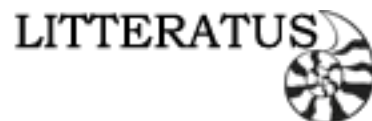




User Manual
Version 2.1.0



Copyright 2014-2015 Litteratus LLC
<http://www.litteratus.net>

Email: contact@litteratus.net

[Introduction](#)

[What is G2U?](#)

[Why do I need this?](#)

[What are some examples of how to effectively use G2U?](#)

[Where does the data come from?](#)

[What spreadsheet formats does G2U support?](#)

[Am I required to use a Google account to use G2U?](#)

[Isn't using my Google account a security risk?](#)

[Who is Litteratus?](#)

[Getting Started](#)

[Setting up G2U for your project](#)

[Signing up for a Google account](#)

[Accessing your Google Drive](#)

[Setting up a Spreadsheet so G2U can parse it](#)

[Ignoring Specific Columns / Rows](#)

[Data Types supported by G2U](#)

[Using Arrays within individual cells](#)

[Using a Spreadsheet for Localization](#)

[Opening G2U](#)

[Logging into your Google account with GoogleFu \(Optional\)](#)

[Accessing Workbooks from G2U](#)

[Workbooks Rollout](#)

[Account Workbooks](#)

[Showing and Hiding Workbooks](#)

[Manual Workbooks](#)

[Uploading Workbooks Using G2U](#)

[Accessing and Exporting Entire Workbooks](#)

[Understanding the different Export Types](#)

[Accessing and Exporting Individual Worksheets](#)

[Changing the Export Settings of a Worksheet](#)

[Legacy Options:](#)

[Whitespace:](#)

[Array Delimiters:](#)

[Export Type Specific Settings](#)

[Object Database Options:](#)

[Global Options:](#)

[Local Options:](#)

[Static Database Options:](#)

[CSV Formatting Options:](#)

[XML Formatting Options:](#)

[JSON Formatting Options:](#)

[NGUI Formatting Options:](#)

[Editing The Worksheets in Unity](#)

[Validating The Worksheets in Unity](#)
[Using Manual Workbooks \(Optional\)](#)
[Uploading Workbooks \(Optional\)](#)
[Accessing your data](#)
[GameObject Database Data Retrieval](#)
[Static Database Data Retrieval](#)
[Retrieving Column Data from a Row](#)
[Samples](#)
[Using the exported data as NGUI Localization files](#)
[Using G2U with Playmaker](#)
[Source Code, Third Party, and Copyright information](#)
[FAQ](#)
[Can I use G2U to access my Google Spreadsheet at runtime?](#)
[What is the difference between the Full version and the Free version?](#)
[Do I have to use my Google account?](#)
[What does G2U do exactly?](#)
[Does this mean my game must contact Google?](#)
[Can I use G2U entirely offline using Excel or other Spreadsheets?](#)
[Can I update my spreadsheet from inside of Unity?](#)
[I am getting a crash! What do I do?](#)

Introduction

What is G2U?

G2U (Formerly GoogleFu) is a tool that works from within the Unity editor to assist in the creation of read-only database-like objects or files, and provide the means to access this data from within Unity. All of the processing and network access is done prior to building the game, so there is no file access, and no network communication needed once the game is built.

Why do I need this?

The use of spreadsheets allows you to better visualize data in relation to other entries. By creating spreadsheets using Google Spreadsheets directly, or by using other popular spreadsheet formats, you are able to design and create a data driven workflow for your game very quickly and easily. G2U provides the mechanism to quickly interpret the spreadsheets, and writes custom classes to either access directly, or through a game object in the scene.

What are some examples of how to effectively use G2U?

One of the most common uses for Spreadsheets is string localization. Google Spreadsheets provide a formula to assist in translating entries from one language to another. This may not be

a perfect translation, but it's enough to get you started. And because Google allows you to collaborate with others seamlessly, providing a localization team the spreadsheet is very easy. Once strings are translated, G2U will be able to import the localized strings into the project, and be ready to access within seconds.

Another use is creating templates for units in your game, loot tables, item stats, enemy stats, asset file mapping, and much more. Imagine creating entities for your game, and being able to visualize and tweak health, speed, and other stats of each unit without editing a single line of code. Or adding all of your sound hooks in the game by ID, and being able to change which sound is being played by simply editing an entry in a spreadsheet.

Where does the data come from?

You provide all of the data needed. There are a few rules to follow when creating a spreadsheet, but it is up to you to provide any type of data you wish. Whether your spreadsheet contains a single column of information, or hundreds, G2U will be able to handle it.

What spreadsheet formats does G2U support?

G2U will only access Google Spreadsheets directly, however Google Spreadsheets is able to import many different types of files, including: .xls, .xlsx, .ods, .csv, .txt, .tsv, .tab

Am I required to use a Google account to use G2U?

Short answer: No.

Long answer: It's recommended, yes. The majority of G2U's functionality revolves around accessing the account of a user, and allowing the user to select individual pages of a workbook to import into the Unity project. While you are able to manually add public, published workbooks and pull the data from them, this would mean that anyone who knew where to find your workbooks could access them. By providing G2U with your account information, it is able to access your private workbooks.

Isn't using my Google account a security risk?

No. G2U utilizes OAuth2, and no confidential information is ever entered into or stored by G2U.

Who is Litteratus?

Litteratus is a small independent studio consisting of two industry veterans. We have been professionally employed in the Video Games / Military Simulation industries for over 8 years, and Litteratus is our hobby. Together we have credits on over 15 different games for PC, Nintendo DS, and PSP, and are currently working in the Rendering and Visualization field. G2U is a product of our hobby work, and a tool that we developed to assist in our own creations. We decided to productize it as we believe it would be useful to anyone, not just Litteratus itself.

Getting Started

Setting up G2U for your project

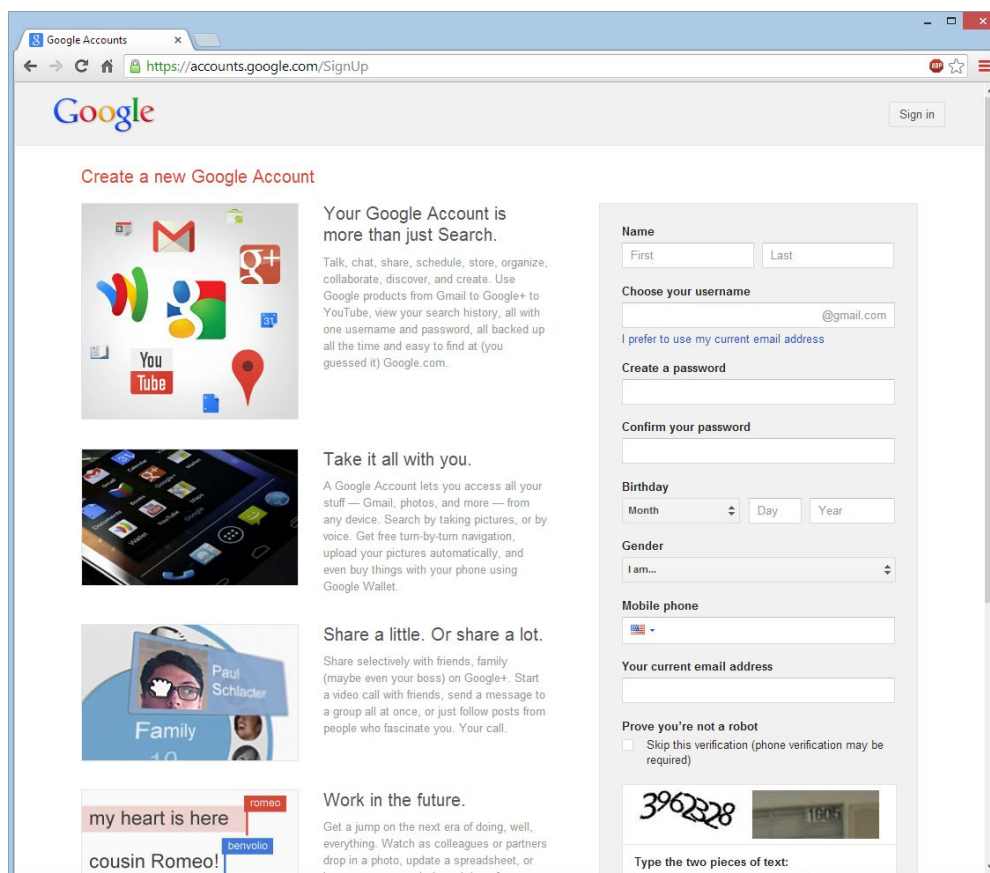
Signing up for a Google account

Whether you decide to provide G2U with your authentication or not, this step is mandatory. You must have a Google account in order to use G2U. If you already have a Google account, and wish to use this account with G2U, you may skip to the next step.

You can create a new Google account at the following website:

[Google Account Creation](https://accounts.google.com/SignUp)

Once you have navigated there using the web browser of your choice, you will need to create an account. Fill in the following form with your account information.

A screenshot of a web browser window showing the Google Accounts sign-up page. The browser's address bar displays 'https://accounts.google.com/SignUp'. The page features the Google logo at the top left and a 'Sign in' button at the top right. The main heading is 'Create a new Google Account'. Below this, there are four promotional images with text: 1. 'Your Google Account is more than just Search.' with icons for Gmail, Google+, YouTube, and Maps. 2. 'Take it all with you.' with an image of a smartphone. 3. 'Share a little. Or share a lot.' with an image of a person's face and the text 'Paul Schlechter Family'. 4. 'Work in the future.' with an image of a person's face and the text 'my heart is here cousin Romeo!'. To the right of these images is a form for creating a new account. The form includes fields for 'Name' (First and Last), 'Choose your username' (with a dropdown for '@gmail.com'), 'Create a password' and 'Confirm your password', 'Birthday' (Month, Day, Year), 'Gender' (I am...), 'Mobile phone' (with a country code dropdown), 'Your current email address', and a 'Prove you're not a robot' section with a checkbox for 'Skip this verification' and a CAPTCHA image showing the numbers '39628' and '1604'. Below the CAPTCHA is a text input field labeled 'Type the two pieces of text:'.

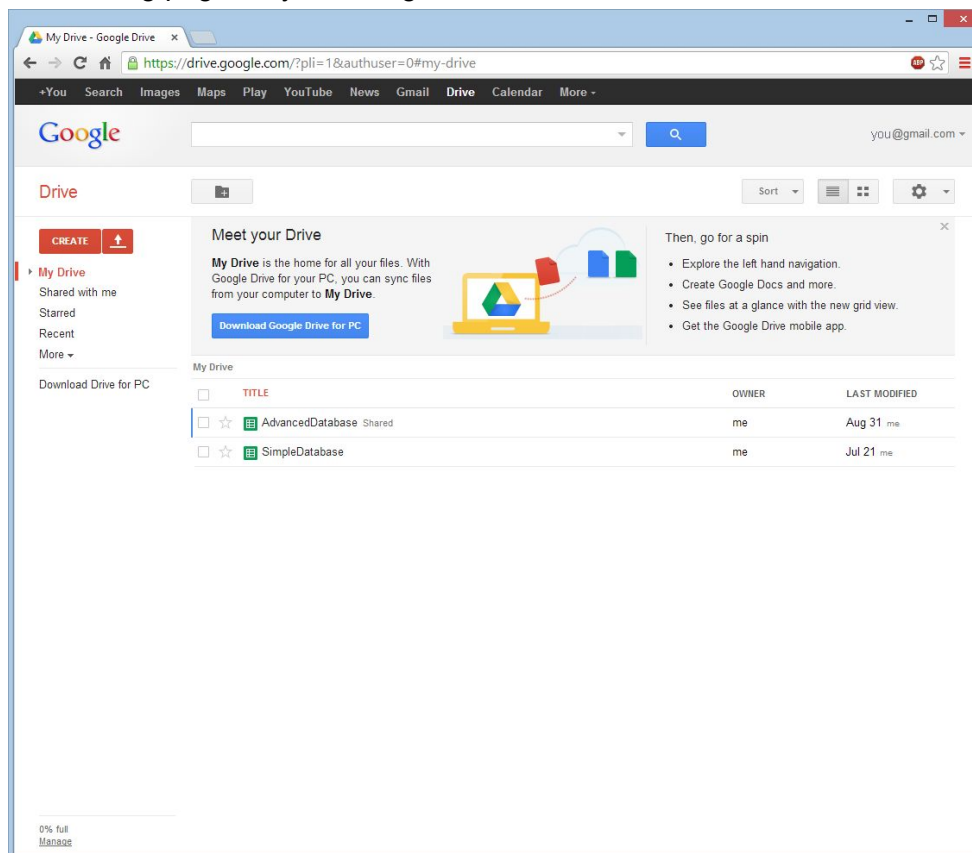
Accessing your Google Drive

Once you have created your new account, you will be able to sign into your Google Drive. This is where all of your Google Spreadsheets will be located. The online access is provided free of charge and you are given a few gigabytes worth of space to save all of your work. It is cloud based and can be accessed anywhere, from any computer with access to Google, provided that you authenticate with Google and log in to your Google Drive.

Your Google Drive can be accessed from here:

[Your Google Drive](#)

The landing page for your Google Drive should look similar to this:



Setting up a Spreadsheet so G2U can parse it

Spreadsheets have 2 basic column types, ID Columns and Data Columns. The values in ID Columns will be the way that G2U indexes the data for retrieval. The values in Data Columns is what will be retrieved when accessing the database.

In addition to columns, there are 3 Row types, Header, Data Type, and Data rows.

- The first row in the spreadsheet will always be the Header row, this contains the variable names of your data.
- The second row is the Data Type row, and is completely optional. If your spreadsheet contains a Data Type row, the final data classes will contain strongly typed values of the types specified, otherwise they will be simple strings and it will be up to you to convert them if necessary.
- The remaining rows are the Data rows, and contain all of your data entries into the spreadsheet.

The diagram illustrates a spreadsheet layout with the following components:

- Row Headers:** A blue box labeled "Row Headers" pointing to the first column (A).
- Col Headers:** A green box labeled "Col Headers" pointing to the first row (1).
- Type Row:** An orange box labeled "Type Row" pointing to the second row (2).
- Data Cells:** A yellow box labeled "Data Cells" pointing to the data rows (3-6).

	A	B	C	D	E
1	ID	Name	Level	HitPoints	Armor
2	TYPE	string	int	float	int
3	ID_Goblin	Goblin	1	1.5	10
4	ID_Orc	Orc	2	3	15
5	ID_Ogre	Ogre	3	4.5	20
6	ID_Troll	Troll	4	6	25

Ignoring Specific Columns / Rows

You can tell G2U to ignore an entire column or row by using the keyword VOID as the column header or row id. Alternatively, if you are using the Data Type row, you can use the keyword VOID or IGNORE to skip that column. Either way is fine, but using the Data Type method will only work if exporting to an Object Database or Static Database.

Data Types supported by G2U

The following Data Types are directly supported by G2U:

Type	Description	Good Example	Bad Example
float	Floating point number designated by a whole number or a whole number followed by a period and then another whole number. using the character 'f' after the number will cause an error.	3.14159 -1.234 3	3.14159f cat 0.1.2
int	32 bit signed whole number Range { -2147483648 to 2147483647 }	1 -512	-21474836499 3.23
unsigned int	32 bit unsigned whole number Range { 0 to 4294967295 }	3 512	-1 false
boolean	Value that represents true or false	true 1	1.0 c
bool	Same as boolean, shorthand	false 0	3 {
byte	An 8 bit signed integer value Valid range { -128 to 127 }	a 33	aa -9000
string	A series, or array of Characters, basically anything goes	Hello -123	It's hard to break this one..
vector2	An array of two floats delimited by a comma	1.0,2.0 3.0, 4.5	2.3f, 3.4f 1,2,3 6.9
vector3	An array of three floats delimited by a comma	1.0,2.0,3.0 9,8,7	blue, 32 1.0
vector	Same as a vector3, shorthand	3, 6.5, 10	99,bottle,beer
color	Array of 3 float values (Red, Green, Blue) cast as a Color inside of Unity. Valid range for colors { 0.0 - 1.0 }	0.5,0.1,0.2	255,255,255
color32	Array of 3 unsigned character values (Red, Green, Blue) cast as a Color32 inside of Unity. Valid range for colors { 0 - 255 }	255,255,255	1.0,1.0,1.0
quaternion	Array of 4 float values cast as a Quaternion inside of Unity.	1,2,3,4	r,g,b,a
quat	Same as a quaternion, shorthand	1.0,2.0,3.0,4.0	1,2,,

But what about { insert another common data type here, such as Short }? Why doesn't G2U support that? Unity has issues serializing some data in MonoBehaviors. Even common data types such as char are not supported.

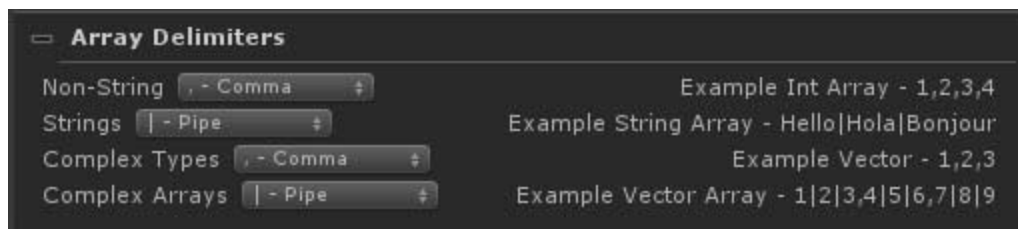
We only support data types that fit the following criteria:

- * Common
- * Generically useful
- * Easily edited inside of Google Spreadsheets
- * MonoBehaviour is able to serialize natively

This rules out some data types such as Matrix. A Matrix is a common data type, but since Unity uses quaternions almost exclusively when describing rotations and it's very difficult to work out what a matrix is doing from inside a single cell of a worksheet, we do not support it.

Using Arrays within individual cells

G2U allows a smaller subset of the available types to be converted to arrays of that type (Actually List<T> to allow for further expansion) when importing into Unity. Arrays will need to be delimited by a character that you choose inside of G2U. Typically the comma is used, however in the case of Strings, another delimiter may be necessary. Choose the delimiter in the Tools menu before importing the data from Google. The delimiter will need to be consistent across all arrays in the Spreadsheet.



Array Delimiters apply to all arrays with simple values:

Byte Array
Boolean Array
Int Array
Float Array

The default delimiters for Simple Values are comma and space “ , ”

Example -

GoogleFu_ID	TestSimple Integer
GoogleFu_Type	Int Array
ID_Test01	1, 2, 3, 4, 5

String Array Delimiters only affect String Arrays

The default delimiter for String Arrays is the pipe “|”

Example -

GoogleFu_ID	TestSimple string array
GoogleFu_Type	String Array
ID_Test02	Hello Hola Bonjour

Complex Type Delimiters apply to Complex Values:

Vector2

Vector3

Color

Color32

Quaternion

The default delimiters for Complex Values are the comma and space “ , ”

Example -

GoogleFu_ID	TestSimple Vector3
GoogleFu_Type	Vector3
ID_Test02	1.0, 2.0, 3.0

Complex Type Array Delimiters apply to arrays of Complex Values:

Vector2 array

Vector3 array

Color array

Color32 array

Quaternion array

The default delimiter for Complex Value Arrays is the pipe “|”

Example -

GoogleFu_ID	TestSimple Vector3 Array
GoogleFu_Type	Vector3 array
ID_Test03	1.0, 2.0, 3.0 1.1, 2.1, 3.1

The following Data Array Types are directly supported by GoogleFu:

Type	Description	Example
float array	Array of delimited floats	1.0, 2.3, 6.4
int array	Array of delimited 32 bit signed integers	-1, -2, 3
boolean array	Array of delimited boolean values	true, true, false
bool array	Same as boolean array, shorthand	0, 0, 1
byte array	Array of delimited 8 bit signed bytes	-2, -4, 88
string array	Array of delimited strings	yes no maybe
Vector2 array	Array of delimited Vector2's	1.0,2.0 2.0,3.0
Vector3 array	Array of delimited Vector3's	1.1,2.1,3.1 1.2,2.2,3.2
Vector array	Same as Vector3 array, shorthand	3, 6.5, 10 916, 748, 996
Color array	Array of Color values	0.5,0.1,0.2 0.5,0.1,0.2
Color32 array	Array of Color32 values	123,165,205 221,50,240
Quaternion array	Array of Quaternions (If anybody actually uses this, I want to know. Email me at Contact@Litteratus.net)	good luck

Using a Spreadsheet for Localization

If you are setting up a Localization spreadsheet, you will not need to use the Data Type row, as all of the data is going to be strings anyway. Google has a handy formula for automatically translating cells:

=GoogleTranslate(value, from language id, to language id)

So, using the following example:

STRING_ID	EN	FR
ID_INITIALIZING	Initializing. Please wait ...	Initialisation. S'il vous plaît patienter ...
ID_AUTHENTICATING	Authenticating with Google. Please wait ...	Authentification avec Google. S'il vous plaît patienter ...
ID_IMPORTING_MESSAGE	Importing Database. Please wait ...	Importation de base de données. S'il vous p
ID_SETTINGS	Settings	Réglages
ID_WORKBOOKS	Workbooks	Classeurs
ID_CREDENTIALS	Credentials	Lettres de créance
ID_LANGUAGE	Language	Langue
ID_PATHS	Paths	Chemins
ID_TOOLS	Tools	Outils
ID_HELP	Help	Aide
ID_HELP_MAIN	Main	Principal
ID_HELP_LOCAL	Localization	Localisation
ID_HELP_DB	Database	Base de données
ID_GENERATE_PATHS	Automatically Create Paths	Créer automatiquement les chemins

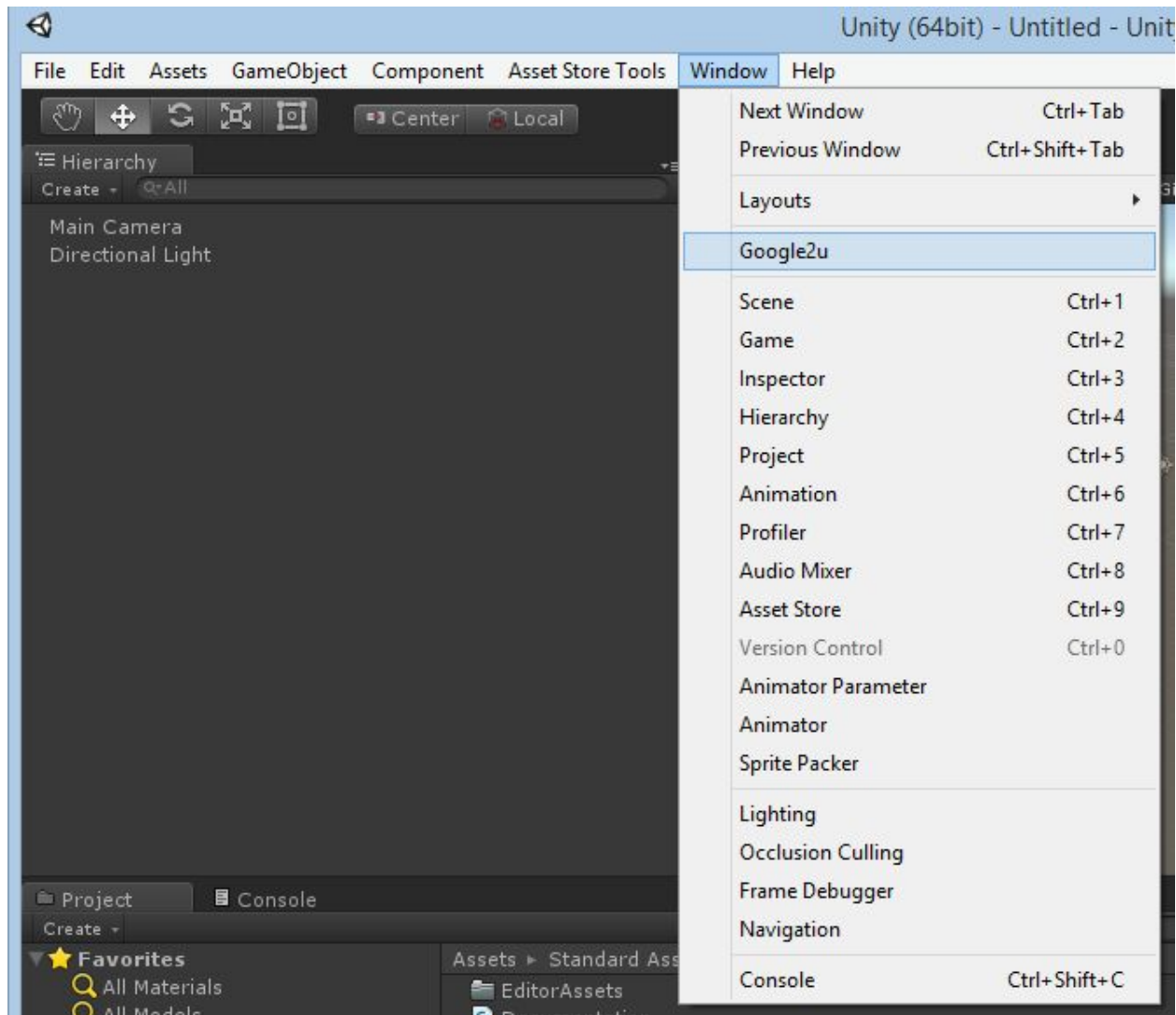
If you want to translate the cell B2 from English to French, you would use the following in Cell C2:

=GoogleTranslate(\$B2,\$B\$1,C\$1)

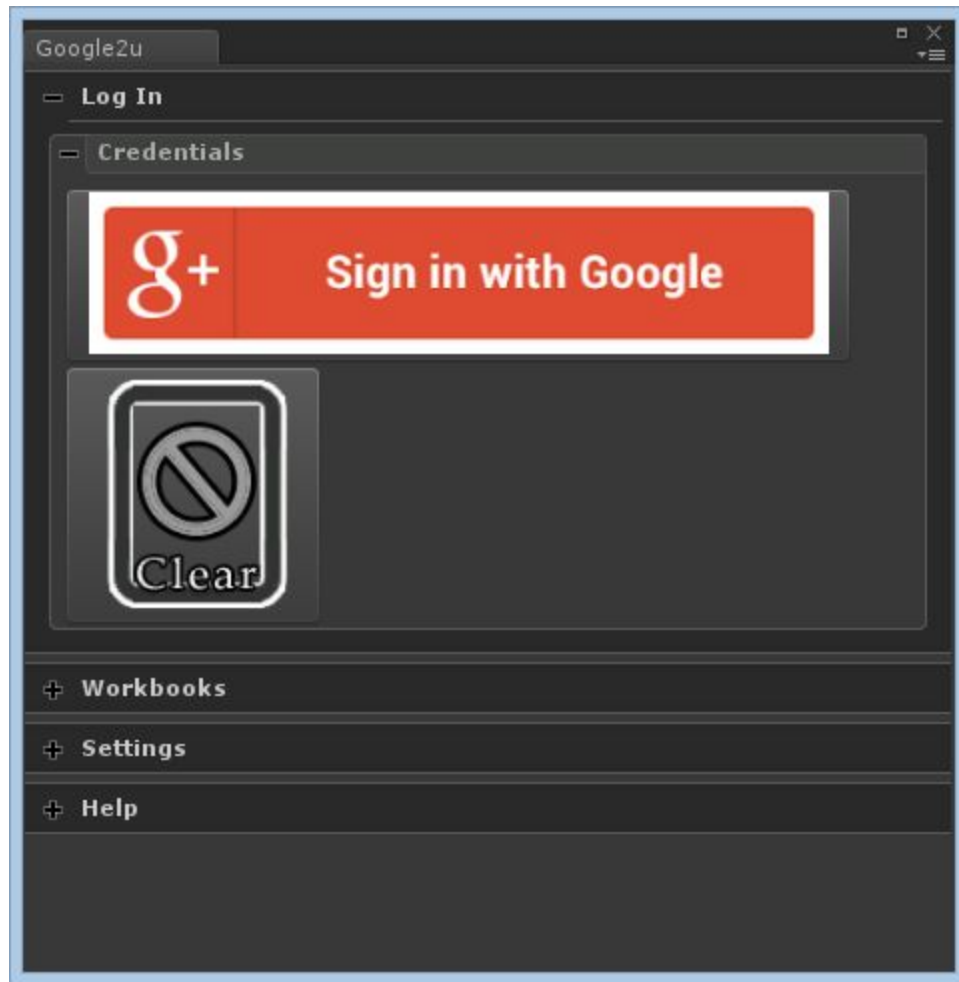
Google Spreadsheet uses the Google Translate functionality internally, and is not a feature of G2U. If you have a large enough string database, and are localizing for many languages, you will find that sometimes Google will stop translating. This is because it only allocates so much time to do translations. A way to resolve this is wait for the sheet to fully load in Google and search for "#N/A" on the page. It will show you cells that have not been fully resolved. If you add or remove a space in the English cell, it will attempt to resolve only the dependent cells. Repeat this process for any unresolved cells.

Opening G2U

When you start G2U for the first time by opening the Unity Editor and selecting Window->Google2U:



You will be presented with the G2U window:

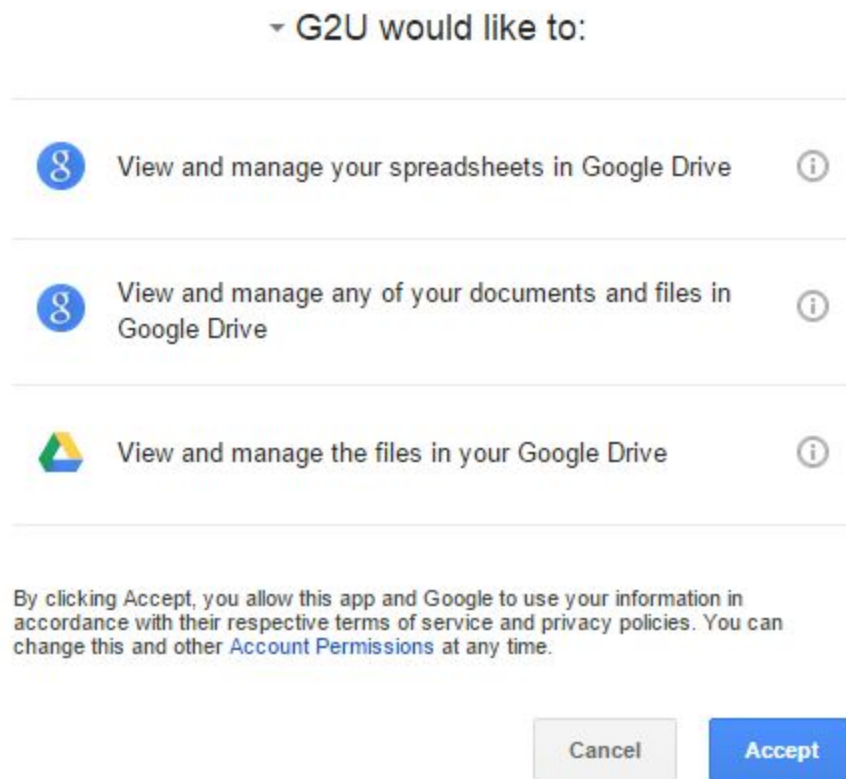


There are four main rollouts:

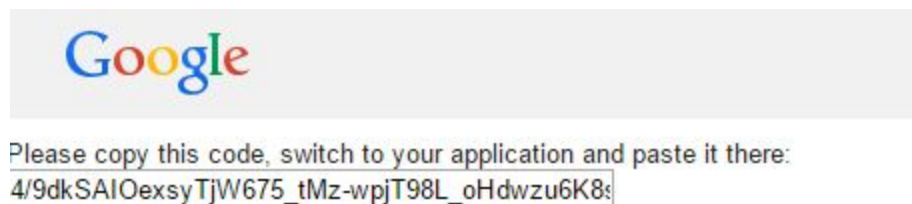
- Log In, for authenticating your Google account to allow G2U to access your Spreadsheets.
- Workbooks, for accessing, editing, and exporting Spreadsheets
- Settings, for general settings regarding G2U
- Help, to access an in-editor version of these documents

Logging into your Google account with GoogleFu (Optional)

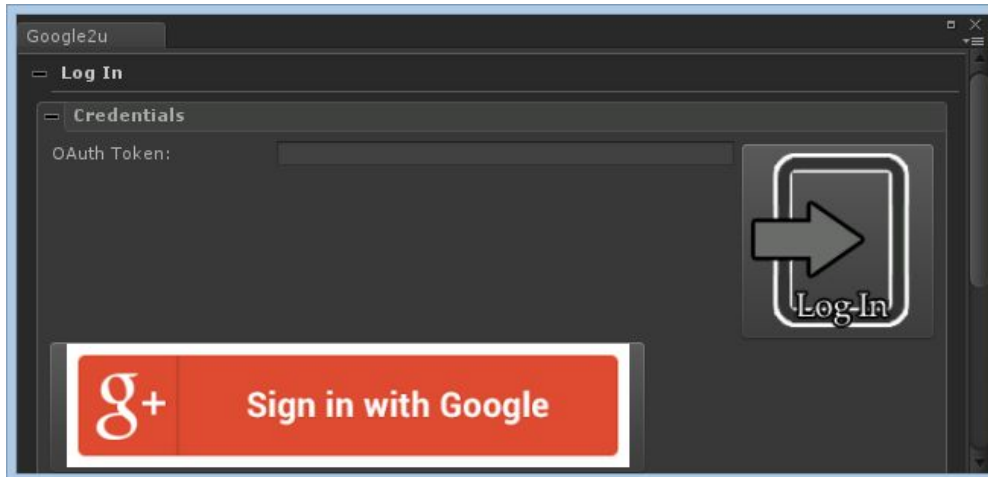
In the Credentials rollout into the Log In section you will find a button to sign in to your Google Account, and another button to clear your saved credentials (In case you need to log into a different account later on). Clicking the “Sign in with Google” button you will be taken to the Google Authentication page, where you will choose a Google account to grant access to.



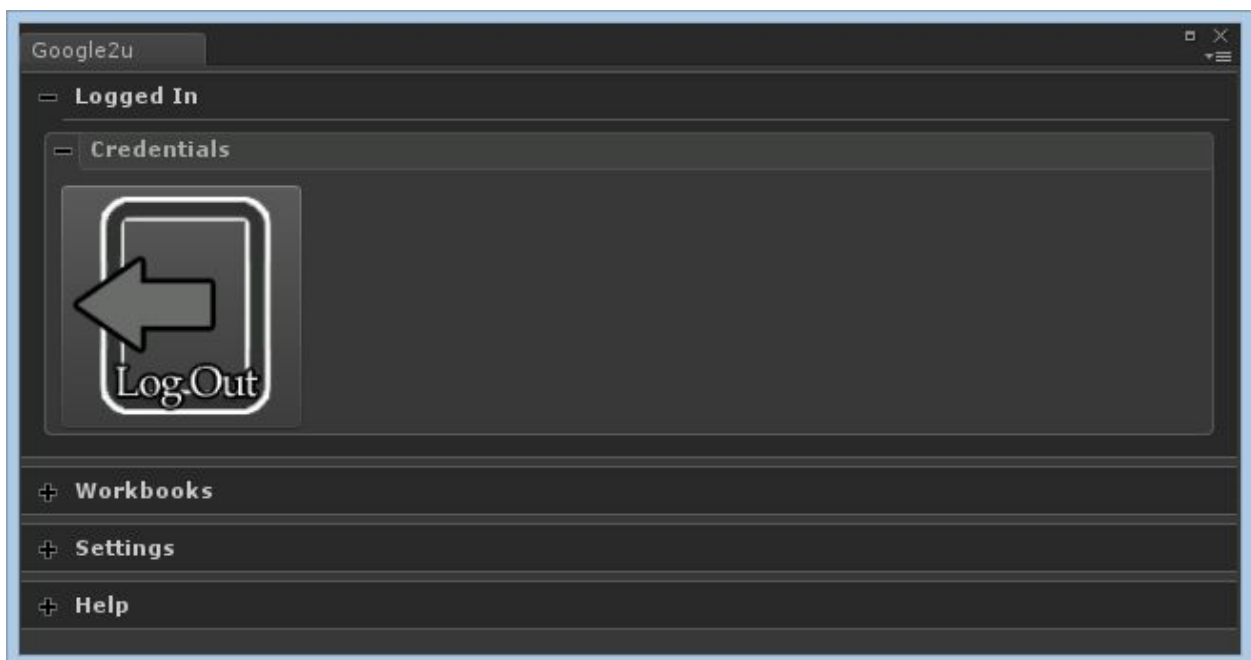
Once you have accepted, you will be presented with an OAuth code.



Copy, and paste the code into the OAuth Token box in G2U and press the Log In button. If you did not receive the token, or closed the window before copying it, you can try again by pressing the “Sign in with Google” button again.

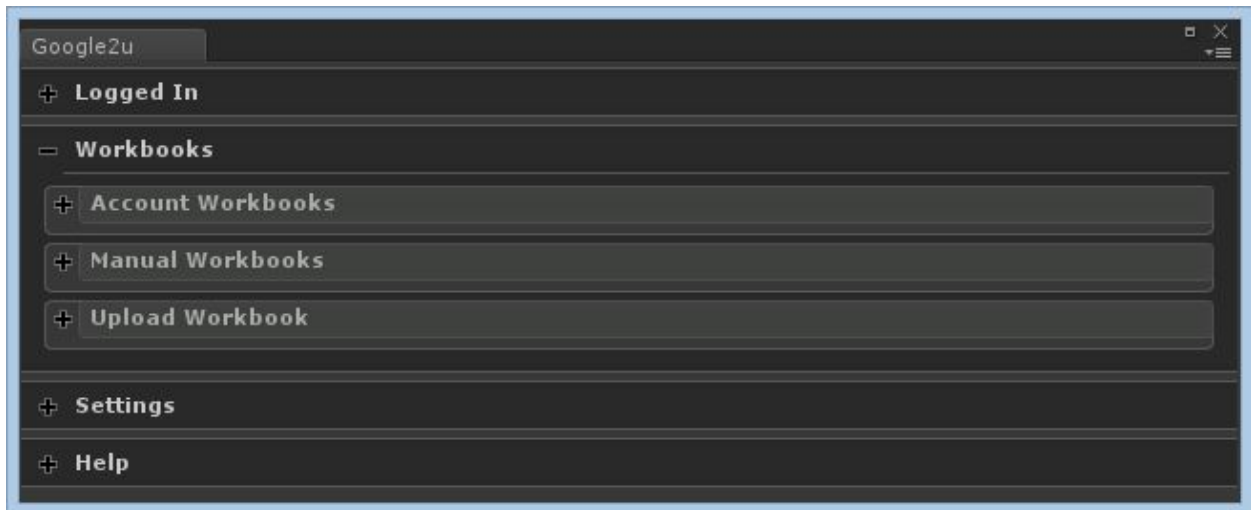


Unless you clear your credentials, the token will be saved, and automatically applied from now on. Once applied, all you should see in the Credentials rollout is the button to Log Out. You can collapse the rollout by pressing the minus sign next to Credentials.



Accessing Workbooks from G2U

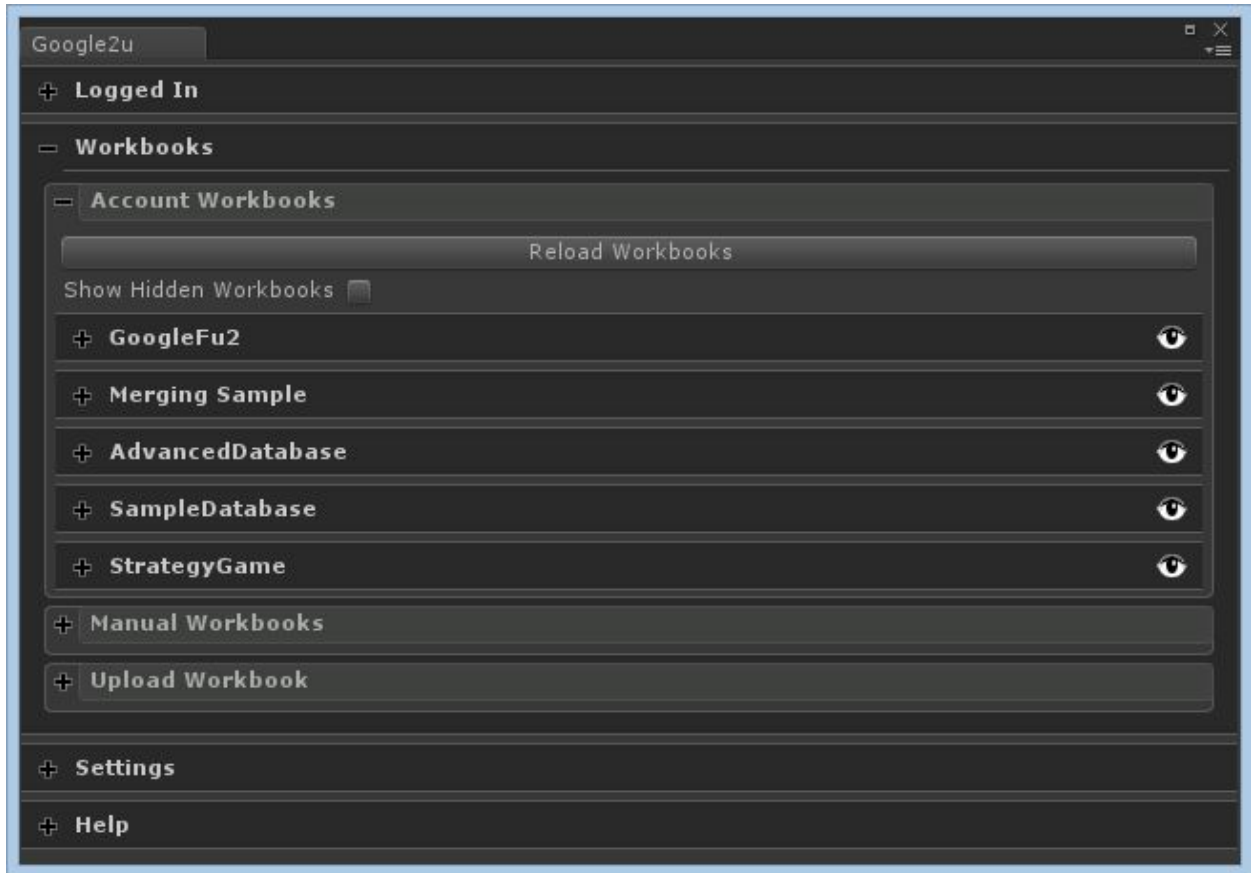
Workbooks Rollout



There are three sections in the Workbooks rollout. Account Workbooks is where you will access the workbooks tied to the account you are currently logged in with. All workbooks that are accessible from this account will be displayed. Manual Workbooks is where you can access workbooks that have been published to the web. Depending on the settings of the workbook, it may be read-only. Upload Workbook is there to assist in the upload of workbooks to your Google Account. It is not required that you use this option to generate workbooks, but you are welcome to use the interface in G2U rather than importing spreadsheets through Google.

Account Workbooks

When you open the Account Workbooks section you will be presented with a list of all of the workbooks associated with your Google account.



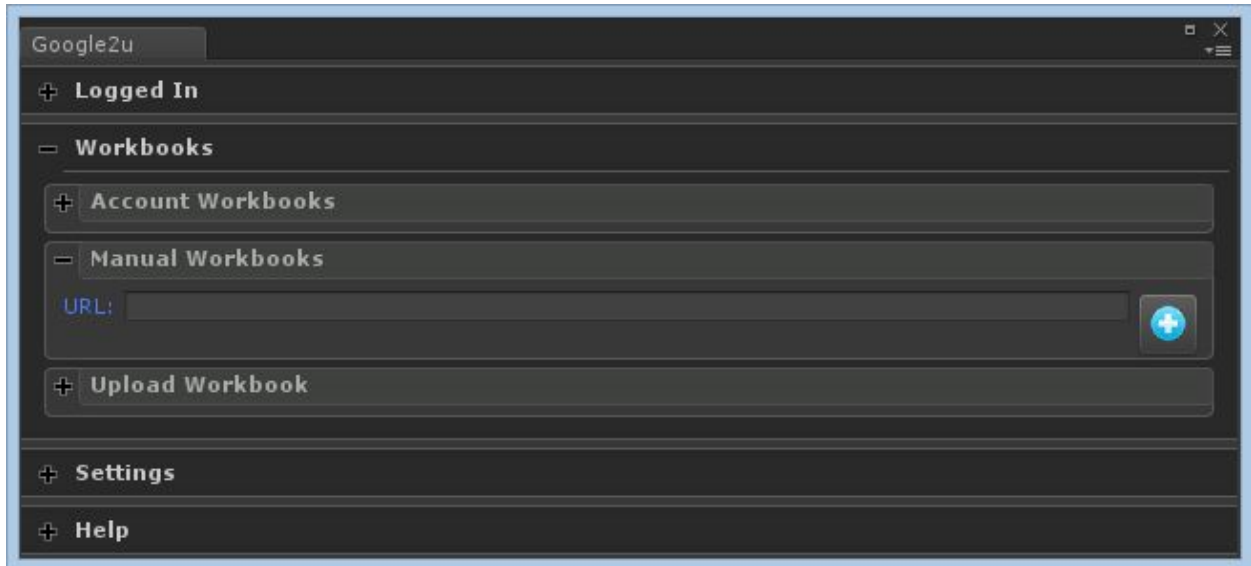
Pressing the Reload Workbooks button will refresh this list. Sometimes this is necessary if you have made changes that haven't automatically synced yet.

Showing and Hiding Workbooks

The Eye icons to the right of the Spreadsheet entries will hide the Spreadsheet from view. If your account contains spreadsheets not relevant to the project, then you can clean up the window by hiding them. If you have hidden a spreadsheet, and later need to reveal it again, simply check the "Show Hidden Workbooks" box, and click the red Eye icon to reveal it again.

Manual Workbooks

If you have a URL for a published workbook, you can enter it into the URL field and press the + button to access it.

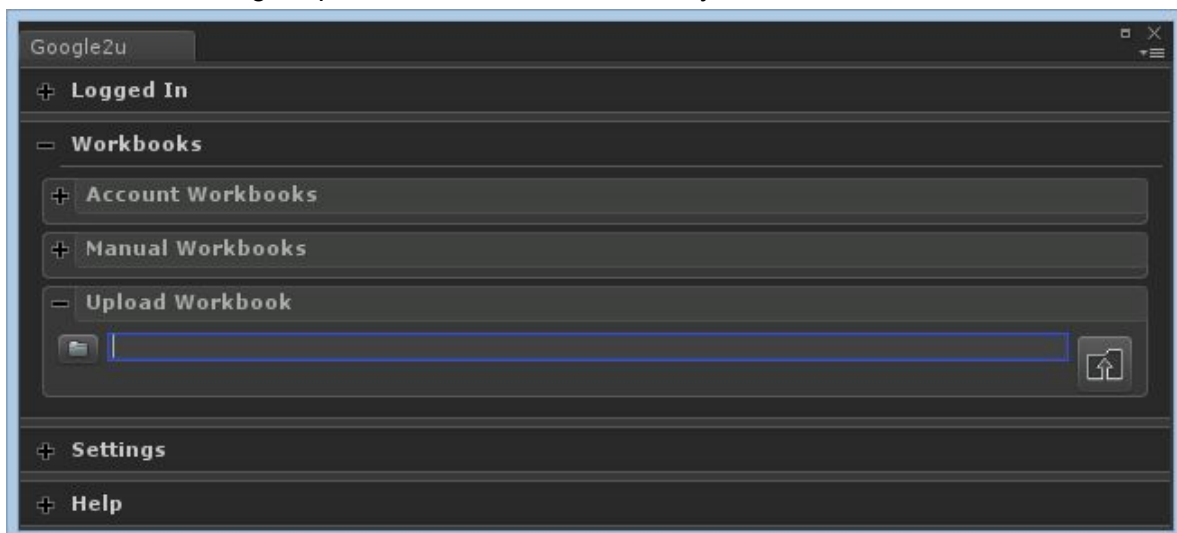


Once a Manual Workbook has been added, it can be removed later.

The URL that you use to access Published workbooks is NOT the same URL that you would use to access it in a Google Drive account. Details on how to obtain the URL will be explained later on in the document under Publishing A Workbook.

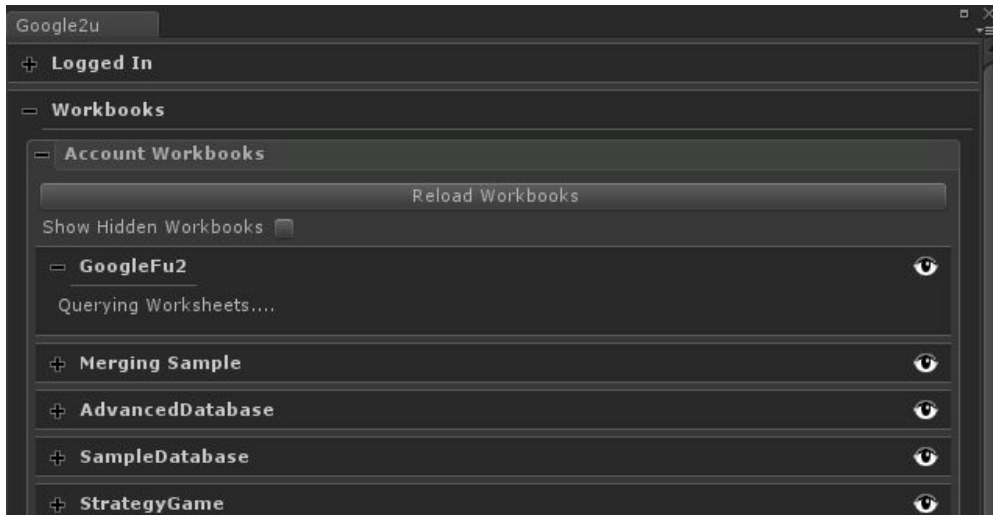
Uploading Workbooks Using G2U

The Upload Workbook section provides a file browser to select a local workbook, such as an Excel Spreadsheet, to upload to Google Docs. Use the folder icon on the left to open a file browser and select the spreadsheet, then use the Upload icon on the right to upload the document to Google. Provided there are no errors with the document, it will be automatically converted to a Google Spreadsheet, and accessible by G2U.

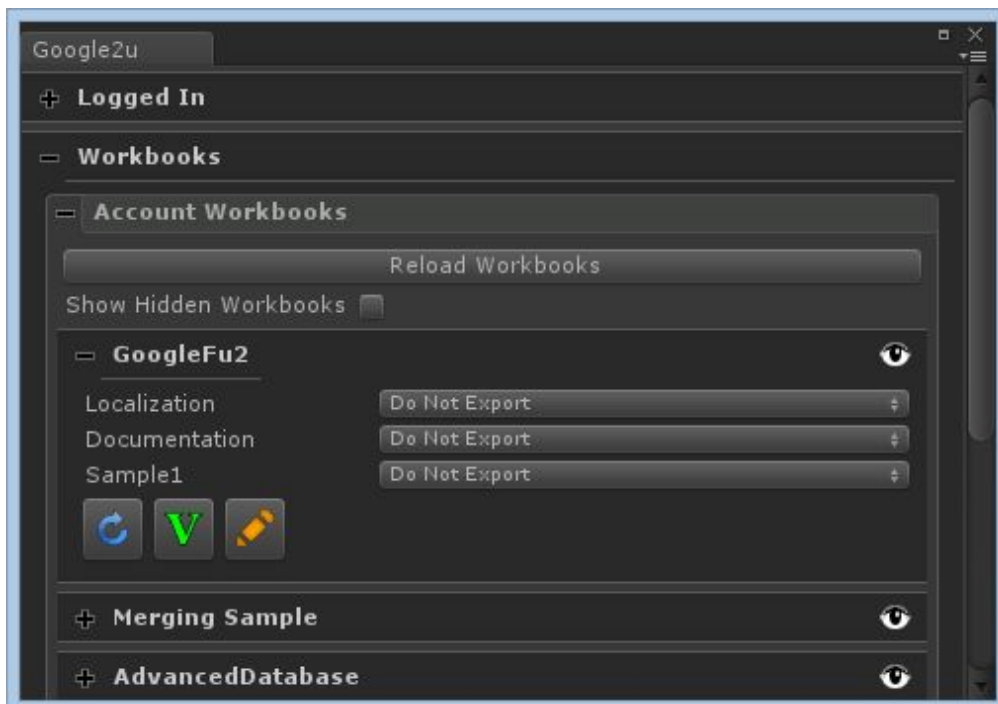


Accessing and Exporting Entire Workbooks

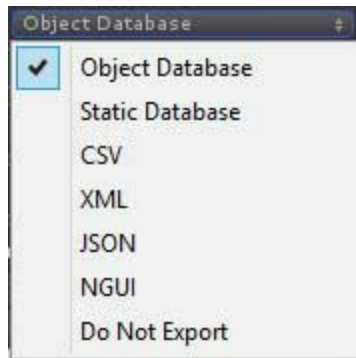
When you expand the workbook that you want to work with, the first thing G2U will do is query your worksheet, and present to you a List View of the worksheets within the workbook.



Once the query is completed, the individual sheets of the workbook will be displayed. By default they are all set to “Do Not Export”. This means when you export the entire workbook, sheets marked this way will not be included.







Each sheet in the workbook has a box that you can use to select the Export Type.




Sheets that are marked with “Do Not Export” are skipped and not processed during the Validation. Specific export options for the sheets can be found in the Edit view of the workbook. More on that later.

The icons at the bottom are:

	Refresh	Reload all of the data in the workbook. It's a good idea to refresh the workbook before exporting to ensure you have the latest data synced up
	Valid Workbook	When this icon is green, it indicates that all of the data in the workbook is in a valid format, and ready to be exported
	Invalid Workbook	When this icon is red, it indicates that there is an error in one of your sheets. Clicking it will open the sheet and highlight the first encountered error.
	Edit Workbook	Clicking this icon will bring up the export settings, and an editor that allows you to make modifications to your spreadsheet through G2U

If all data is valid in the workbook, and at least one sheet is selected to export, then another icon will appear on the bar.

	Export Workbook	Exports all selected sheets in the workbook.
---	-----------------	--

Understanding the different Export Types

There are few choices available for the type of data you wish to export. Depending on the project, type of data in the spreadsheet, or personal preferences your project may favor one type over another. Here is a rundown of the different types available.

Object Database: This mode will collect all of the data in the spreadsheet, and write a custom class to access the data. It will then attach this class as a MonoBehaviour to a Game Object, and insert all of the data into the Game Object itself. This is also the only option that allows Playmaker Action Generation, a process that will make custom tailored Playmaker Actions to assist getting information from your Game Object.

Static Database: This mode will generate a script that contains all of the data in your database, accessible statically through a Singleton-type class. This mode is perfect for Localization, and in fact the Localization in G2U was done using this export type.

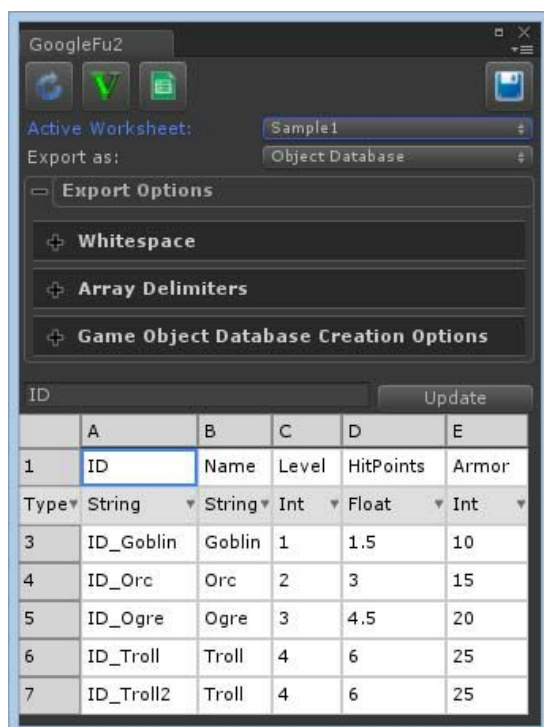
CSV: This mode will generate a flat CSV that contains your data. While Google provides a mechanism for generating CSV data, using this option enforces certain standards on the data, making it more reliable.

XML: Just like CSV, the XML output can be achieved through Google itself, but ignoring rows/columns and the like is only achievable by using our XML exporter. When you export, G2U generates a flat .xml file.

JSON: The JSON export has a few options available that make it ideal for use in games. This mode improves the quality of the exported JSON code by enforcing standards through G2U. When you export, G2U generates a flat .json file.

NGUI: This option is for Localization. It will generate the .CSV file in the format that NGUI is expecting to use the built-in controls without any manual work.

Accessing and Exporting Individual Worksheets








The screenshot shows the GoogleFu2 application window. At the top, there are icons for Google, Unity, and a document. Below these, the 'Active Worksheet' is set to 'Sample1' and 'Export as' is set to 'Object Database'. The 'Export Options' panel is expanded, showing sections for 'Whitespace', 'Array Delimiters', and 'Game Object Database Creation Options'. Below the options, there is a table with columns A through E. The first row is a header with 'ID', 'Name', 'Level', 'HitPoints', and 'Armor'. The second row shows data types: 'String', 'String', 'Int', 'Float', and 'Int'. The following rows contain numerical data for various entities like 'Goblin', 'Orc', 'Ogre', 'Troll', and 'Troll2'.

	A	B	C	D	E
1	ID	Name	Level	HitPoints	Armor
Type	String	String	Int	Float	Int
3	ID_Goblin	Goblin	1	1.5	10
4	ID_Orc	Orc	2	3	15
5	ID_Ogre	Ogre	3	4.5	20
6	ID_Troll	Troll	4	6	25
7	ID_Troll2	Troll	4	6	25

When you click on the Edit button of a workbook in the List View, a new window will pop up, allowing you to modify export settings, and the worksheets themselves. This is called the Full View.

The icons at the top are:

	Refresh	Reload all of the data in the workbook. It's a good idea to refresh the workbook before exporting to ensure you have the latest data synced up
	Valid Workbook	When this icon is green, it indicates that all of the data in the workbook is in a valid format, and ready to be exported
	Invalid Workbook	When this icon is red, it indicates that there is an error in one of your sheets. Clicking it will open the sheet and highlight the first encountered error.
	Edit Workbook	Clicking this icon will open a browser window and connect to your Google Drive for direct editing of the workbook. Some things, such as formula editing, is a lot easier to do through the Google interface.
	Export Worksheet	This will only export the Active Worksheet. It will not export the rest of the worksheets in the workbook, even if they are marked for export. To export the entire workbook, use the button in the List View.

Selecting the Active Worksheet will change the worksheet that is currently being previewed / edited, and selecting the Export Type will change the type of data output that the sheet will exported as.

Changing the Export Settings of a Worksheet

Opening the Export Options rollout will show you the settings for the export process. In most cases these settings are applied globally, and among all worksheets that share the same export type. There are a couple cases where settings are on a Per-Worksheet basis. More on that later.

Legacy Options:

Make Columns Lowercase	In previous versions of G2U (GoogleFu) columns headers were forced to be lowercase. This is no longer enforced, but to ensure backwards compatibility, this option exists. Checking it will force all column headers to be lowercase.
------------------------	---

Whitespace:

<u>Trim Strings</u>	Checking this option will remove all whitespace at the beginning and end of strings.
<u>Trim String Arrays</u>	Checking this option will remove all whitespace at the beginning and end of strings within string arrays

Array Delimiters:

These are the separators that designate a new entry in the array

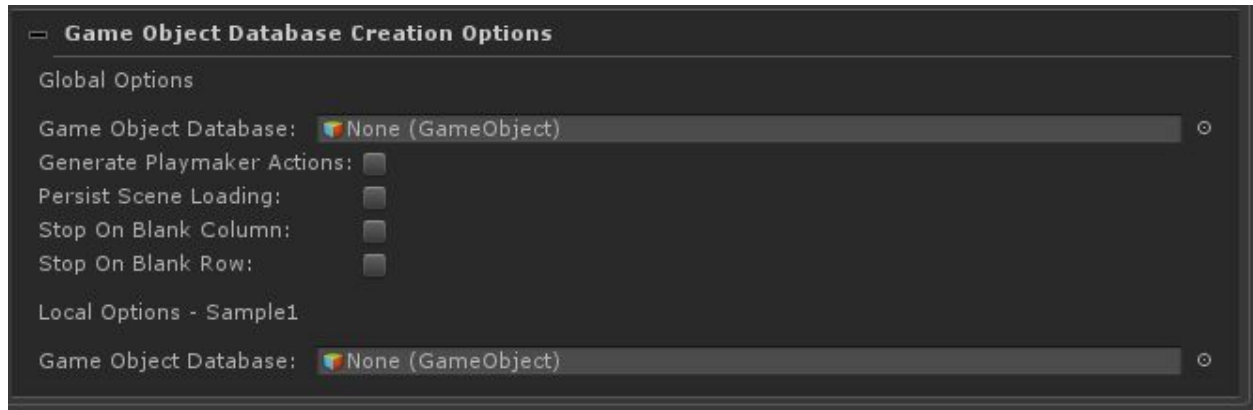
Non-String	Arrays consisting of non-string data
Strings	Arrays of single words
Complex Types	Arrays that represent a single complex type, such as a Vector
Complex Arrays	Arrays that represent many complex types, such as a Vector Array

Export Type Specific Settings

When you select an Export Type for a sheet, another Export Option section will appear specifically for that type.

Object Database Options:

There are two parts to the Object Database Options, one that affects all of the sheets, and then a Local Options that only affects the current sheet.



Global Options:

Game Object Database	A reference to the game object that will host all of the data. This is the game object where all of the scripts will be attached to once the spreadsheet has been exported
Generate Playmaker Actions	Whether or not to generate Playmaker Actions based on the Game Object Database
Persist Scene Loading	Include the DoNotDestroy flag, to prevent the Game Object Database from being deleted when a new scene is loaded
Stop on Blank Column	When the export process reaches a blank column, stop processing any further columns
Stop on Blank Row	When the export process reaches a blank row, stop processing any further rows

Local Options:

Game Object Database	This option will override the Global Game Object Database if it's set. This allows you to select a new object on a per-sheet basis.
----------------------	---

Static Database Options:

Static Database Creation Options

Stop On Blank Column: ☐

Stop On Blank Row: ☐

Stop on Blank Column	When the export process reaches a blank column, stop processing any further columns
Stop on Blank Row	When the export process reaches a blank row, stop processing any further rows

CSV Formatting Options:

CSV Formatting Options

Convert Quotes: ☒


Stop On Blank Column: ☐

Stop On Blank Row: ☒

Escape Line Breaks: ☒

Convert Quotes	Place quotes around every entry in the CSV file
Stop on Blank Column	When the export process reaches a blank column, stop processing any further columns
Stop on Blank Row	When the export process reaches a blank row, stop processing any further rows
Escape Line Breaks	Replace Line Breaks with an escape sequence

Preview CSV:



```
"ID","Name","TestInt","TestFloat","TestBool","TestFloatArray","TestVector3","TestColorArray"
"TYPE","string","int","float","bool","float[]","Vector3","Color[]"
"ID_Goblin","Goblin","1","1.5","TRUE","1.0,2.0,3.0","1.0, 1.1, 1.2","1.0,1.1,1.2,1.3|2.0,2.1,2.2,2.3"
"ID_Orc","Orc","2","3","FALSE","2.0,3.0,4.0","2.0, 2.1, 2.2","1.0,0.1,0.2,0.3|2.0,0.1,0.2,1.0"
"ID_Ogre","Ogre","3","4.5","TRUE","3.0,4.0,5.0","3.0,3.1,3.2","1.0,0.1,0.2,0.3|2.0,0.1,0.2,1.1"
"ID_Troll","Troll","4","6","FALSE","4.0,5.0,6.0","4.0,4.1, 4.2","1.0,0.1,0.2,0.3|2.0,0.1,0.2,1.2"
```

Generate a preview of the CSV output. The Unity window is not designed for very large amounts of text, thus if your CSV output is too long, it will cut off part of the file. Use this preview window to verify that you are getting the output you expect, but use the CSV Export feature to ensure your file is complete.

XML Formatting Options:

XML Formatting Options

Convert Cell Array to String:

Stop On Blank Column:

Stop On Blank Row:

Convert Cell Array to String	Instead of generating tags for entries in arrays, convert arrays to strings.
Stop on Blank Column	When the export process reaches a blank column, stop processing any further columns
Stop on Blank Row	When the export process reaches a blank row, stop processing any further rows

Preview XML File:

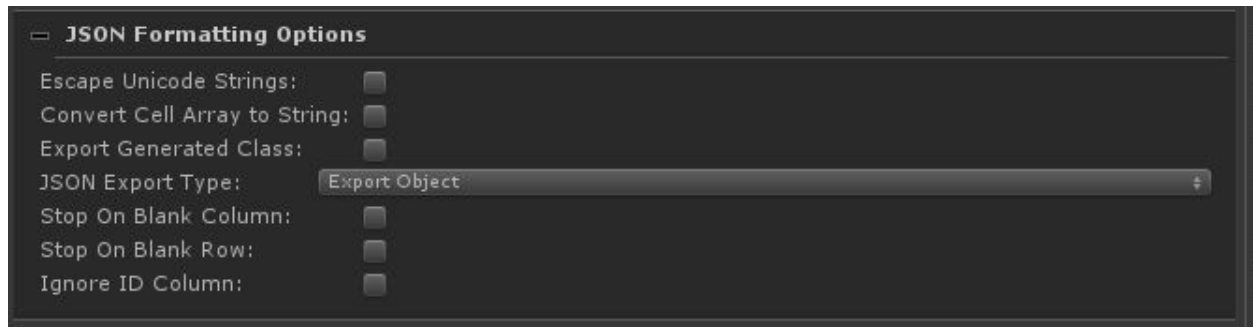
Preview XML File

Generate Preview

```
<Sheets>
  <sheet name="Sample1">
    <row name="ID_Goblin">
      <Name type="string">Goblin</Name>
      <TestInt type="int">1</TestInt>
      <TestFloat type="float">1.5</TestFloat>
      <TestBool type="bool">TRUE</TestBool>
      <TestFloatArray type="float[]">
        <entry>1.0</entry>
        <entry>2.0</entry>
        <entry>3.0</entry>
      </TestFloatArray>
      <TestVector3 type="Vector3">1.0, 1.1, 1.2</TestVector3>
      <TestColorArray type="Color[]">
        <entry>1.0</entry>
        <entry>1.1</entry>
        <entry>1.2</entry>
        <entry>1.3|2.0</entry>
        <entry>2.1</entry>
      </TestColorArray>
    </row>
  </sheet>
</Sheets>
```

Generate a preview of the XML output. The Unity window is not designed for very large amounts of text, thus if your XML output is too long, it will cut off part of the file. Use this preview window to verify that you are getting the output you expect, but use the XML Export feature to ensure your file is complete.

JSON Formatting Options:



JSON Formatting Options

Escape Unicode Strings: ☐

Convert Cell Array to String: ☐

Export Generated Class: ☐

JSON Export Type: Export Object

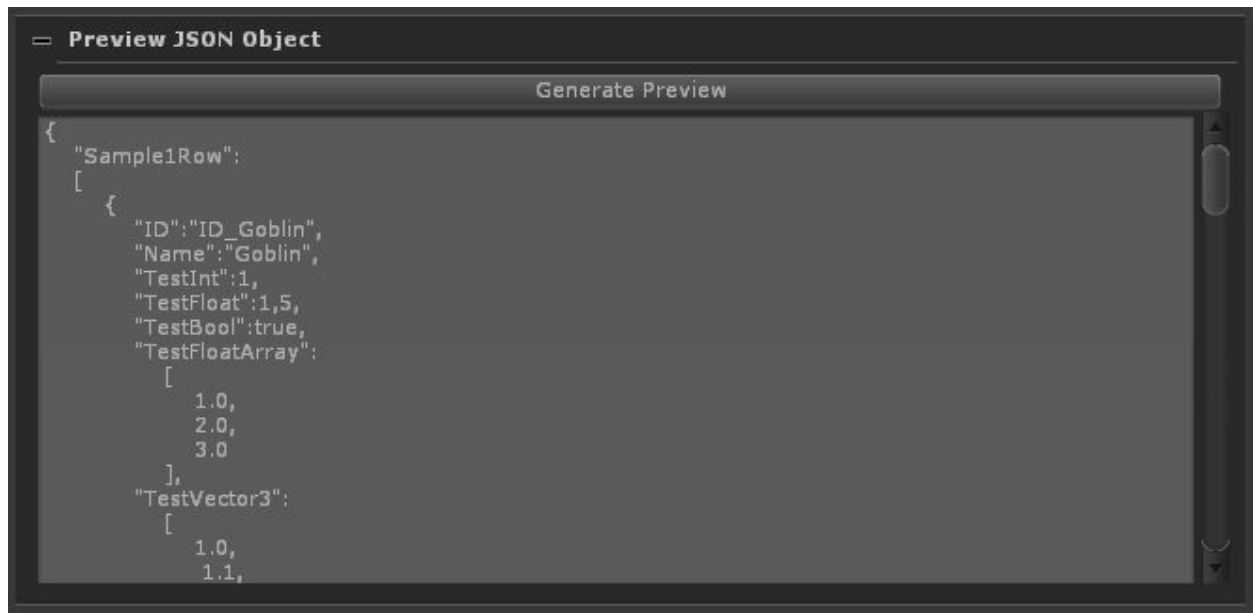
Stop On Blank Column: ☐

Stop On Blank Row: ☐

Ignore ID Column: ☐

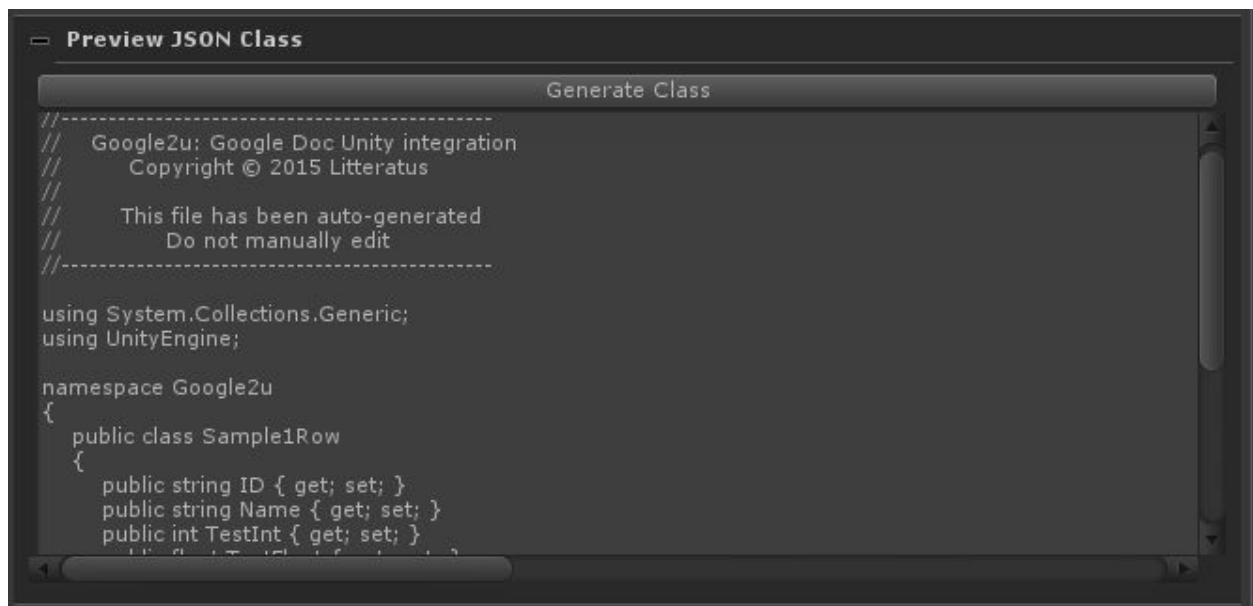
Escape Unicode Strings	Convert Unicode characters to Ascii using Escape sequences
Convert Cell Array to String	Instead of generating tags for entries in arrays, convert arrays to strings.
Export Generated Class	Export the sample class that is generated along with the normal JSON output
JSON Export Type	Export Object: The exported JSON will be encapsulated in a JSON object. Export Array: The exported JSON will be in an array format
Stop on Blank Column	When the export process reaches a blank column, stop processing any further columns
Stop on Blank Row	When the export process reaches a blank row, stop processing any further rows
Ignore ID Column	Omit the ID Column from the generated output and error checking. Note: The ID column is still necessary for processing the spreadsheet

Preview JSON Object:



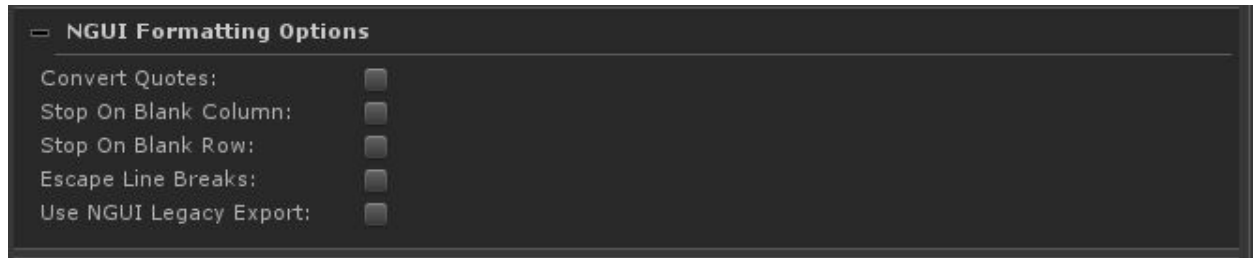
Generate a preview of the JSON output. The Unity window is not designed for very large amounts of text, thus if your JSON output is too long, it will cut off part of the file. Use this preview window to verify that you are getting the output you expect, but use the JSON Export feature to ensure your file is complete. The final JSON output will not contain formatting, and will be a single line. The preview contains all the formatting to make the JSON output more readable.

Preview JSON Class:



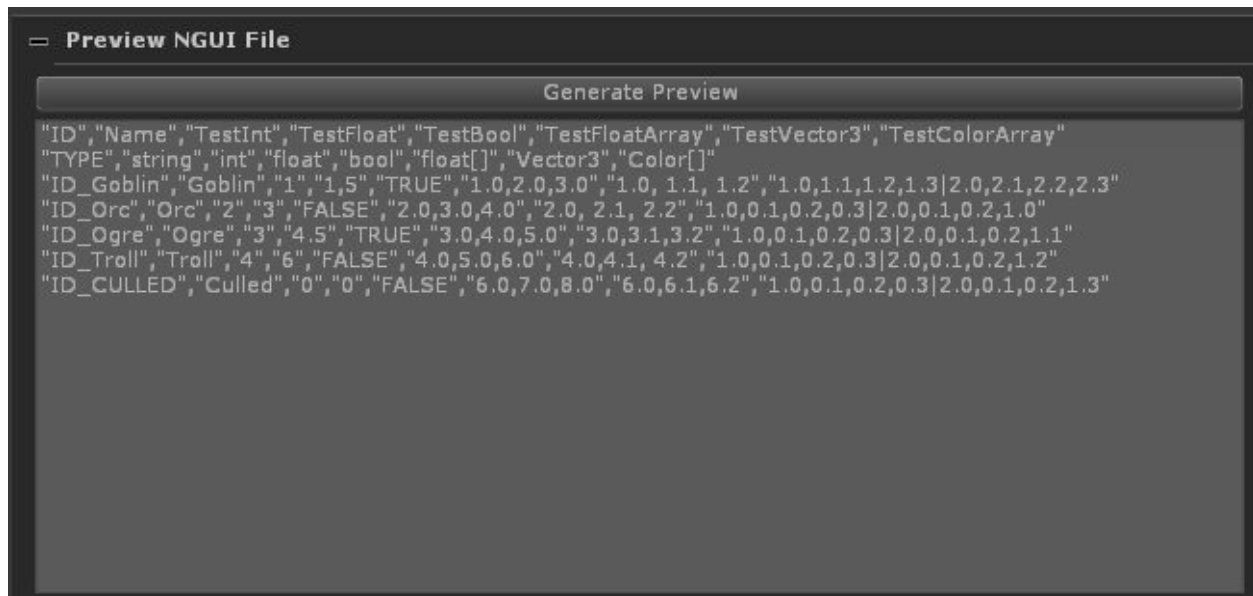
This class contains the code necessary for JSON object serialization of your exported JSON class. It is not necessary to use, but provided as a handy reference.

NGUI Formatting Options:



Convert Quotes	Enclose all data with Quotes
Stop on Blank Column	When the export process reaches a blank column, stop processing any further columns
Stop on Blank Row	When the export process reaches a blank row, stop processing any further rows
Escape Line Breaks	Replace Line Breaks with an escape sequence
Use NGUI Legacy Export	If you are using an NGUI version prior to 3.5.0 then you will need to use this option, as the Localization files were exported individually for each language

Preview NGUI File:



Generate a preview of the NGUI output. The Unity window is not designed for very large amounts of text, thus if your NGUI output is too long, it will cut off part of the file. Use this preview window to verify that you are getting the output you expect, but use the NGUI Export feature to ensure your file is complete.

Editing The Worksheets in Unity

	A	B	C	D	E	F	G	H	I
1	ID	Name	TestInt	TestFloat	TestBool	void	TestFloatArray	TestVector3	TestColorArray
Type▼	String ▼	String ▼	Int ▼	Float ▼	Bool ▼	Void ▼	FloatArray ▼	Vector3 ▼	ColorArray ▼
3	ID_Goblin	Goblin	1	1,5	TRUE		1.0,2.0,3.0	1.0, 1.1, 1.2	1.0,1.1,1.2,1.3 2.0,2.1,2.2,2.3
4	ID_Orc	Orc	2	3	FALSE		2.0,3.0,4.0	2.0, 2.1, 2.2	1.0,0.1,0.2,0.3 2.0,0.1,0.2,1.0
5	ID_Ogre	Ogre	3	4,5	TRUE		3.0,4.0,5.0	3.0,3.1,3.2	1.0,0.1,0.2,0.3 2.0,0.1,0.2,1.1
6	ID_Troll	Troll	4	6	FALSE		4.0,5.0,6.0	4.0,4.1, 4.2	1.0,0.1,0.2,0.3 2.0,0.1,0.2,1.2
7	void	Troll	4	6	TRUE		5.0,6.0,7.0	5.0,5.1,5.2	1.0,0.1,0.2,0.3 2.0,0.1,0.2,1.3
8									
9	ID_CULLED	Culled	0	0	FALSE		6.0,7.0,8.0	6.0,6.1,6.2	1.0,0.1,0.2,0.3 2.0,0.1,0.2,1.3

You are able to make simple modifications to your worksheet from directly within Unity. Please consider that formulas are not in the same format as in the Google Document, and while it is possible to edit them in Unity, you will have to take that into consideration.

Just like Google Docs, the Column Headers and Rows are displayed. The only exception is the second row, which can be a Type row. The top of the spreadsheet contains the Column Letter Designation. Clicking and Dragging this column letter will resize the column.

The first row of the spreadsheet needs to contain your column headers. These describe the data in the column, and is what you will use later to access the data in your exported database.

The second row of the spreadsheet can either be a normal Data row, or a Type row.

Type▼	String ▼	String ▼	Int ▼	Float ▼	Bool ▼	Void ▼	FloatArray ▼	Vector3 ▼	ColorArray ▼
-------	----------	----------	-------	---------	--------	--------	--------------	-----------	--------------

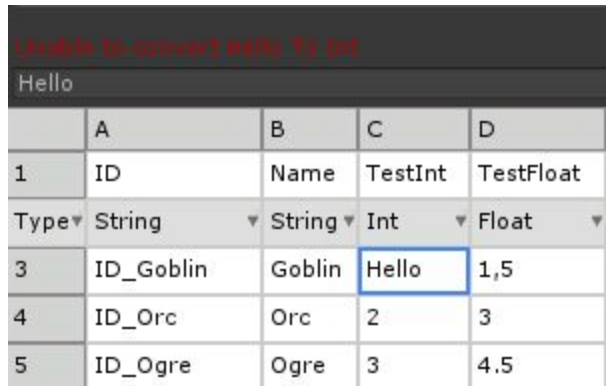
This can be changed by selecting the down arrow in the Row Header. Keep in mind that if Type row is selected, it will overwrite data in your Google Doc. Only select the Type Row designation if you are using the second row to designate data types for the columns.

Clicking a cell will populate the edit box above the grid view. There you can change the value contained in the cell. When finished updating, click the Update button to commit your change. If you select another cell without updating, the value will revert.

ID_Goblin									Update
	A	B	C	D	E	F	G	H	I
1	ID	Name	TestInt	TestFloat	TestBool	void	TestFloatArray	TestVector3	TestColorArray
Type▼	String ▼	String ▼	Int ▼	Float ▼	Bool ▼	Void ▼	FloatArray ▼	Vector3 ▼	ColorArray ▼
3	ID_Goblin	Goblin	1	1,5	TRUE		1.0,2.0,3.0	1.0, 1.1, 1.2	1.0,1.1,1.2,1.3 2.0,2.1,2.2,2.3
4	ID_Orc	Orc	2	3	FALSE		2.0,3.0,4.0	2.0, 2.1, 2.2	1.0,0.1,0.2,0.3 2.0,0.1,0.2,1.0

Validating The Worksheets in Unity

If your sheet contains errors, they will be highlighted in Red in the Grid View. Clicking a Validation Button will highlight the first error, and display a message indicating what the error is.

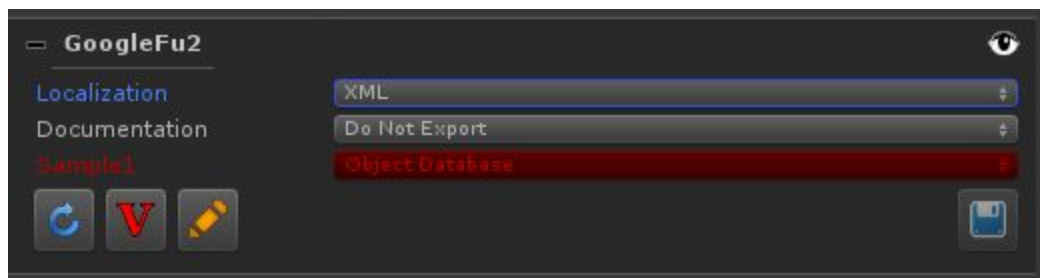


The screenshot shows a validation error message at the top: "Unable to convert Hello To int". Below the message is a table with 5 rows and 5 columns. The first row is a header with columns A, B, C, and D. The second row is a sub-header with columns ID, Name, TestInt, and TestFloat. The third row is a sub-header with columns Type, String, String, Int, and Float. The fourth row contains the values "ID_Goblin", "Goblin", "Hello", and "1,5". The fifth row contains the values "ID_Orc", "Orc", "2", and "3". The sixth row contains the values "ID_Ogre", "Ogre", "3", and "4.5". The cell containing "Hello" is highlighted with a blue border, indicating it is the source of the error.

	A	B	C	D
1	ID	Name	TestInt	TestFloat
Type	String	String	Int	Float
3	ID_Goblin	Goblin	Hello	1,5
4	ID_Orc	Orc	2	3
5	ID_Ogre	Ogre	3	4.5

In this case, the string “Hello” occupies a column designated as an Integer column. This would ultimately cause a build error, so G2U does not allow it.

While a sheet contains an error, it will be highlighted in Red in the list view, indicating that it is not able to be exported. If a sheet marked for export has errors, the Export All button in the list view will also be disabled.

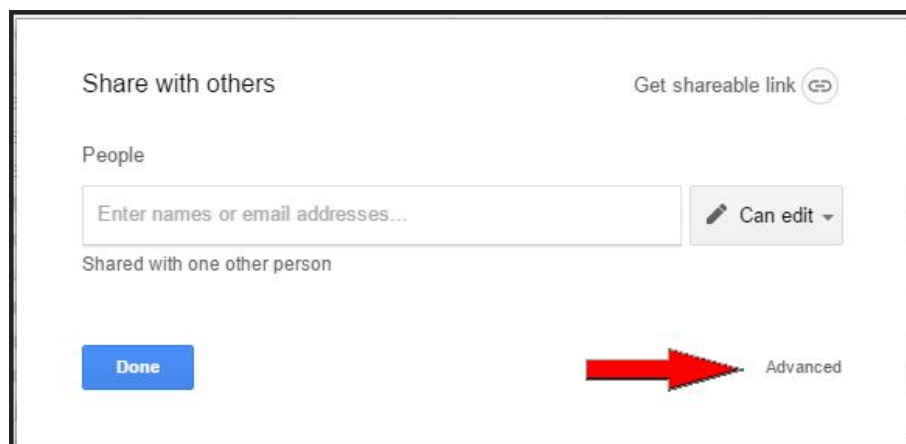


Using Manual Workbooks (Optional)

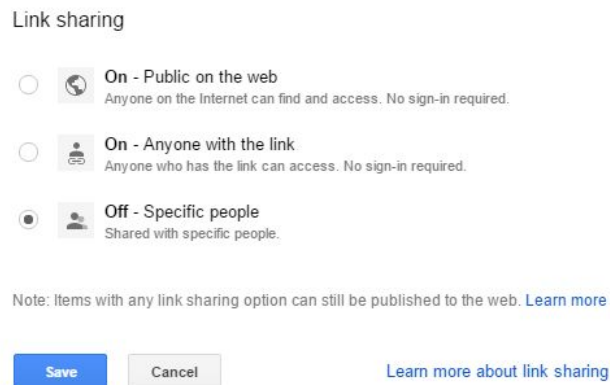
If you are planning to use a published workbook, you will need to tell G2U how to access this workbook. You will need to first have the URL of the published workbook, to let G2U know where it is located. If you have authenticated with your Google credentials, G2U will have access to your workbooks, and this step is purely optional.

The Workbook you are attempting to access must be public, and published to the web. If you are attempting to access a workbook that is private, you will need to authenticate with Google to view it.

To set the workbook to public, open it in Google, open the File menu and select “Share...”. In the Sharing Settings dialog you will need to click the “Advanced” link.

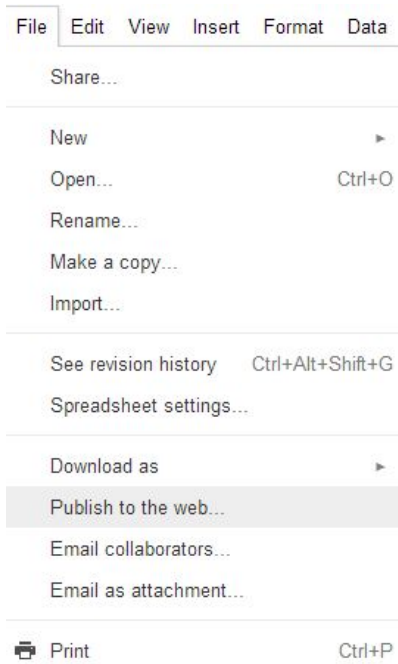


Another dialog will appear allowing you to select the sharing settings. Choose “Change...” option. If you wish to allow anyone to edit the workbook publically you may select the “Public on the Web” option, or “Specific People” if you simply wish to allow G2U to access the data.

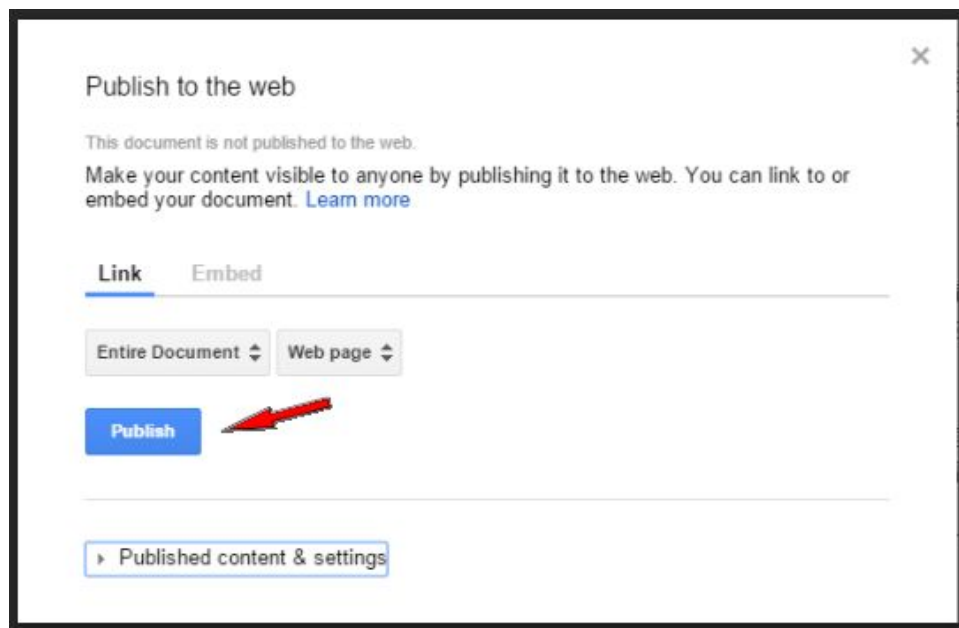


Save, and exit the dialog. Next you will need to publish the workbook.

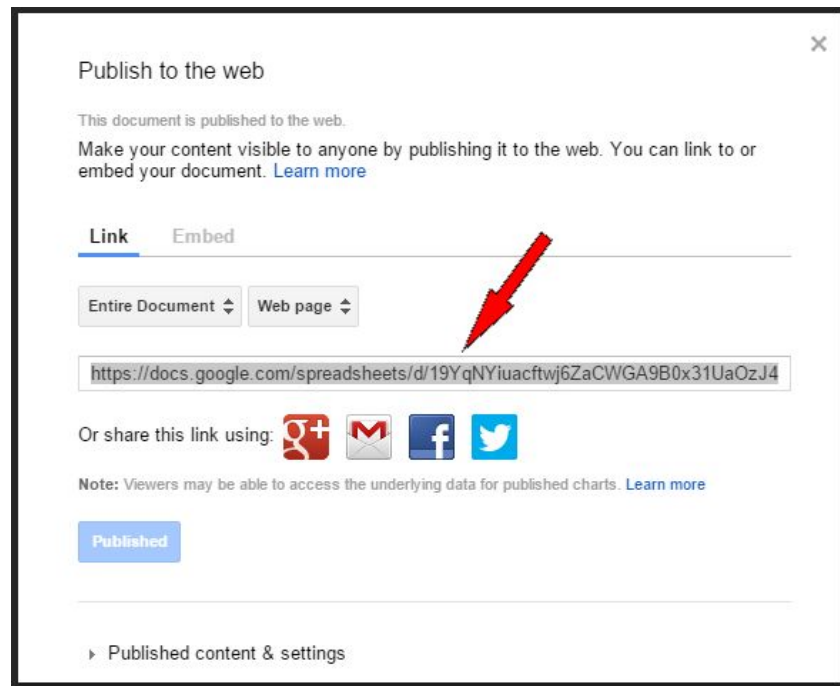
To find the published workbook URL, open the spreadsheet in Google and open the File menu, and select “Publish to the web...”



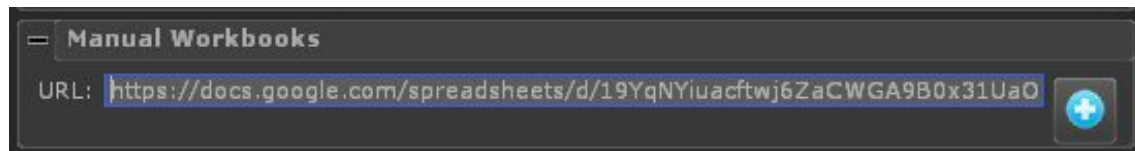
You will then see a dialog that gives you options for publishing the worksheet. Leaving all of the options as defaults and choosing the “Publish” option, Google will generate a URL for the workbook.



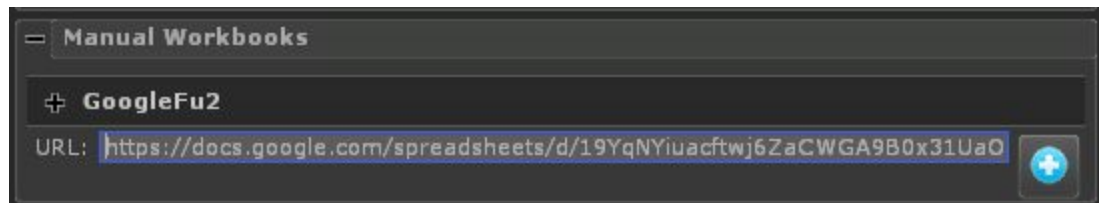
The URL in the Document Link box is the URL that you will be giving G2U. Copy it from the box.



Once you have published the workbook, return to the G2U window and navigate to the Manual Workbooks section. There you will paste the URL obtained from Google into the box, and click the “Add” button.



The workbook will appear under the “Manual Workbooks” heading.



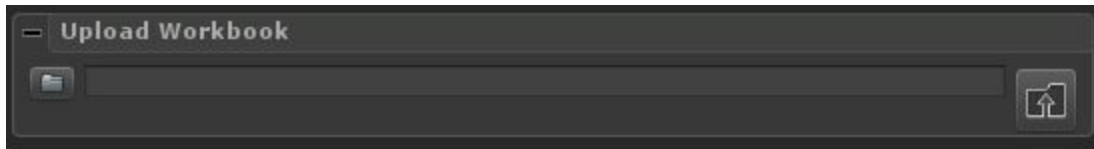
Here you will have all of the options available as if it were an Account Workbook. If you are signed into Google and have access to the published workbook, you will be able to edit the values, otherwise these workbooks are considered Read-Only.



The icon in the List View will remove the Manual Workbook from your list.

Uploading Workbooks (Optional)

If you have a spreadsheet on your drive, G2U can assist in uploading that to Google Spreadsheets by using the “Upload Workbook” section.



Google is able to process the following file types: .xls, .xlsx, .ods, .csv, .txt, and .tsv
Use the folder icon to browse to a Spreadsheet and click the “Upload” button on the right.

Some uploaded spreadsheets may use “Untitled” as the name until you change it manually in Google. This happens rarely, and is an issue with the Google API. In most cases when you upload the file it will appear in your documents list as the name of the file.

Accessing your data

After exporting the worksheets, you will want to access your data. Depending on how you exported, this is done in different ways. The XML and JSON files are designed to be used with other tools, or read into your game manually. The NGUI Localization files are used by NGUI directly, and the NGUI Localization documents will illustrate how to use it. The GameObject Database and Static Database have generated functions that allow you to access the data. Below are examples on how to use them.

GameObject Database Data Retrieval

A GameObject Database is an object that exists in the scene, and has all of the database information stored within it. It contains a script that is generated by GoogleFu that provides access to the data contained in it. You can access the data by first gaining access to the script.

The code below is a snippet from the GoogleFu Sample

```
// Example of an Object based Database.
CharacterStats _statsDb;

// Since our database exists within an object in the scene, we'll need to find it.
// Alternatively you could expose the database as a public member, and use the
// inspector to set it.
GameObject statsdbobj = GameObject.Find("databaseObj");
if ( statsdbobj != null )
{
    // Get the CharacterStats component out of the GameObject.
    // CharacterStats is the name of the worksheet in the google spreadsheet
    // that we are getting the monster information from
    _statsDb = statsdbobj.GetComponent<CharacterStats>();
}
```

Once you have access to the script, accessing the data is simple

To access entire rows, use the code below

```
// The rowIds enumeration is generated from the first column of the spreadsheet
// It serves as a lookup table for accessing specific data rows
_statsDb.Rows[ ( int )CharacterStats.rowIds.AI_TROLL ];

// You can also access data randomly
_statsDb.Rows[ Random.Range( 0, _statsDb.Rows.Count ) ];
```

```
// You may also use the GetRow function
// GetRow accepts either the rowIds enum
_statsDb.GetRow( CharacterStats.rowIds.AI_GOBLIN );

// GetRow also accepts a string
_statsDb.GetRow( "AI_GOBLIN" );
```

Object Databases can be used in conjunction with Playmaker, and also have custom inspector that is generated to view the data.



Static Database Data Retrieval

A static database is not tied to a GameObject, but rather a static class that can be accessed anywhere. The data is stored in a variable inside of the class, and is populated when the class is instantiated. Static databases should be instantiated at load time, not during gameplay, as it does take time to populate the database.

The code below is a snippet from the GoogleFu Sample

```
// Example of a Static Database.
Items _itemsDb;

// The Items database is a Static class. Use it by grabbing the .Instance, this will
// ensure the database is correctly initialized. Larger databases may take a while
// to initialize, so grabbing an instance before the game is updating is recommended.
_itemsDb = Items.Instance;
```

To access entire rows, use the code below

```
// The rowIds enumeration is generated from the first column of the spreadsheet
// It serves as a lookup table for accessing specific data rows
itemsDb.Rows[ (int)Items.rowIds.ITEM_RING ];

// You can also access data randomly
_itemsDb.Rows[ Random.Range(0, _itemsDb.Rows.Count) ];

// You may also use the GetRow function
// GetRow accepts either the rowIds enum
_itemsDb.GetRow( Items.rowIds.ITEM_PANTS );

// GetRow also accepts a string
_itemsDb.GetRow( "ITEM_PANTS" );
```


Retrieving Column Data from a Row

The generated scripts contain variables that are named from the column headers in the spreadsheet. The names have been altered to ensure proper variable names, but are very similar. You will want to open the generated script to verify the column names. To access the data, you have a choice of accessing it directly, or by using the GetStringData function, which will return a string value (Regardless if the variables are strongly typed or not).

```
// Accessing a specific value from a row can be done multiple ways
ItemsRow myRing = itemsDb.Rows[ (int)Items.rowIds.ITEM_RING ];

string RingName1 = myRing ._NAME;

string RingName2 = myRing.GetStringData("name");
```

Samples

Included with GoogleFu is an example spreadsheet that you can upload to your Google account. This example has 3 worksheets to illustrate the syntax you should use when creating your own GoogleFu compatible spreadsheets. The example is located at:

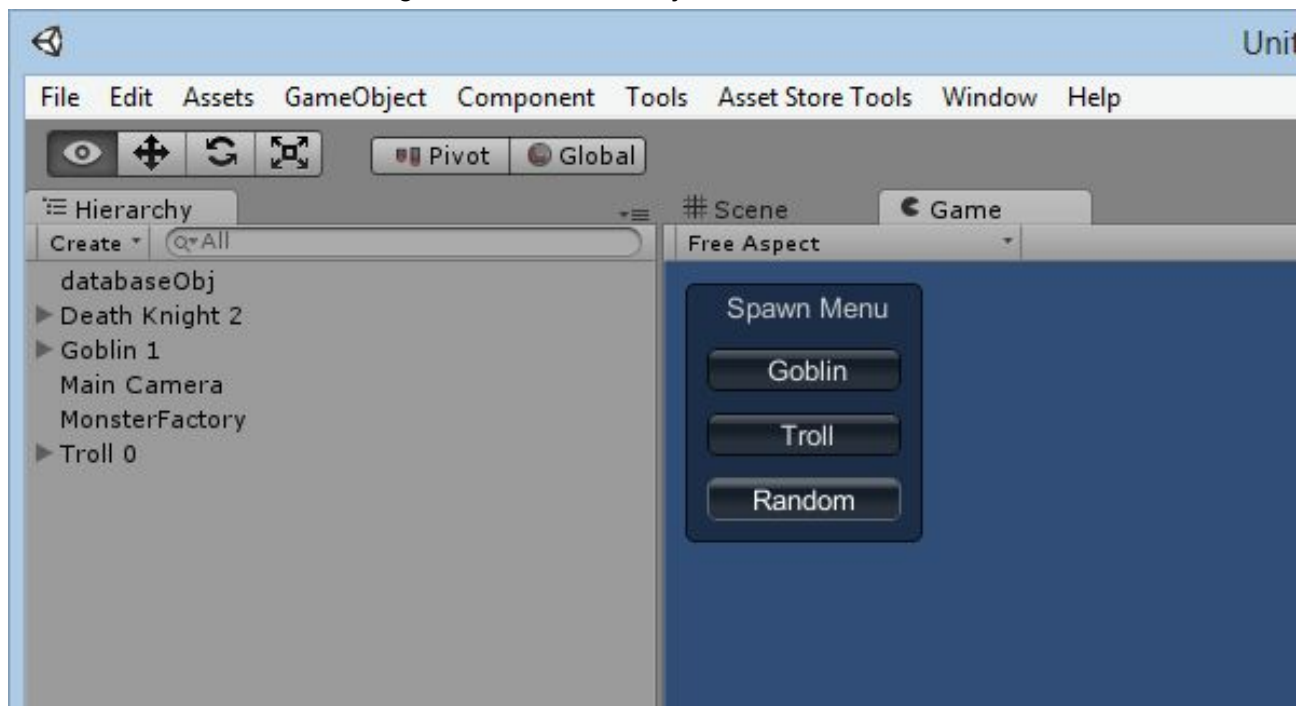
../Assets/GoogleFu/Samples/Data/SampleDatabase.xlsx

You do not need to be able to open this file locally in order to use it. Simply use the “Upload Workbook” page in the “Workbooks” tab to upload it to Google, and let Google take care of opening it.

Also you will find a sample that uses both Object Database and Static Database to generate dynamic game objects. This example is located at:

../Assets/GoogleFu/Samples/Scenes/DemoScene

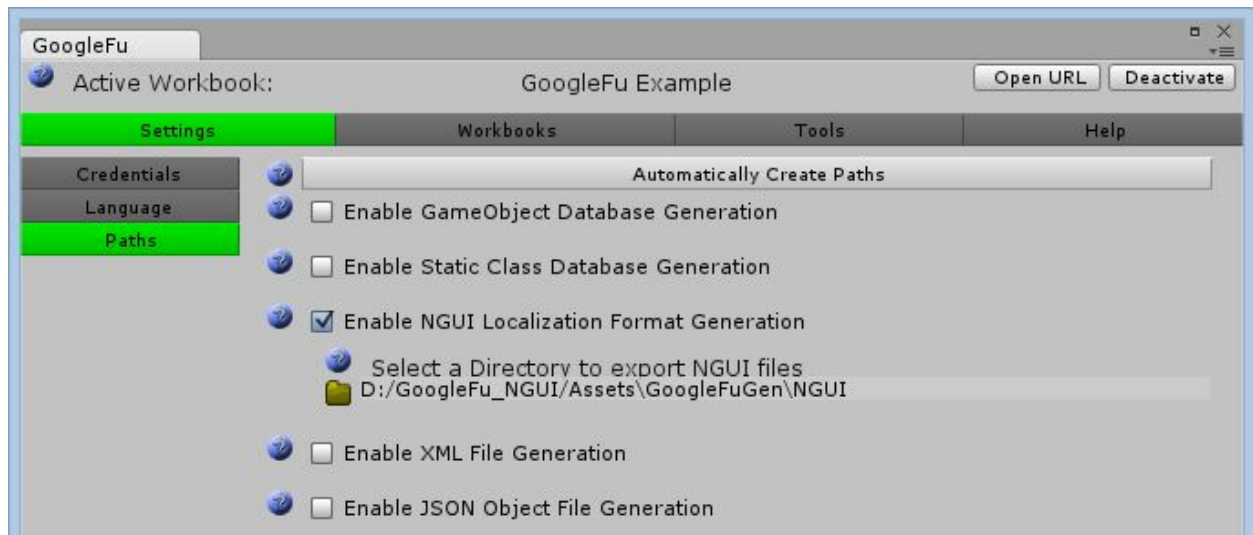
This is a bare bones example of how to use the databases once exported. Simply load the scene and click the buttons to generate a Game Object with data from the database.



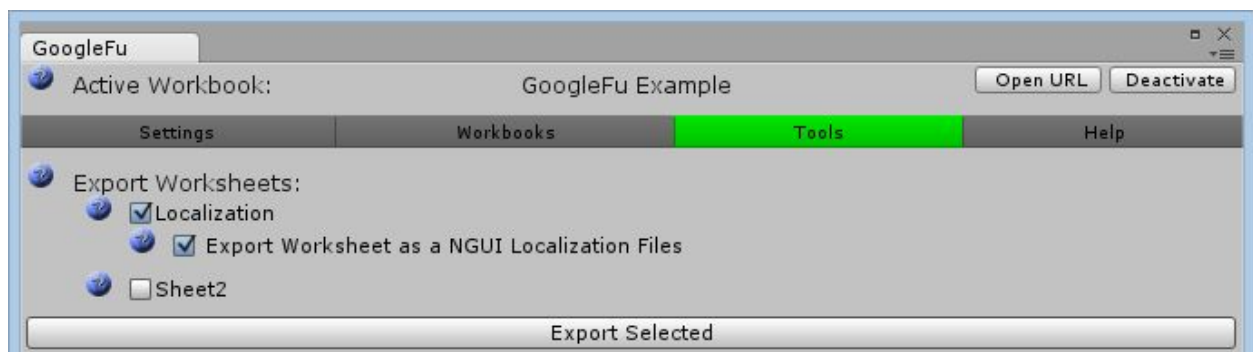
Using the exported data as NGUI Localization files

If NGUI is present in the project, GoogleFu will recognize it* and allow you to export the database as NGUI Localization files.

*GoogleFu looks for a specific file that NGUI provides.

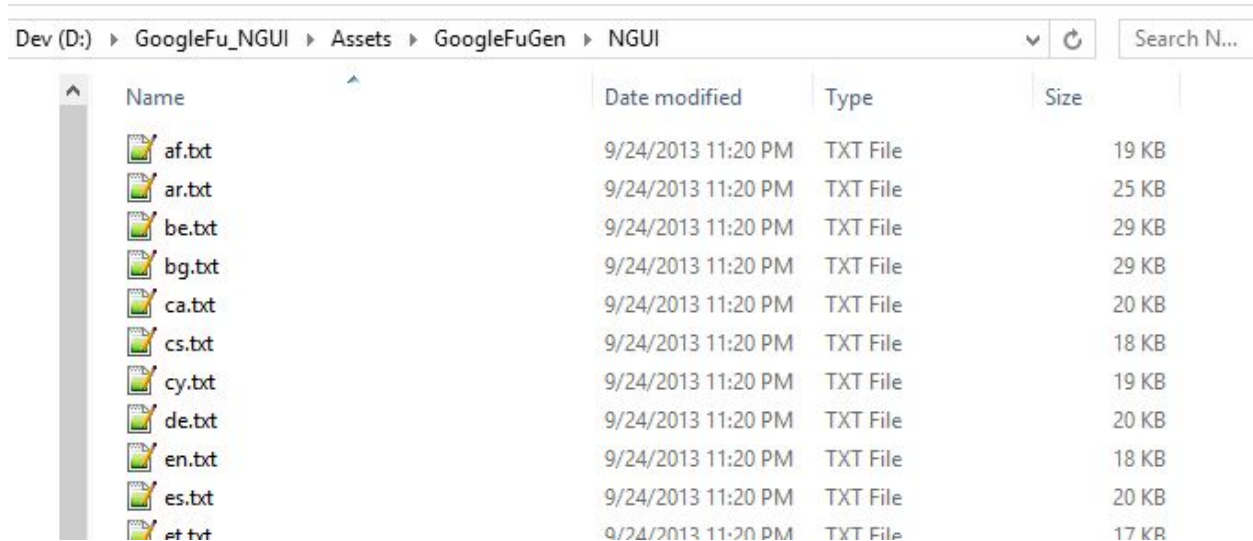


Once you have selected an Active Workbook, you will be given the option to export as NGUI Localization Files.



The following illustrates what will be generated if the “Legacy” option is checked. This is only for old versions of NGUI. The newer version requires a CSV file, and is generated instead of the individual .txt files if the “Legacy” option is unchecked.

When you export these files, they will be created in the path you have specified.



The screenshot shows a Windows Explorer window with the address bar set to 'Dev (D:) > GoogleFu_NGUI > Assets > GoogleFuGen > NGUI'. The search bar contains 'Search N...'. The file list displays the following data:

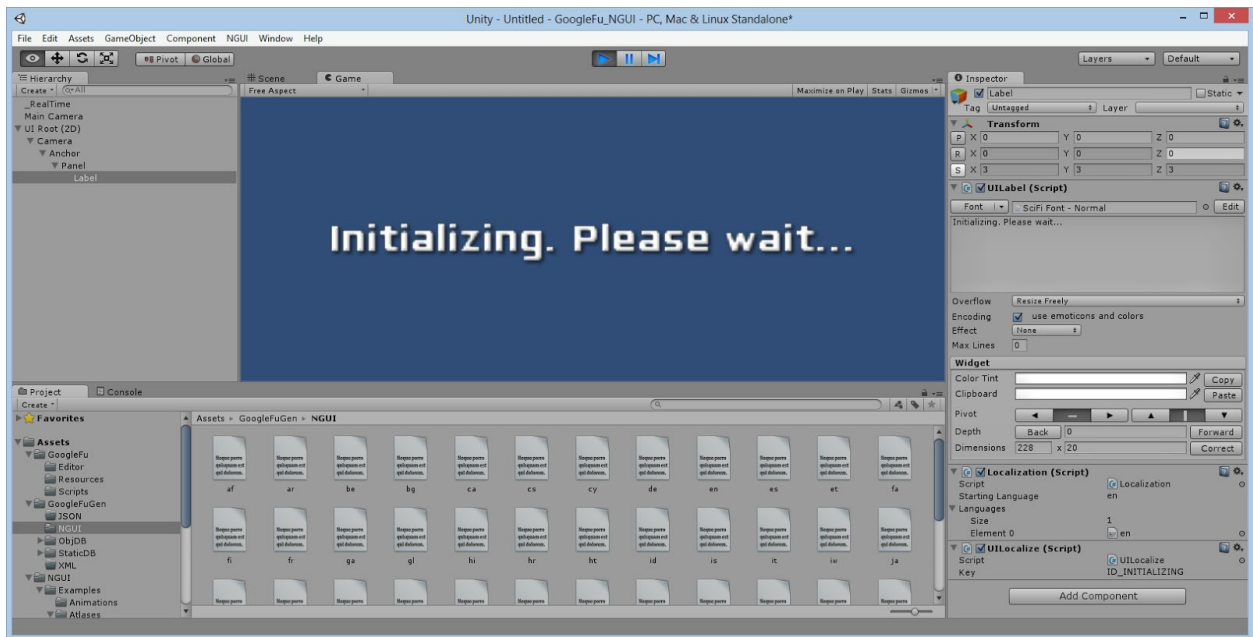
Name	Date modified	Type	Size
af.txt	9/24/2013 11:20 PM	TXT File	19 KB
ar.txt	9/24/2013 11:20 PM	TXT File	25 KB
be.txt	9/24/2013 11:20 PM	TXT File	29 KB
bg.txt	9/24/2013 11:20 PM	TXT File	29 KB
ca.txt	9/24/2013 11:20 PM	TXT File	20 KB
cs.txt	9/24/2013 11:20 PM	TXT File	18 KB
cy.txt	9/24/2013 11:20 PM	TXT File	19 KB
de.txt	9/24/2013 11:20 PM	TXT File	20 KB
en.txt	9/24/2013 11:20 PM	TXT File	18 KB
es.txt	9/24/2013 11:20 PM	TXT File	20 KB
et.txt	9/24/2013 11:20 PM	TXT File	17 KB

Each file is written in a format that NGUI expects for proper localization

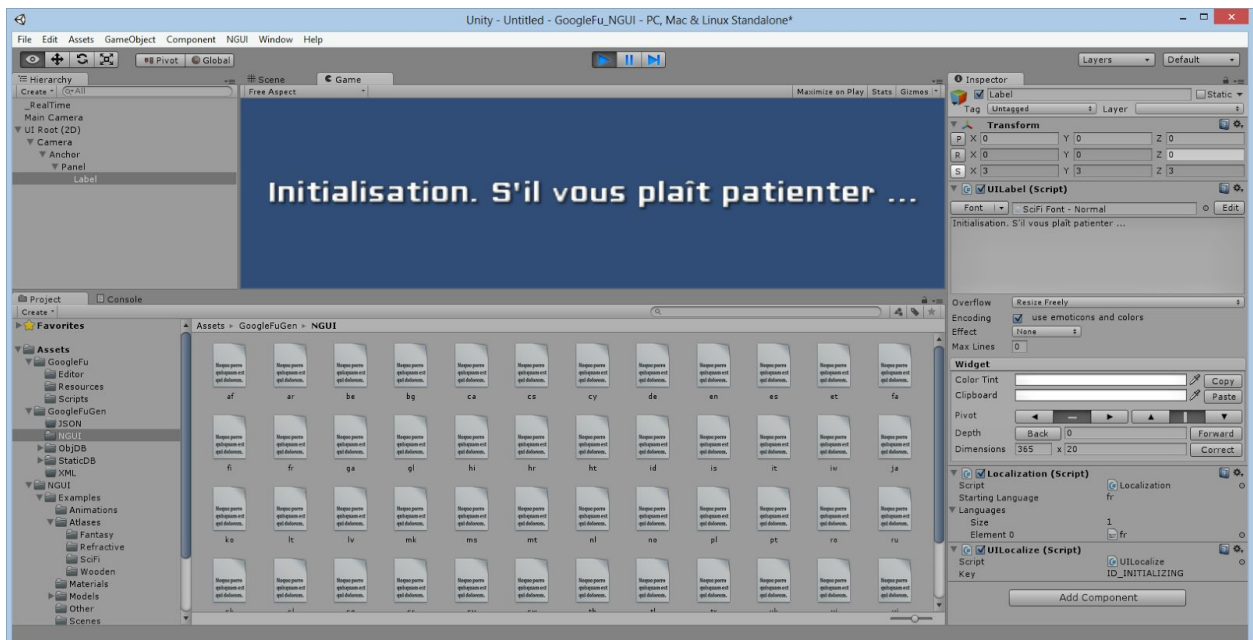
```
1  Flag = Flag-en
2  ID_INITIALIZING = Initializing. Please wait...
3  ID_AUTHENTICATING = Authenticating with Google. Please wait...
4  ID_IMPORTING_MESSAGE = Importing Database. Please wait...
5  ID_SETTINGS = Settings
6  ID_WORKBOOKS = Workbooks
7  ID_CREDENTIALS = Credentials
8  ID_LANGUAGE = Language
9  ID_PATHS = Paths
10 ID_TOOLS = Tools
11 ID_HELP = Help
12 ID_HELP_MAIN = Main
13 ID_HELP_LOCAL = Localization
14 ID_HELP_DB = Database
15 ID_GENERATE_PATHS = Automatically Create Paths
```

Use the NGUI UILocalize script to designate a row ID for the UILabel, and the NGUI Localization script to load the languages.

English:



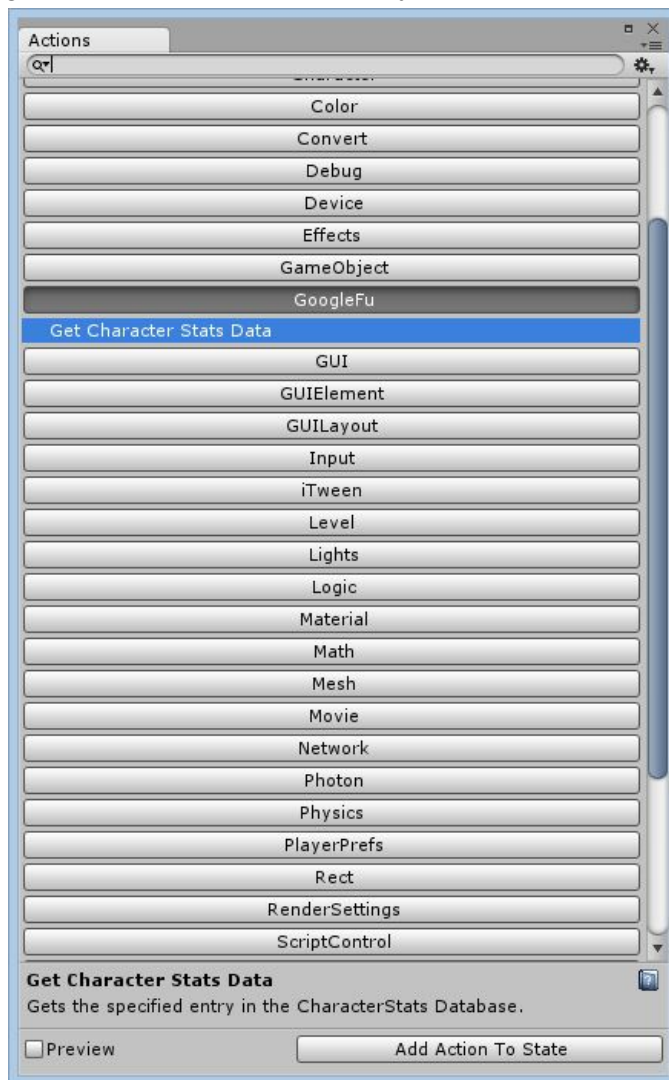
French:



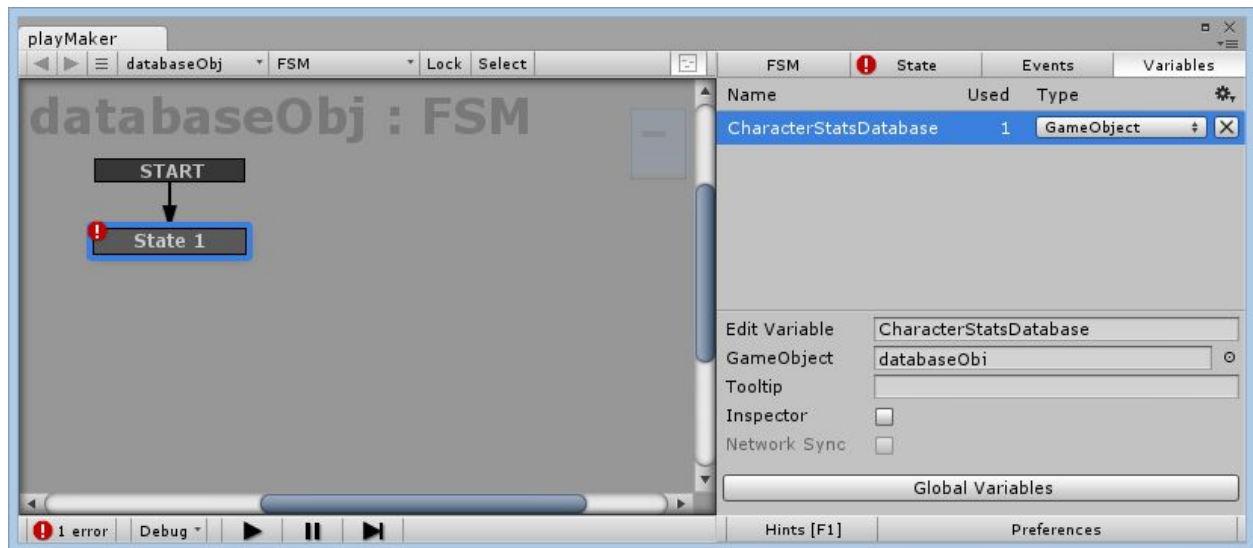
Using G2U with Playmaker

If Playmaker is present in the project, G2U will recognize it and allow you to generate Playmaker actions as part of the Object Database Export

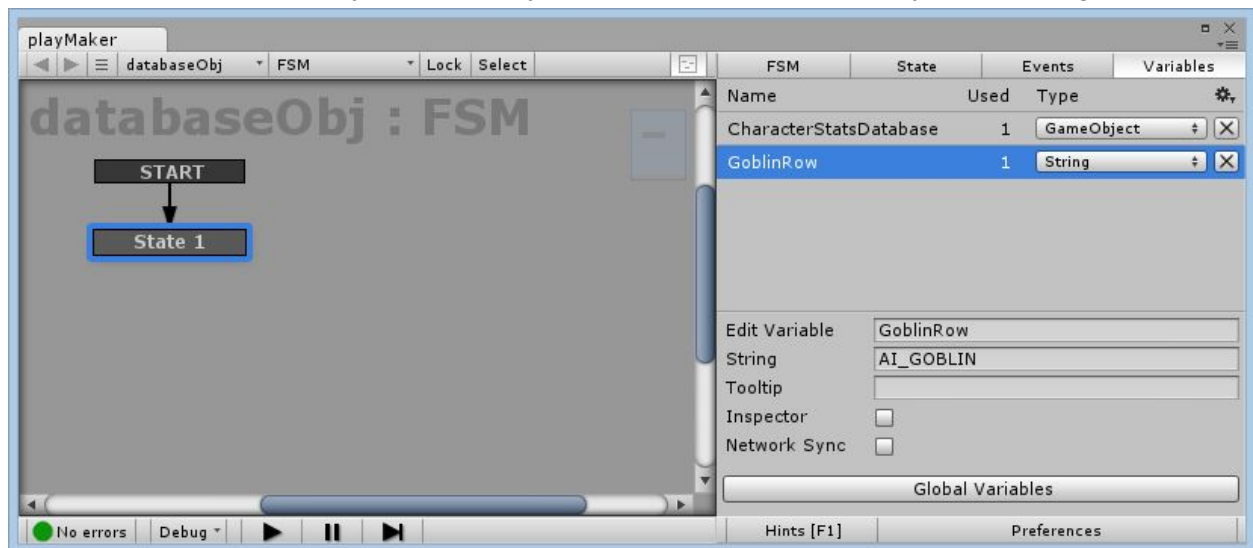
When you have generated the database, create a Playmaker FSM on a Game Object and open the Action Browser. A new option will be available for G2U, and the Action that has been generated will be named after your worksheet



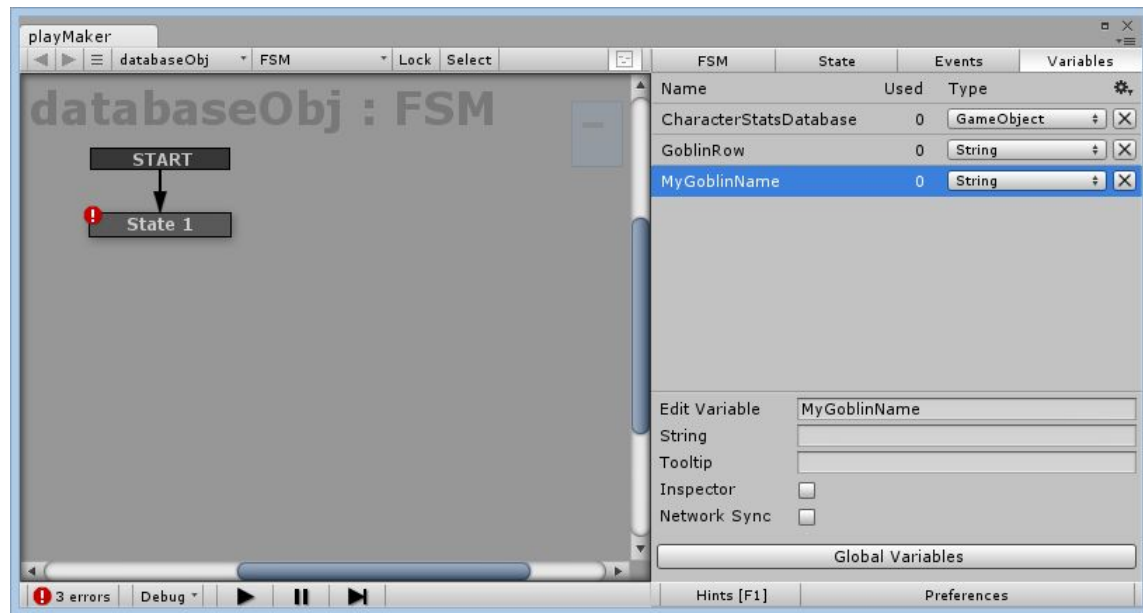
You will need to supply the object that you attached the database to via a Playmaker GameObject variable



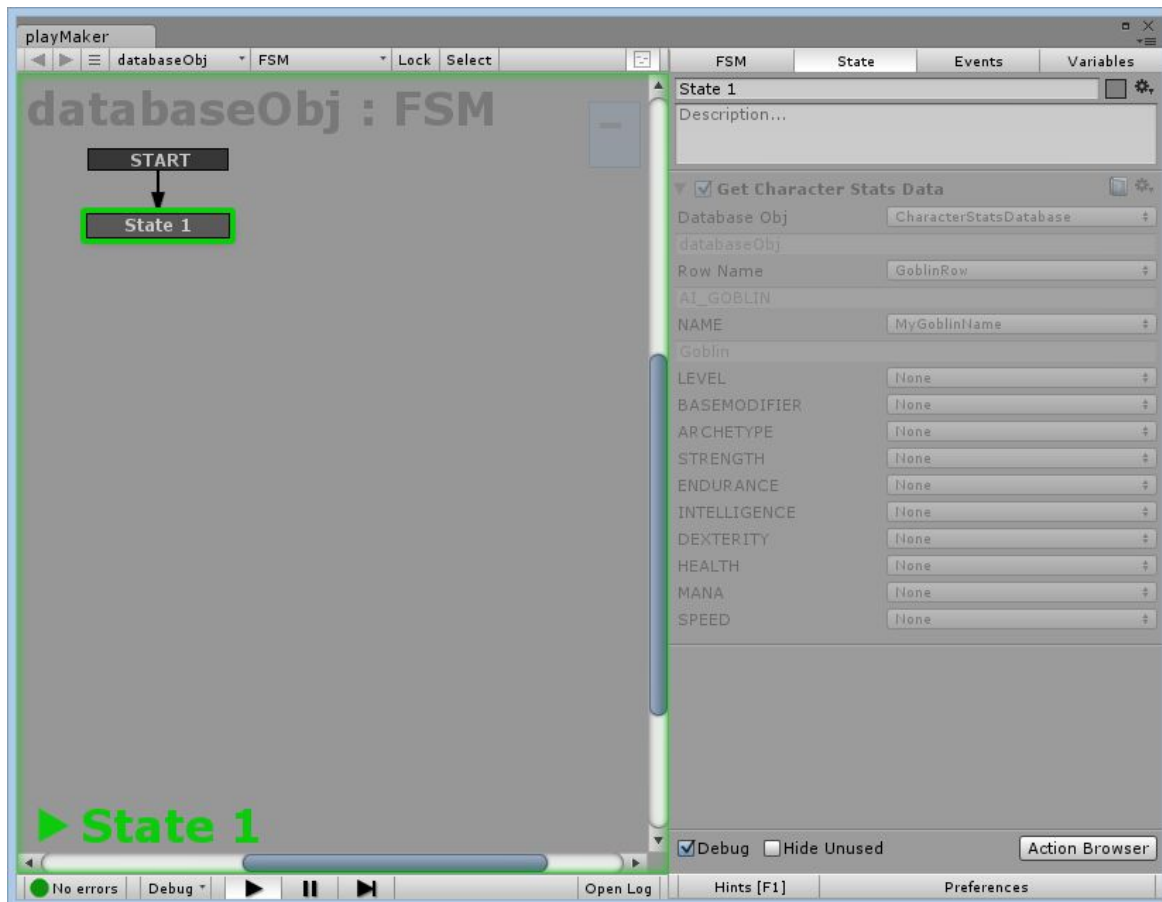
You will also need to supply the Row ID you want to access with a Playmaker String Variable



Any data you wish to retrieve from the database will also need to be stored in variables



Once you supply the State with the relevant information, all you need to do is Play, once the action is executed, all of the data will be pulled from the database and placed in the variables.



Source Code, Third Party, and Copyright information

Always, always keep a backup of your data. G2U does overwrite existing files when it generates the database.

The DLL's that are included with G2U are redistributed as part of the Google GData API. With the exception of removing the dependency to Newtonsoft.JSON.Net.dll, They have not been modified in any way, and full source code for the API is available through Google.

NGUI is copyrighted by [Tasharen Entertainment](#)

Playmaker is copyrighted by [Hutong Games](#)

Google is copyrighted by... wait for it... [Google](#)

Tell your friends about G2U. We are a couple of guys just trying to make a living, not some huge faceless corporation. Also, please rate us on the Unity Asset Store if you like G2U. If not, let us know, and we'll try to fix the problem!

<http://www.litteratus.net>

FAQ

Can I use G2U to access my Google Spreadsheet at runtime?

No. G2U is strictly a Pre-Build process. It simply streamlines pulling the information from a Google Spreadsheet and saving it in a way that is easily accessible from the game. Communication with Google can sometimes be very slow, and is not suited for real-time network traffic. Aside from that, it would also require you to place your Google Credentials directly into the code, which is a huge security risk. SQL databases that are accessed via PHP scripts on a webserver is the fastest, most secure way to handle runtime database access.

What is the difference between the Full version and the Free version?

The Free version of G2U is limited to exporting 10 Row x 10 Col spreadsheet. It is designed to allow you to use the plugin in it's full capacity, but restrict the amount of data it will process. If you only need a 10x10 spreadsheet, then go nuts, there are no other restrictions. The Free version of G2U is also compiled into a DLL, where the Full version contains all of the raw code files.

Do I have to use my Google account?

No, you are not required to sign in with a Google account. You can make any of your personal spreadsheets publicly accessible, and G2U will be able to import the data. However if you do allow G2U to authenticate, it

will be able to access private spreadsheets tied to your account.

What does G2U do exactly?

G2U retrieves data from Google Spreadsheets, and formats it in a way that is easy to retrieve in your game. This makes a Data Driven design extremely efficient, and also allows you to very easily localize all of your strings, and retrieve them by using a string ID.

Does this mean my game must contact Google?

No, G2U does all of the data retrieval outside of your game, and packs it up into files, or stuffs all of the data into a GameObject in your scene. This is all done prior to running the game, so there is no communication with Google at runtime.

Can I use G2U entirely offline using Excel or other Spreadsheets?

No, G2U utilizes the Google Spreadsheets API and requires access to the Google Docs cloud in order to retrieve data. However you absolutely can create your spreadsheets offline using Excel, OpenOffice, or other programs, and G2U will assist in uploading those files to Google Docs when it's time to import them into your game.

Can I update my spreadsheet from inside of Unity?

Yes! Using the Full View, you can see your spreadsheet and update the values directly within the Unity Editor.

I am getting a crash! What do I do?

Please contact us! We want to know exactly what you did when the crash happened. We are more than willing to help get you back up and running immediately, and will update G2U with a fix as soon as possible!

contact@litteratus.net

Version History

2.1.0

IMPORTANT NOTICE

The default install folder has been changed. This means you MUST remove the previous /Assets/Google2u folder (You may leave your Google2uGen folder in-tact) before you install the newest version. Otherwise you will get a screen full of warnings/errors.

- G2U now lives in the Standard Assets folder. This means it will not need to recompile along with the rest of your code. This is a huge time-saver!
- All Using directives have been moved inside of the Google2u namespace to help prevent accidental library collisions.
- Newtonsoft.Json.net has been removed, and the GDATA libraries that required it have been modified. G2U is now fully compatible with the other JSON.Net libraries
- JSON Validation no longer requires the ID column to be valid... but it still MUST exist in your spreadsheet! There is now an option to suppress it from the export entirely
- Path issues and other misc bugs have been resolved

2.0.10

- NGUI Legacy Support returns to G2U (For those of you sticking with NGUI <3.5.0)
- Object Database Export now supports setting a GameObject per-sheet

2.0.9

- Fixed a null object when starting a new project

2.0.8

- G2U will now save preferences properly in all cases

- Byte arrays are once again properly supported
- Vector2 will once again validate properly
- CSV export will automatically enclose strings with commas, quotes, and newlines with quotes
- Rows will not attempt to validate past blank rows if "Stop on Blank Row" is checked
- Sheets will automatically be re-validated when changing options

2.0.7

- OAuth2 token retention improved
- Added the ability to hide Account Workbooks from the view
- Preferences are now saved to a file that can be transferred to other computers

2.0.6

- Fixed Manual Workbook access bug

2.0.5

- Replaced the Newtonsoft.Json.net.dll with one built against .NET 3.5 to comply with the Unity Editor's mono compatibility.
- Removed the legacy GoogleFu 1.0.15 project ([Rest in Peace](#))

2.0.4

- Fixed Authentication Issues with Google
- G2U no longer stores any kind of personal information, as it is all handled via OAuth2
- Fixed the ID column missing from JSON exports
- Fixed issues with JSON and Unity classes / complex data types, and arrays
- G2U is once again dependent on Newtonsoft.Json.Net

2.0.3

- Fixed issue with Playmaker export path.

- Fixed void rows not validating properly in some cases.

2.0.2

- Fixed Playmaker export issues
- Added Legacy support for old GoogleFu projects that had column names forced to lower case
- Fixed a CSV export issue with line breaks
- Added missing export options for NGUI export 2.0.1
- Recompiled and Submitted with Unity 5
- G2U is BACKWARDS COMPATIBLE with Unity 4.5.5+

2.0.0

- Rebranded to G2U
- Major UI Overhaul
- Multithreading Support
- View / Edit / Update spreadsheets from within Unity

1.0.15

- Updated to be Unity 5 compatible

1.0.14

- Fixed an issue where workbooks were not refreshing when using multiple Google accounts

1.0.13

- Fixed JSON output bugs
- Fixed Playmaker code generation when using void/ignore rows and columns
- Improved authentication and workbook regeneration speed. (Use Auto-Login.. it's faster!)

1.0.12

- Added feature to ignore marked columns and rows
- Fixed missing strings in the editor
- Fixed a bug with submitting non-excel documents when uploading to Google
- Improved auto-generation of output directories
- Fixed a bug with CSV exporting
- Added an option to surround strings with quotes in CSV export. Off by default.

1.0.11a

- Fixed issues with zero length arrays
- Fixed the lockout issue with Android and iOS build targets

1.0.10

Generated Code:

- Added a property for the number of columns in a Row (Row.Length)
- Added the ability to get the string value of a Row using the bracket operators

Arrays:

- Added support for data arrays in individual cells

GUI:

- Added the ability to choose delimiters for single cell arrays
- Added an option to trim whitespace on strings
- Added an option to trim whitespace on string arrays

Export:

- Added the option for CSV export directly

Compatibility:

- Restricted GoogleFu from operating under invalid build targets
- Updated Manual Workbook support to work with the new URL generated when you publish to the web
- Removed support for char, short and double value types

Bug Fixes:

- When generating an object database, if an error is detected the generation will stop immediately and report an error
- Quaternions are now displaying correctly in the Editor
- Improved some error conditions and debug output

1.0.9

- Compatible with NGUI 3.5.0 and greater

NGUI <3.5.0 still supported through the use of a Legacy checkbox

1.0.8

- Added more Playmaker actions to assist in retrieving data
- Added a ToString function to Rows to print out all data within the row

1.0.7

- Fixed an authentication issue when shared workbooks are marked as private/read only

1.0.6d

- Fixed an issue with Color not being properly generated in the custom editor script

1.0.6c

- Removed all using directives to maximize compatibility

1.0.6a

- Fixed a bug in the Database Object creation
- Fixed index out of range exceptions with static databases

1.0.6

- Fixed an issue with bools in the spreadsheet. They are now properly supported
- Changed a backslash to forward slash when exporting NGUI Localization, Mac compatibility
- Added an option to JSON exporting to choose between automatically escaping Unicode characters or not

1.0.5

- Moved DLL's into the Editor directory to prevent them from being included in final builds*
- Added support for Diakon Forge localization file export

*If you have previously installed GoogleFu, updating this package will NOT remove the old DLL's from the project. You must manually remove all DLL's contained in the Assets/GoogleFu directory, or remove GoogleFu from the project entirely, and re-import. This will not remove any of your saved settings.

1.0.4

- Fixed issues regarding Vector2, Vector3, and Quaternion types in the Custom Inspector code generation

1.0.3

- Rearranged file structure to remove icons from being included into final builds
- Removed a Using directive from a file that was causing build issues

1.0.2

- Recompiled the Newtonsoft.Json.dll. It should play nice with other plugins now

1.0.1

- Initial version.