

IDOC documentation

Version 3.0.0

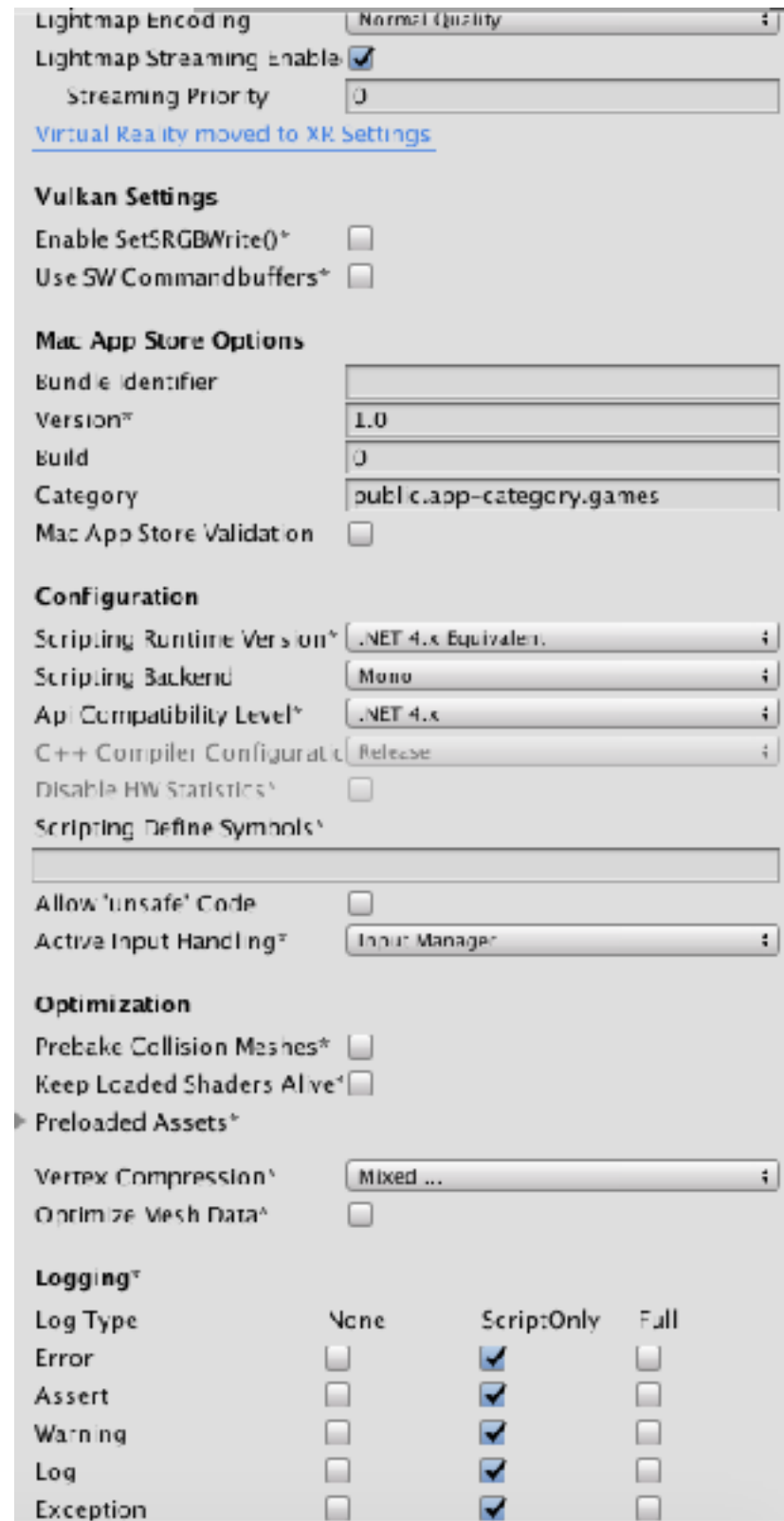
Mamadou Cisse : mciissee@gmail.com

Table of contents

IMPORTANT	3
WHAT IS IT?	4
WHY USE IDOC?	4
WHAT'S IN THE PACKAGE?	5
• IDOC/DII	5
• IDOC/Example	5
• IDOC/Scripts	5
• IDOC/Templates	5
GETTING STARTED	6
• Check if IDOC support your programming language	6
• Setup the editor	6
• Understand the output documents	8
HTML OUTPUT	8
DOCUMENTING YOUR PROJECT	9
• Xml Tags	9
• Special Comments	10
• Guide pages	10
FINAL WORDS	11

IMPORTANT

THE PLUGIN WORKS ONLY IN .NET4 SO ACTIVATE THIS OPTION IN UNITY EDITOR



WHAT IS IT?

IDOC was created to automate the process of project documentation. It allows you to create, publish and maintain professional quality documentation for C#, Java, and C in the current version.

The plugin is fully integrated with Unity Editor thanks to an editor window that allow to set up a project and build the documentation.

I am always open to hearing new ideas for improvements or suggestions and of any problems that you might encounter while using IDOC plugin.

You can email me any time at mciissee@gmail.com and I will respond shortly.

WHY USE IDOC?

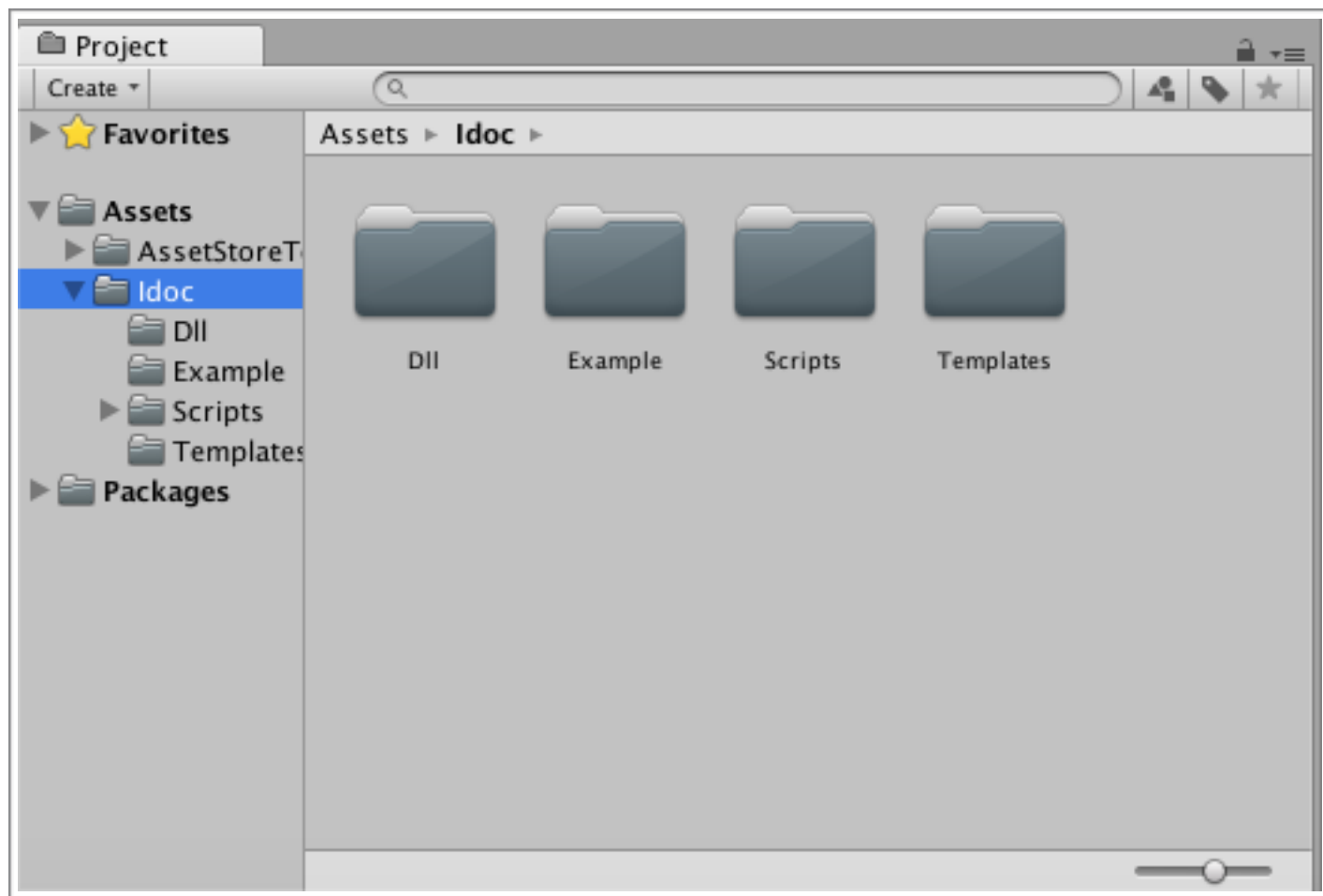
Developing proper documentation, particularly for complex project is normally a time consuming, costly and skills intensive task.

Documentation during development can be considered as an optional task for developers. Developers often have the tendency to start documenting their project at the end of development, which can leave most projects poorly documented.

With IDOC, documentation can be automatically produced throughout design and development without requiring investment or developer resources, providing development teams with an accurate and up to date reference and allowing new developers to jump the learning curve of new components and schemas. IDOC makes producing documentation a natural and productive activity for developers and technical writers alike.

- ✓ Fast, accurate, professional quality documentation.
- ✓ Automatically uses source content where available.
- ✓ Customizable HTML based template.
- ✓ Generate documentation to HTML including a full search engine.
- ✓ Generate HTML for user manual content.

WHAT'S IN THE PACKAGE?



- **IDOC/Dll**

Contains the assemblies that hold the core code of the plugin.

- **IDOC/Example**

Contains an example documented source code for each supported language.

- **IDOC/Scripts**

Contains the editor scripts that interact with the assemblies.

- **IDOC/Templates**

Contains Html templates projects.

GETTING STARTED

IDOC plugin is a template based documentation tool.

The plugin uses html, CSS, and JavaScript file template to build the documentation

You are free to changes the templates to customize your output

The default template shared with the project produces a Single Page Application thanks to JQuery and Handlebar frameworks so you need to code a little if you want to customize the default template. The source codes are fully commented to made things easy for you.

- Check if IDOC support your programming language

First, check that your programming language is recognized by IDOC. These languages are supported by default: **C#**, **Java** and **C**

- Setup the editor

Open the editor by clicking on the tab “**Tools/IDOC Editor**” inside Unity Editor.

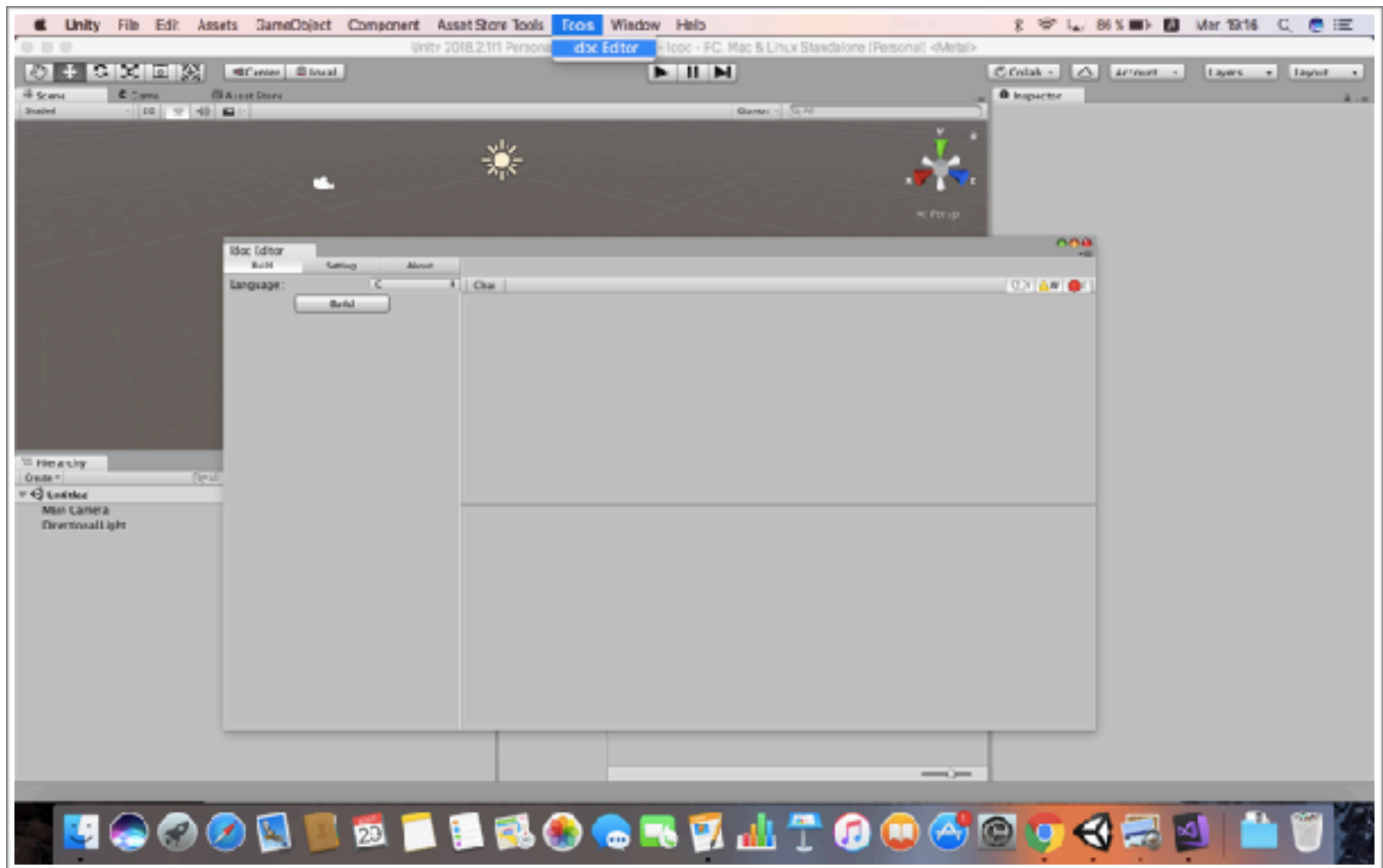
The editor is separated in 3 tabs :

- **Build**

This is the main tab of the editor, it allows you to run IDOC program

The program works in a separated thread so the editor is not blocked during the process but **do not run the program when Unity is in play mode or enter in play mode while the program is running.**

- **Setting**



Use this tab to customize the options of the program and setup the input and output folders.

- **About**

Show informations about the plugin

When you click on the button Build the program search the files with the extension specified in the tab below depending on the selected language.

Language	Extension
C#	.cs
C	.h
Java	.java

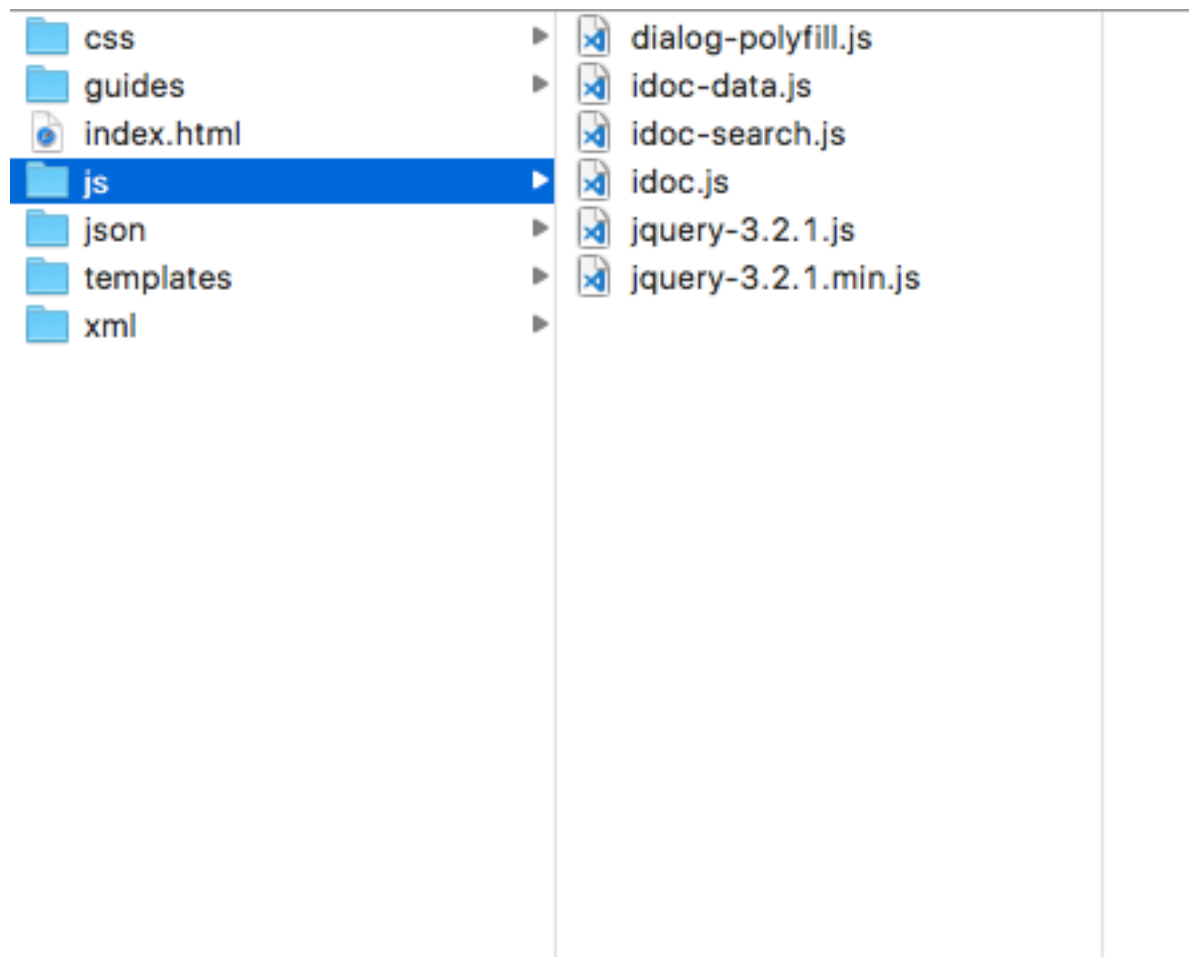
- Understand the output documents

The program produces 3 files in the output path that you specify.

For example if your path is 'Output', the program generated 3 folders at the given path if needed

- **Output/json** this folder will contain a json file representing the members extracted from your source codes. You can copy and paste the content of the file in a web site like <https://jsonformatter.org/> to learn more about the structure of the generated file.
- **Output/xml** this folder will contain an xml document representing the members extracted from your source codes. As for the json document you can explore the content of the xml document to learn more about the structure of the document.
- **Output/js** The purpose of the json and xml documents is to provide you a way to do extra things with the parsed codes and the js file inside this folder 'IDOC-data.js' is what you should use if you want to use the default template project that comes with the plugin.

HTML OUTPUT



Once the program generate the output documents specified in the second above, go to the folder **IDoc/Templates** and copy one of the template projects inside the folder where you want outside of Unity Editor.

Then replace the file **IDOC-data.js** inside **/js** folder by the new one generated generated by the program an open the file index.html in any browser and all .js.txt extensions by .js.

DOCUMENTING YOUR PROJECT

- Xml Tags

IDOC parse only XML documentation comments use triple forward slashes (`///`) and an XML formatted comment body used by default with C# langage . For example:

```
///<summary>This tag is used to resume the function</summary>
///<param name="arg">
/// This tag is used to explain the role of the parameter 'arg'
/// </param>
///<returns>This is an example of function return value comment</returns>
public int TestFunction(int arg){
    return 0;
}
```

The plugin parses only the same xml tags specified here excepts the tags `<include>`, `<para>`, `<permission>` and `<list>`.

For each entity in the code there are three (or in some cases more) types of tags, which together form the documentation for that entity:

- `<summary>`

This tag should be used to describes an entity.

- `<remarks>`

This tag should be used to add extra information to an entity.

- `<example>`

This tag should be used to specify an example of use of an entity.

The two lasts are optional. For methods there is also other type of tags, as the tags:

- `<param>`
This tag should be used to describe a parameter of a method
- `<typeparam>`
This tag should be used to describe a generic parameter of a method
- `<returns>`
This tag should be used to describe a the return value of a method
- `<exceptions>`
This tag should be used to describe the exceptions of a method
Having more than one `<summary>`, `<remarks>` and `<example>` tag is not allowed.

Each of the tags `<summary>`, `<remarks>` and `<example>` create a collapsible section in the output html documentation page.

In addition to the standard tags of IDOC plugin, you are free to use any other html tag and CSS class but it is not recommended.

• Special Comments

IDOC comes with 2 special comments in the current version.

- **///`#IGNORE`** place this comment before a member like a class or a method to exclude it from the documentation. If you exclude a member like a class, you exclude all of its child members (fields, nested classes, method...)
- **///`#EXIT`** place it anywhere you want if you want explicitly end the parsing of a file in a specific line. For example if you place this comment in the line n of your file, the parsing of the file will end at this line.

• Guide pages

IDOC provide you a way to generate custom guide pages.

To generate a custom guide, create a .js file inside the folder guides of the unzipped template and include it in a script tag inside of the file index.html. Open the file quick-start.js if you want to know how a guide file should be structured.

FINAL WORDS

Thanks you if you purchase this asset, if not you can purchase it at <http://u3d.as/Rh4> if you like the asset, please [rate it](#).

Support is available at mciissee@gmail.com.

Make sure you check out our other assets at our [assets store page](#).

Good luck for your projects.

