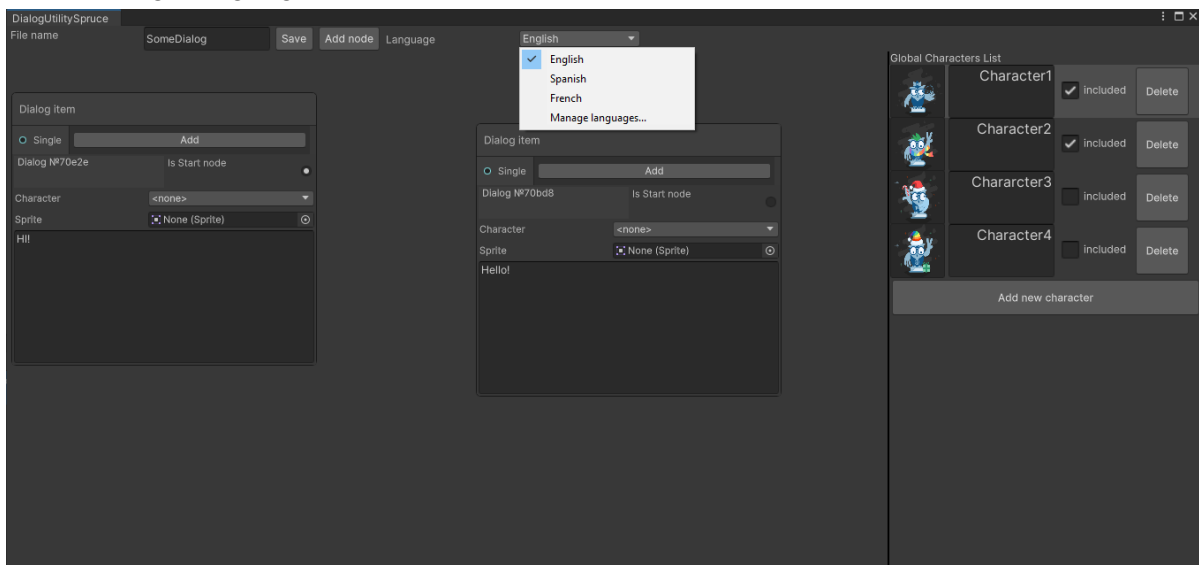


Each character included now can be used in dialog:

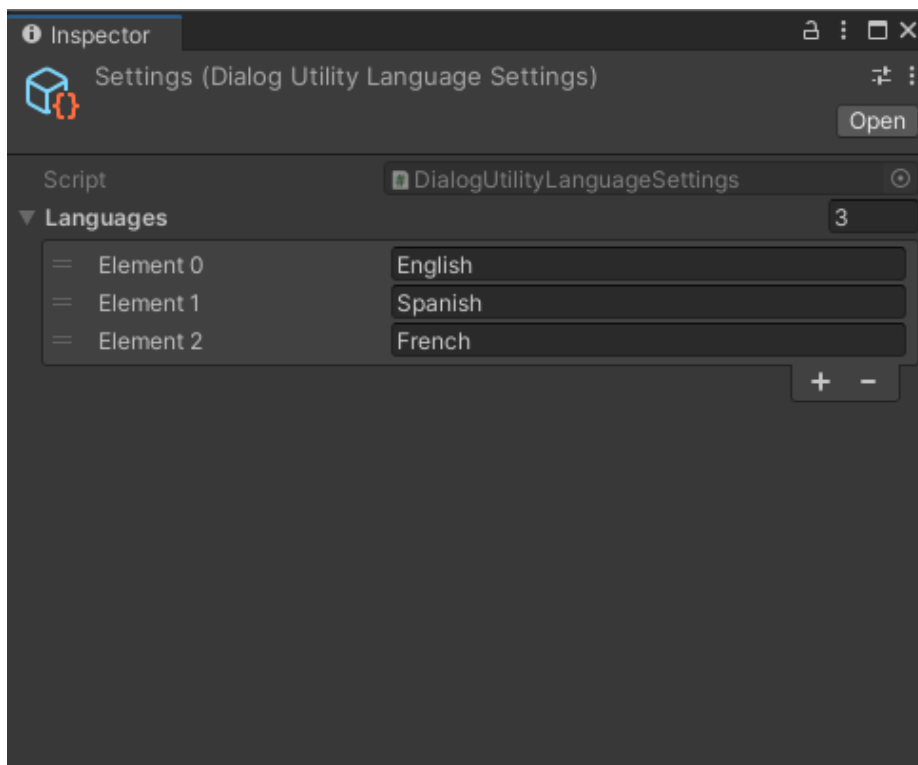


Adding localisation

You can set different text field values for different language open language dropdown and click “Manage languages...”



Inspector will select language settings object, that includes list with all languages:



You can add new languages there, they will appear in languages dropdown.

Switching languages will switch current localisation. **By default all values from previous localisation will be copied for new localisation!**

Scripting

DialogReader

DialogReader class is used to read data from dialog containers. Take a look at DialogBox as an example of DialogReader usage.

Properties:

IsActive	Returns true if DialogReader begun to read dialog and has not ended yet
Graph	Adjacency-list representation of dialog graph

Events:

OnNextMessage	This event is called with next message DialogNode as a parameter when the reader iterates to next message.
OnDialogEnded	This event is called when reader iterates to the end of the dialog

OnDialogStarted	This event is called with starting message DialogNode as a parameter
-----------------	--

Methods:

BeginDialog	Loads dialog graph container into dialog reader and begins iteration. Localization is loaded on this step
NextMessage	Iterates to next message, prioritizing available or empty options if choice parameter is null. Ends dialog iteration if the message dialog node has no continuation
EndDialog	Ends dialog.

DialogReaderSettings

DialogReaderSettings is a static class that is used to setup language for dialog readers.

Constants:

DefaultLanguage	Returns default language string
-----------------	---------------------------------

Properties:

Language	Returns language string
----------	-------------------------

Events:

OnInitialize	This event is called with language string as a parameter when DialogReaderSettings has been initialized
--------------	---

Methods:

Initialize	Initializes DialogReaderSettings with language provided. This method has to be called before using any dialog readers.
GetContainerLocalisation	Returns Localisation resource for container with given name

DialogNode

DialogNode is a class that contains all data of dialog node.

Properties:

Character	Returns Character object of character assigned to this node
Text	Returns string containing text of message
Sprite	Returns Sprite assigned to the dialog node
ChoiceOptions	Returns List of DialogChoiceOption

Character

Character is a class containing all data of the character.

Properties:

Name	Returns string containing character name
Icon	Returns sprite containing icon of the character

DialogChoiceOption

DialogChoiceOption is a class describing dialog node output connection options.

Properties:

Condition	Returns Condition scriptable object, describing condition of availability of this dialog option
Index	Returns index of this dialog option

Creating conditions

Condition

Condition is a class that is used to create scriptables describing dialog options availability. Here is an example of creating your own Condition type:

```
[CreateAssetMenu(menuName =
"DialogUtility/Examples/SomethingExistsCondition")]

public class SomethingExistsCondition : Condition
{
    public string[] objectName;

    public override bool IsTrue(DialogNode node, int index)
    {
        return GameObject.Find(objectName[index]);
    }
}
```

IsTrue method will be called to check if the dialog option is available.

Localization files:

All localization files are stored separately in Resources/DialogUtility/ContainerLocalizations. You can add to build only localizations required with addressables/assetbundles. Spruce will load only localisation defined in DialorReaderSettings.