PDF Documentation Asset

Table of Contents

able of Contents	I
ntroduction	I
cript Reference	1
PDF Documentation	1
Example Folder	1
Example File.cs	1

Introduction

This is an example of a PDF file created using the PDF Documentation asset. To create PDF documentation of your project, follow these steps:

- 1. Import the PDF Documentation asset. The PDF Documentation asset requires C# 7.1 or higher. The project for which the documentation is created does not have to be in the same project as this asset.
- 2. Open the 'PDF Documentation' menu at the top of Unity to open the PDF Documentation window. All settings have tooltips to help clarify their purpose.
- 3. Fill in the Base Settings if necessary. By default, the file name is 'Documentation', the document title is directly the product's name, and the PDF is created in the project root folder.
- 4. Fill in the Script Reference settings if necessary. The default value for the Project root is the 'Assets' folder. The project root can be anywhere, even the directory of another Unity project. Access to read the files is still required. Choose whether you want to include non-public elements, empty methods, or shader files. You can write any keywords in the 'Ignore' fields that will be excluded from the script reference when contained in a file or folder name. Ignored files/folders are separated with commas, and values inside quotation marks (") have to be an exact match.
- 5. Select what you want to see in the Table of Contents of Script Reference. For larger projects, it is advised only to include folders and files. Everything will be in the Script Reference regardless of having a Table of Contents entry.
- 6. Write the introduction text for the document and/or a step-by-step setup guide for using the asset. Unity Asset Store often requires offline documentation, including a step-by-step guide in the format of pdf or rtf in submissions with a set-up process.
- 7. Press the 'Create PDF' button at the bottom right of the window. The bottom of this window and the console will display the result of the process. The document can be edited further with 3rd party PDF modification tools. If any errors occur, the asset cannot parse some files, or you have any feedback, please contact us.

Script Reference

PDF Documentation

Example Folder



Namespaces

ExampleNamespace

```
namespace ExampleNamespace
```

This is the summary comment for ExampleNamespace.

Enumerations

ExampleEnum

```
enum ExampleEnum{
    a = 1,
    b = 2,
    c = 3}
```

This is an enumeration.

Classes

ExampleClass

```
public class ExampleClass
```

This is the summary comment for ExampleClass.

Classes

ExampleChildClass

private class ExampleChildClass

Constructors

ExampleChildClass

```
public ExampleChildClass()
```

Constructor for ExampleChildClass.

Destructors

~ExampleChildClass

```
~ExampleChildClass()
```

Destructor for ExampleChildClass.

Operators

operator ==

```
public static bool operator ==(
    ExampleChildClass a,
    ExampleChildClass b)
```

This is an operator overload for the operator ==.

- a: Left side of the comparison.
- b: Right side of the comparison.

Returns: Boolean value specifying if the given two objects are considered similar.

operator !=

```
public static bool operator !=(
   ExampleChildClass a,
   ExampleChildClass b)
```

This is an operator overload for the operator !=.

- a: Left side of the comparison.
- b: Right side of the comparison.

Returns: Boolean value specifying if the given two objects are considered different.

Variables

exampleInt

```
[SerializeField]
private int exampleInt
```

This is an integer with a 'SerializeField' attribute.

Properties

exampleProperty

```
public int exampleProperty
      { get;
      set; }
```

This is a property.

Methods

ExampleMethod

```
public string ExampleMethod(
  int parameter)
```

This is a method inside the ExampleClass.

parameter: The parameter passed to the method.

Returns: The given parameter as a string.