

# Workflow Toolkit

## Documentation

Chomp Studios © 2024

# Contents

Document Information	3
Thanks	4
Getting Started	5
How To Use	6
Localisation	7
Bugs & Feedback	8
Adding Languages	9-14
Tools	15-19
Future Plans	20
Version Archive	21
Join Our Community	22

# Document Information

Start Date: 30/10/2024

Current Version: 0.6

Last Modified: 26/02/2025

# Thanks

## Thank You

Thank you for using this tool. It has been developed to optimise the Unity Experience, and I hope it works for you, too. If you have any feedback or suggestions, please share them - your input improves the tool for everybody.

# Getting Started

Genre: Plugin / Tool

Programmer(s): Wattie

## Description

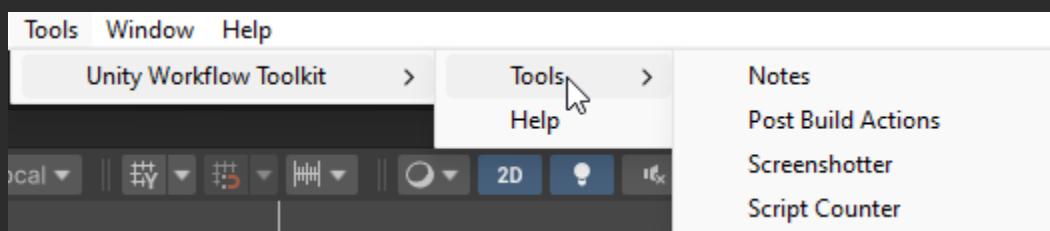
Unity Workflow Toolkit is a set of Plugins / Tools for Unity. This project focuses on delivering QOL (Quality Of Life) Features, making it easier for developers to create and manage their projects.

## Dependencies

This tool is designed for Unity **2022.3.18f1**, but I plan to allow this to work from Unity 2019 to the present.

## How To Locate

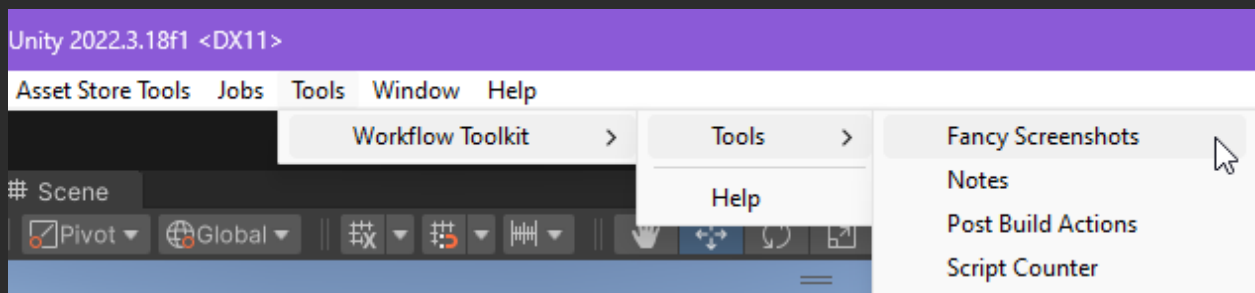
Once imported, look in the Menu Bar. To the left of “Window”, you should see the **Tools** Bar; hover over it, go to “Unity Workflow Toolkit”, then select Tools or the help window. (**Tools > Workflow Toolkit > Tools**)



# How To Use

## Accessing the Toolkit

Go to `Tools > Workflow Toolkit > Tools` in the Unity toolbar. This will open the main menu where all toolkit tools are available.



## Configuring Tools

Each tool comes with customisable settings. Open a tool and look for sections like “Settings” or “Options” to adjust according to your needs.

## Saving and Exporting

The tools can save data and configuration. These can be configured in each tool’s settings for easy reuse.

## Getting Help

For additional guidance on specific tools, check the "Help" section within the toolkit. You’ll find documentation and links to more resources on the website.

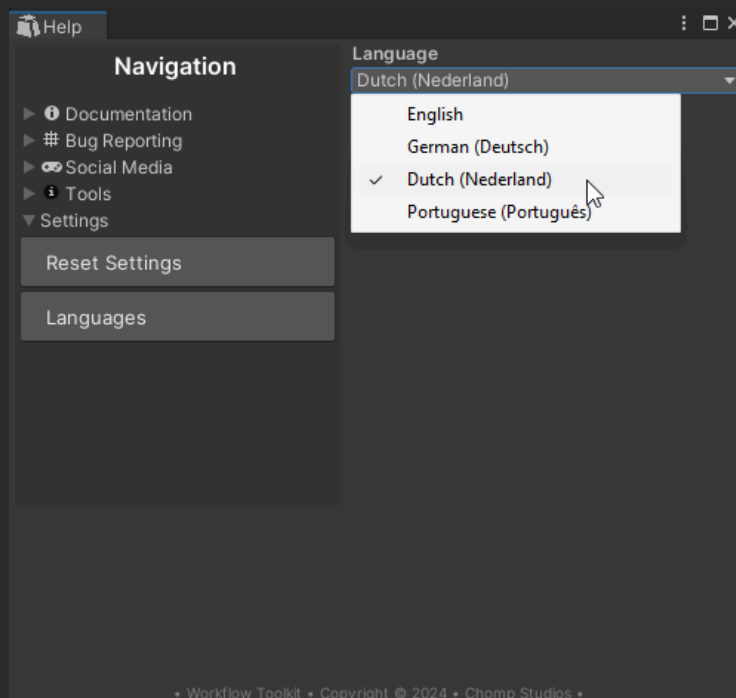
# Localisation

## Which languages are supported?

Currently, English, Dutch (Netherlands) and German are supported, with plans to support Portuguese (European) and Spanish - translators to this project are always welcome.

## How to switch language

Go to the [Help](#) window and go to the “settings” navigation area, click on “Languages”, and switch languages with the dropdown.



# Bugs & Feedback

## How To Report Bugs

To report a bug, open the [Help](#) window. In the "Bug Reporting" section, click the hyperlink to the Google questionnaire sheet - alternatively, find it [here](#).

## How To Submit Feedback

To submit feedback, open the [Help](#) window and head to the “bug reporting” section. In the dropdown, click the “Submit Feedback” button to display the feedback questionnaire hyperlink - alternatively, find the link [here](#).



# Adding Languages

## Disclaimer

This is a tutorial on how to implement your own language, or feel free to be silly with it! like Pirate and High Valyrian. If you would like to be an official translator for this project, feel free to get in touch with me VIA my email: [localisation@chompgames.co.uk](mailto:localisation@chompgames.co.uk)

## Getting Started

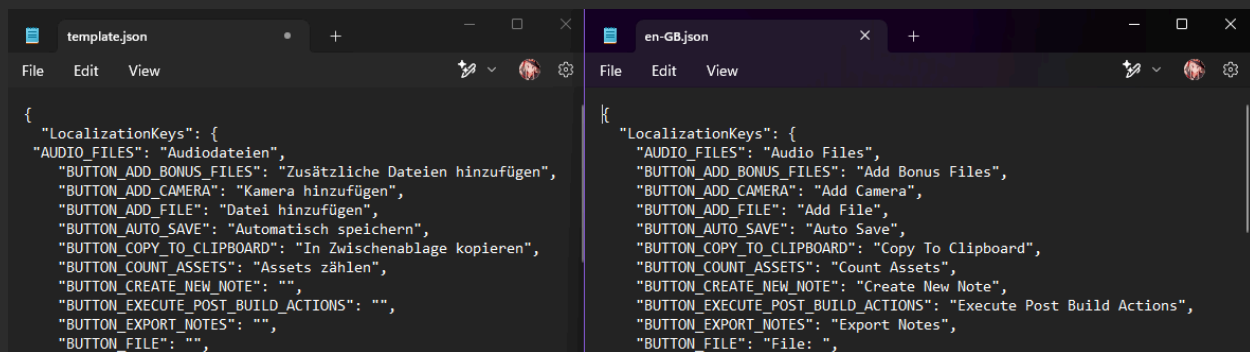
On the website, please go to the “Documentation” tab, and scroll down to “Language Tracker”, you’ll see a dropdown that says to download the language template, please do this. You’ll see a template.json file.



# Adding Languages

## Translate!

Now, open up the template.json file, and your reference language file and translate each key until complete.



## Naming

Although this step can be skipped, for clarity please name the now-translated text file to your locale identifier with region subtag - for example, English in the United Kingdom would be en-GB, whereas the USA would be en-US. You can find a list of these [\[here\]](#) . Find yours by ctrl + F'ing your country's name.

de-DE.json

pt-PT.json

nl-NL.json

# Adding Languages

## Implementing

Now, head to Workflow Toolkit. Add your language file to the “Utilities > Localisation > Languages” directory. Open up these two scripts: LanguageCode.cs & LanguageElements.cs.

## LanguageCode.cs

Add the name of your language, with the following number in line.

```
13
14 namespace Workflow_Toolkit.Utilities.Localisation {
    27 usages  wattle *  8 exposing APIs
15     public enum LanguageCode {
16         ENGLISH = 0,
17         GERMAN = 1,
18         DUTCH = 2,
19         PORTUGUESE = 3,
20         SPANISH = 4,
21         DANISH = 5,
22         TUTORIAL_LANGUAGE = 6
23     }
24 }
25
```

# Adding Languages

## LanguageElements.cs

In this file, you will see two functions - `GetLanguageDisplayName()` and `GetLanguageFileName()`. Both of these functions require you to add your language to them. Above the “default” line, add your language with this code:

```
case LanguageCode.TUTORIAL_LANGUAGE:
    return "LanguageNameInEnglish (LocalisedLanguageName)";
```

For example, German would look like this:

```
case LanguageCode.GERMAN:
    return "German (Deutsch)";
```

```
32     public static string GetLanguageDisplayName(LanguageCode code) {
33         switch (code) {
34             case LanguageCode.ENGLISH:
35                 return "English";
36             case LanguageCode.GERMAN:
37                 return "German (Deutsch)";
38             case LanguageCode.DUTCH:
39                 return "Dutch (Nederland)";
40             case LanguageCode.PORTUGUESE:
41                 return "Portuguese (Português)";
42             case LanguageCode.SPANISH:
43                 return "Spanish (Español)";
44             case LanguageCode.DANISH:
45                 return "Danish (Dansk)";
46             case LanguageCode.TUTORIAL_LANGUAGE:
47                 return "LanguageNameInEnglish (LocalisedLanguageName)";
48             default:
49                 return code.ToString();
50         }
51     }
```

# Adding Languages

## LanguageElements.cs

For `GetLanguageFileName()`, the process is largely the same, but now ensuring your language file's name is connected. Find the template code below, and the German example.

```
case LanguageCode.TUTORIAL_LANGUAGE:
    return "tut-TUT";
```

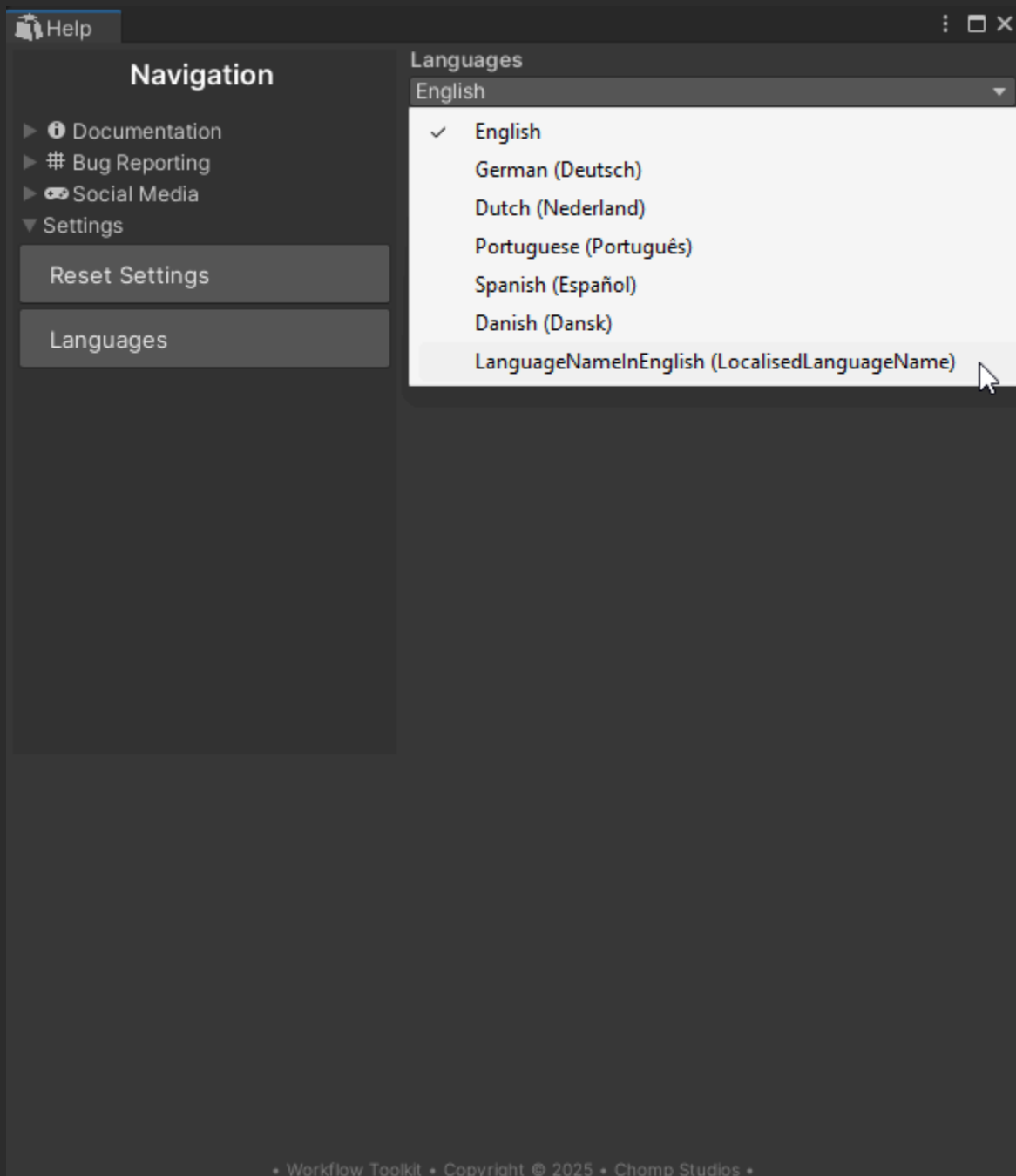
```
case LanguageCode.GERMAN:
    return "de-DE";
```

```
58         public static string GetLanguageFileName(LanguageCode languageCode) {
59             switch (languageCode) {
60                 case LanguageCode.ENGLISH:
61                     return "en-GB";
62                 case LanguageCode.GERMAN:
63                     return "de-DE";
64                 case LanguageCode.DUTCH:
65                     return "nl-NL";
66                 case LanguageCode.PORTUGUESE:
67                     return "pt-PT";
68                 case LanguageCode.SPANISH:
69                     return "es-MX";
70                 case LanguageCode.DANISH:
71                     return "da-DK";
72                 case LanguageCode.TUTORIAL_LANGUAGE:
73                     return "tut-TUT";
74                 default:
75                     return "en-GB";
76             }
77         }
```

# Adding Languages

## All Done

All done! Now go to the “Help” Page in Workflow Toolkit, and your language (as well as the translations) should all be readily available.



# Tools

## Post Build Actions

The Post-Build Actions tool can automatically do a few different things; after your project has been built, it will compress the builds into a .zip file (while you also have the option to add more files, like a “Credits” file or your Game’s Soundtrack to the zip file.

## Script Counter

This tool is similar to a “word count” in document editing software; it will be able to count different types of scripts, text files, assets, and models your project has, you will be able to choose which directories it targets, and there will be settings, such as being able to copy/paste your script names, sort the list as alphabetically, see file size of scripts, and total file sizes and much more - this is very handy if you are writing documentation for your games.

## Notes

Adds a small window to make quick reference notes while programming; this tool will be influential in late-night coding sessions.

## Fancy Screenshots

Screenshot your game quickly; it is also customisable, for example, borders.

# Tools - Post Build Actions

## Player Settings

**Development Build** - This makes the built game a Development Build.

**Allow Debugging** - Allow Debugging when using Development Build.

**Clean Build Cache** - Completely rebuilds all scripts and files.

**Build Strict Mode** - Does not build if there is an error.

**Compression** - Modes are None, Lz4 & Lz4HC; they compress game files and assets.

## File Settings

**Add Bonus Files** - Add additional files to the zip build; remember to name the destination folder.

## Naming Settings

**Development Stage** - Select the current development stage; this affects the zip file build.

**File Name Order** - Choose in which order you want the file name to be



**Timestamp** - Select if you would like a Timestamp at the end of the build name, and choose which type.

## Tools - Notes

### Saving Settings

**Save Notes** - Enables Saving for your notes

**Auto Save** - Automatically saves your notes.

### File Settings

**File Type** - Save your notes as .txt or .json

# Tools - Asset Counter

## Display Settings

***Display Asset Names*** - visually show script names.

***Include Hidden Files*** - show files that have been hidden inside of your file settings.

***Sort Alphabetically*** - Sort assets/scripts in alphabetical order

***Show File Sizes*** - Show the size of your files.

# Tools - Fancy Screenshots

## Naming Settings

**Select Naming Type** - Choose which type of naming you'd like

## Customisation Settings

**Apply Border** - enable if you'd like to choose a border, and decide the size and colour.

**Apply Resolution Multiplier** - makes the screenshot bigger.

# Future Plans

## Spell Checker

I plan to make a spell-checking tool that hooks onto all of the text boxes inside of Unity; it will highlight and make suggestions based on your text and tell you if you have spelt something wrong.

## Unused Asset Remover

I plan to make an Unused Asset Removing tool that strips your project of unused scripts, assets, and files. This makes build files much smaller.

# Version Archive

Vers	Download Links
v 0.3.0	Currently Unavailable
v 0.4.0	Currently Unavailable
v 0.5.0	Currently Unavailable
v 0.6.0	Currently Unavailable
v 0.7.0	Currently Unavailable
v 0.8.0	Currently Unavailable
v 0.9.0	Currently Unavailable
v 1.0.0	Currently Unavailable

# Join Our Community

Currently Unavailable, as the server is being designed.