

Text Table Manual

Pixel Crushers Common Library

Copyright © Pixel Crushers. All rights reserved.

Contents

| | |
|---|---|
| Chapter 1: Introduction to Text Tables..... | 4 |
| In This Manual..... | 4 |
| How to Get Help..... | 4 |
| Chapter 2: Text Tables..... | 5 |
| Languages..... | 5 |
| Fields..... | 6 |
| Using Text Tables in Scripts..... | 6 |
| Global Text Table..... | 6 |
| Chapter 3: Localization..... | 7 |
| Note About Empty Fields..... | 7 |
| Localized Fonts..... | 7 |
| Importing Localization Fields..... | 7 |
| Chapter 3: Import and Export..... | 8 |

Chapter 1: Introduction to Text Tables

The Pixel Crushers Common Library includes text tables and related components which will help you manage localization and text data lookups. Text tables support exporting and importing from comma-separated value (CSV) spreadsheet format, which is useful for working with translators.

In This Manual

The rest of this manual contains these chapters:

- **Chapter 2: Text Tables** – *How to create and edit text tables*
- **Chapter 3: Localization** – *How to configure UI localization*
- **Chapter 4: Import and Export** – *How to work with external formats*

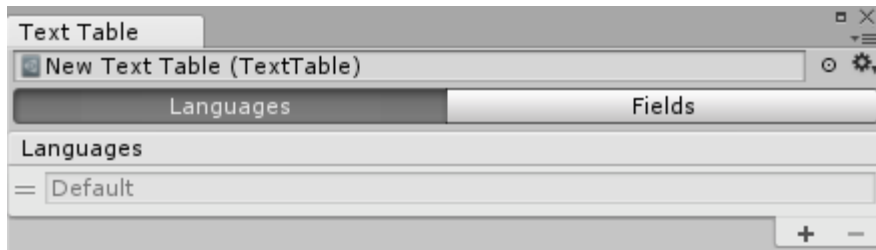
How to Get Help

We're here to help! If you get stuck or have any questions, please contact us any time at support@pixelcrushers.com or visit <http://pixelcrushers.com>.

We do our very best to reply to all emails within 24 hours. If you haven't received a reply within 24 hours, please check your spam folder.

Chapter 2: Text Tables

Text tables are project assets. To create a text table, right-click in the Project view and select **Create** → **Pixel Crushers** → **Common** → **Text** → **Text Table**. In the Inspector, click **Open Text Table Editor**.

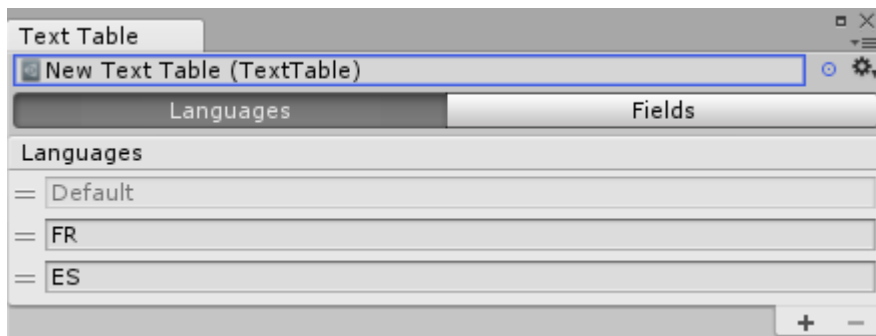


Text tables have two tabs:

- **Languages:** Let you define the list of languages in this text table.
- **Fields:** Contains the text table's entries (fields). Each field has a translation for each language defined on the Languages tab.

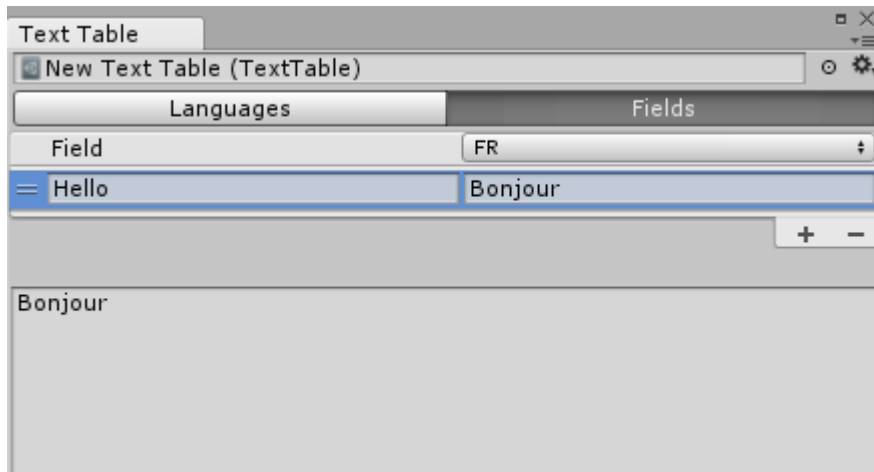
Languages

To add a new language, on the **Languages** tab click the “+” button. Then enter the name by which you want to refer to the language.



Fields

To add a new field, on the **Fields** tab click the “+” button. Enter the field name in the **Field** column. Then select each language from the dropdown menu in the header of the right column, and enter the field value. You can enter the field value directly into the list row, or you can click on the list row to edit it in a larger text area at the bottom of the window.



Using Text Tables in Scripts

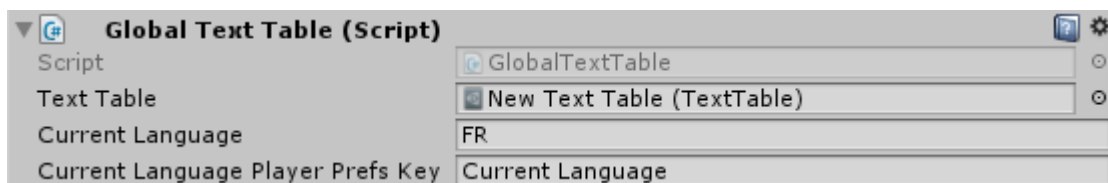
To get the value of a text table's field in a script, call the text table's `GetFieldTextForLanguage` method:

```
var hello = myTextTable.GetFieldTextForLanguage("Hello", "FR");
```

Text tables have several other helpful methods that you can explore through the online API reference. However, you may find it more convenient to use a Global Text Table, described below.

Global Text Table

The **Global Text Table** component is optional. However, it allows you to add a reference to a text table that other components can use without explicitly referencing the table. The `GlobalTextTable` class is also covered in the online API reference.



Chapter 3: Localization

To set up localization of Unity UI elements:

1. Optionally add a **Global Text Table** to the scene, and assign a text table.
2. Add a **UI Localization Manager** component to the scene. If you didn't add a Global Text Table, assign a text table to this component.
3. Add a **Localize UI** component to any UI Text or Dropdown. Optionally assign the **Field Name**. Otherwise the component will use the Text or Dropdown's starting value as the field name.

When the scene starts, and whenever you change languages by calling the `UILocalizationManager's UpdateUIs(string language)` method, the Localize UI components will look up the values of their field names in the text table and update their text content.

Note About Empty Fields

If you leave a field blank for a language, the Localize UI component will use the blank string. It will *not* revert to the default language for that string. This is because it may be intentional that you want to show a blank string for some languages.

To quickly populate a language's fields with default values, you can export the text table to CSV as described in the next section. Then copy the default language's column and reimport the CSV.

Localized Fonts

You can create a **Localized Fonts** assets to specify which font to use for each language. Assign it to the UI Localization Manager to use it globally, or assign it to Localize UI components or Set Localized Font components. The Set Localized Font component sets the font on a GameObject that doesn't use a Localize UI component, such as a Subtitle Text GameObject in the Dialogue System for Unity.

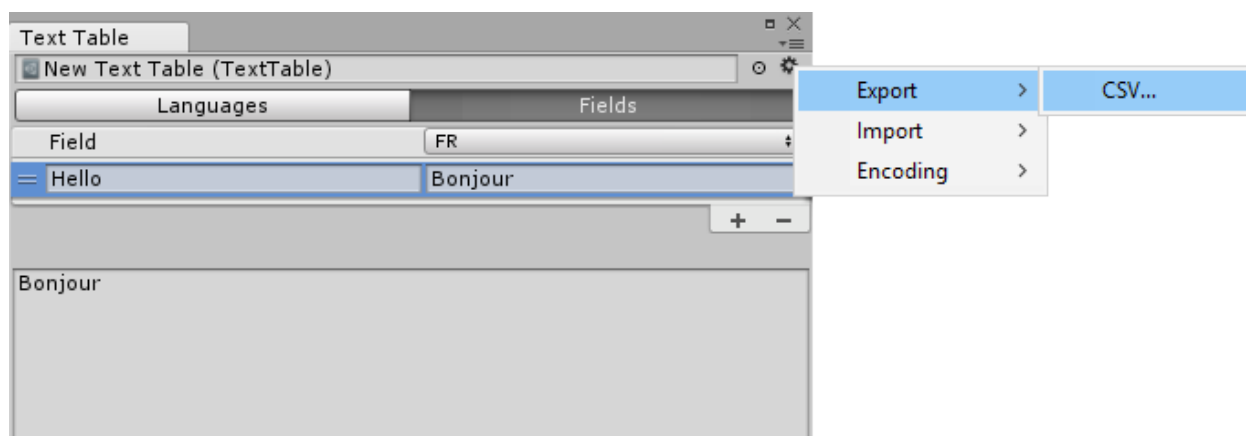
Importing Localization Fields

Use the Text Table Editor window's gear menu > **Import > Fields From Localize UI** to scan the currently-open scene for all Localize UI components and add corresponding fields to the text table.

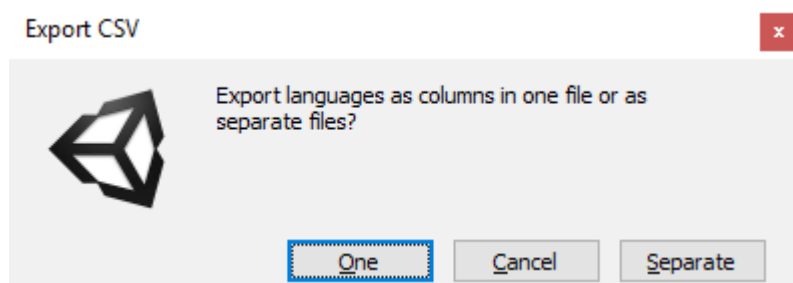
This feature also offers an option to automatically add Localize UI components to applicable GameObjects in your scene first.

Chapter 3: Import and Export

You can export text tables to comma-separated value (CSV) spreadsheets that you can open in any spreadsheet application such as Microsoft Excel or Google Sheets. To export the text table, select **Export** → **CSV...**



After specifying a filename, a popup window will ask if you want to export all languages to one file or in separate files.



If you export to one file, each language will occupy a column of the spreadsheet. If you export to separate files, the filenames will be appended with the language name (e.g., output_FR.xsv), and each file will only have one language column.

You can export and import in different encoding formats. For international characters, you may need to set the proper encoding format for exporting and importing.