Version from git

Documentation

Introduction

This asset gives the ability to set the Game Version (*ProjectSettings/Player/Version*) to match your project Git Tag🏷️. It could be set automatically when Unity builds the project or manually by using the Tools menu. There are various settings available to customize the version format and the asset usage.

⚠ Important notice ⚠

This asset obviously works only if you have git installed and initialized for your Unity project. The version will be created by fetcher the last git tag found on your repo. This tag should follow the format vX.Y.Z or X.Y.Z (X : Major number, Y : Minor number, Z : Revision number).

More information about git tag here:

🔗https://git-scm.com/book/en/v2/Git-Basics-Tagging

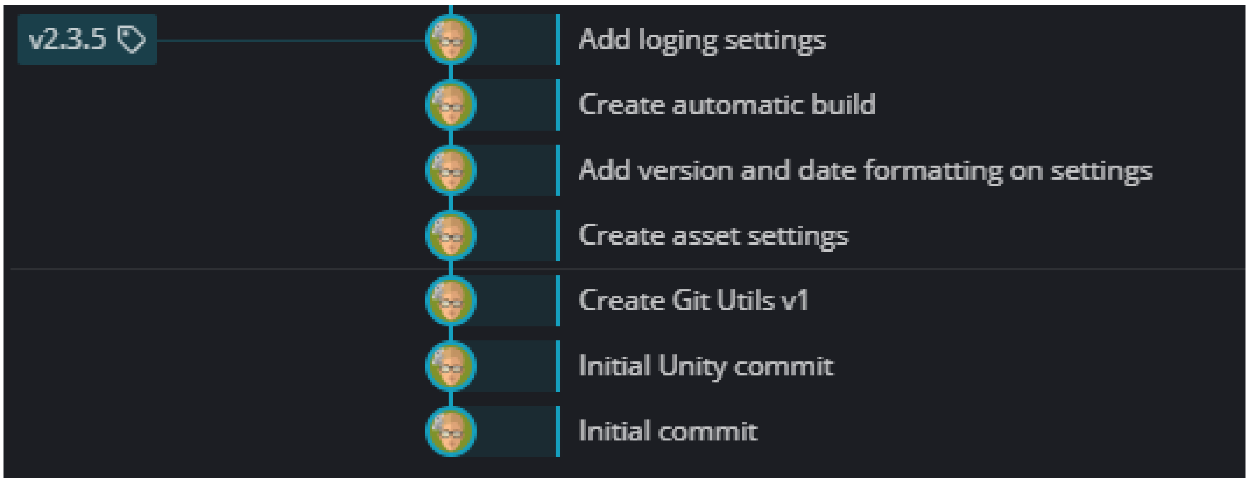
More information about software versioning format could be found here:

🔗https://semver.org/

🔗https://en.wikipedia.org/wiki/Software_versioning

Quick Start

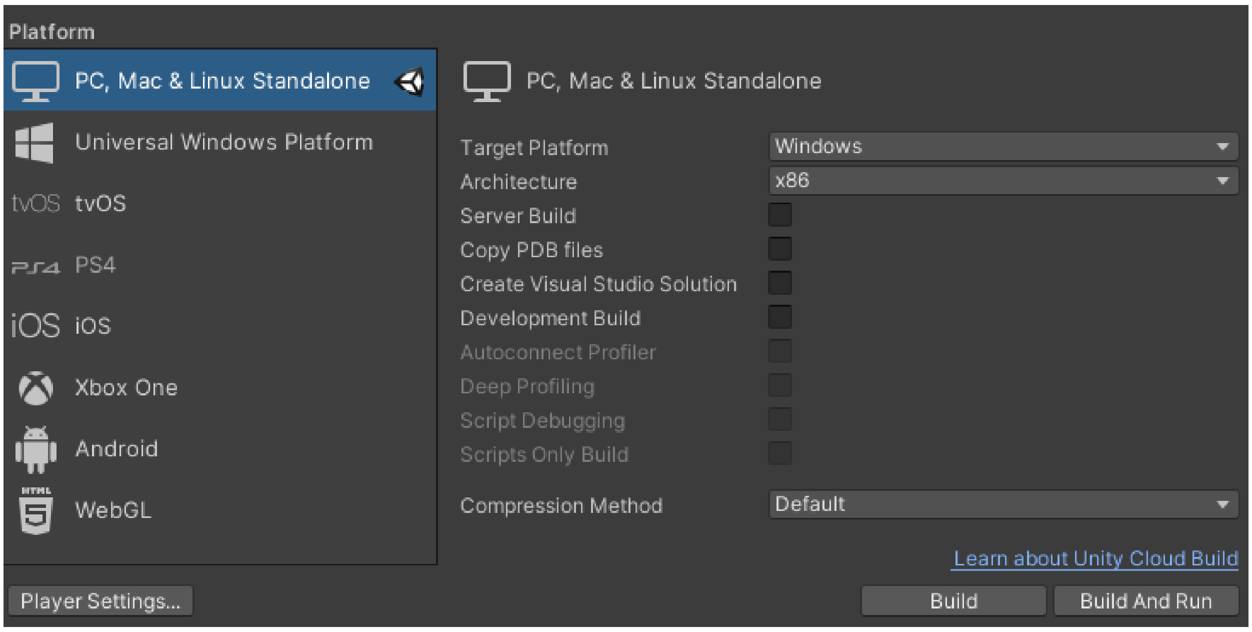
1. Create a tag on your Git repo on format "vX.Y.Z".



v2.3.5

Add loging settings
Create automatic build
Add version and date formatting on settings
Create asset settings
Create Git Utils v1
Initial Unity commit
Initial commit

2. Build your game



Platform

PC, Mac & Linux Standalone
Universal Windows Platform
tvOS
PS4
iOS
Xbox One
Android
WebGL

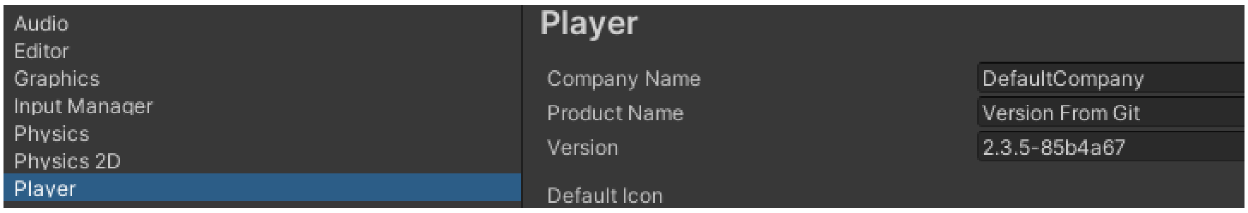
PC, Mac & Linux Standalone

Target Platform: Windows
Architecture: x86
Server Build
Copy PDB files
Create Visual Studio Solution
Development Build
Autoconnect Profiler
Deep Profiling
Script Debugging
Scripts Only Build
Compression Method: Default

Learn about Unity Cloud Build

Player Settings... Build Build And Run

3. And the version will be set to the Unity Player Version field!



Audio
Editor
Graphics
Input Manager
Physics
Physics 2D
Player

Player

Company Name: DefaultCompany
Product Name: Version From Git
Version: 2.3.5-85b4a67
Default Icon

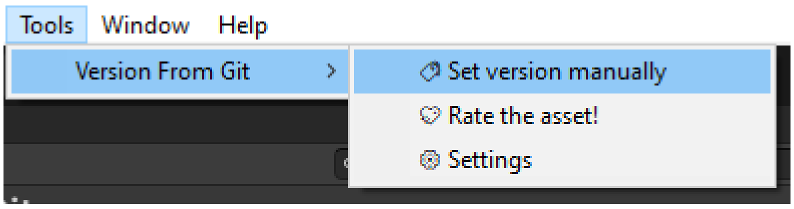
Usage

Automatically

If you have enabled the “Automatic On Build” setting the version will be set before each build. It will use the format settings for the version and the date. You don’t have to do anything.

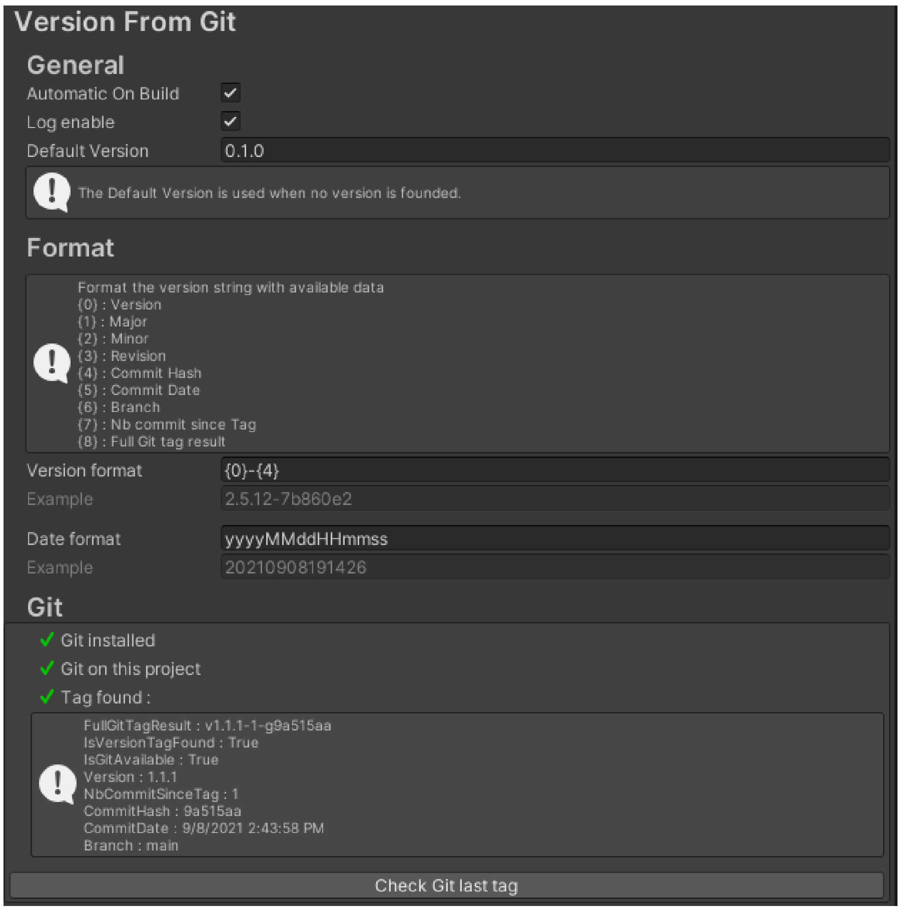
Manually

If you want to set the game version to the git version, you can use the option found on the Unity top menu “Tools/Version From Git/Set version manually”.



Settings

Asset’s settings could be found on “ProjectSettings/Version From Git” or by using the top menu “Tools/Version From Git/Settings”



Version From Git

General

Automatic On Build ✓

Log enable ✓

Default Version 0.1.0

The Default Version is used when no version is founded.

Format

```
Format the version string with available data
{0} : Version
{1} : Major
{2} : Minor
{3} : Revision
{4} : Commit Hash
{5} : Commit Date
{6} : Branch
{7} : Nb commit since Tag
{8} : Full Git tag result
```

Version format {0}-{4}

Example 2.5.12-7b860e2

Date format yyyyMMddHHmmss

Example 20210908191426

Git

✓ Git installed

✓ Git on this project

✓ Tag found :

```
FullGitTagResult : v1.1.1-1-g9a515aa
IsVersionTagFound : True
IsGitAvailable : True
Version : 1.1.1
NbCommitSinceTag : 1
CommitHash : 9a515aa
CommitDate : 9/8/2021 2:43:58 PM
Branch : main
```

Check Git last tag

General settings

Automatic On Build: If you want the Version to be fetch and assign on every Build. Work when building with command line too.

Log Enable: When assigning tag or using the asset it will log some information about git fetching tag or git availability. It could log some error breaking the build if no tag or no git is found. Disabled this option if you don't want to see any log or error.

Default Version: When no git tag is found the default version is used. You can customize it.

Format settings

Version format: You can customize the version format by using all possible variable specified in the help box above. If you want to reset to the initial format, you can press the reset button on the right side of the field. You'll see an example of your format on the disabled field below the version format field.

Variables available:

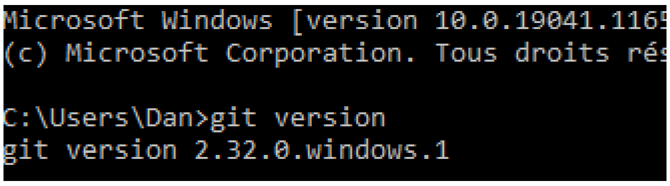
- {0} : Version on format « x.y.z » (« Major.Minor.Revision »)
- {1} : The major number
- {2} : The minor number
- {3} The revision number
- {4} Commit hash of the tagged commit (e.g., 9a515aa)
- {5} DateTime of the tagged commit (format of this date could be defined on the field below)
- {6} Name of the current branch
- {7} Quantity of commits since the last tag
- {8} The complete git tag result. You probably don't want to use it.

Date format: The custom DateTime format. It's the same as string parameter when using "SomeDate.ToString(dateFormatParameter)". You can find the official Microsoft documentation about date formatting here: 🔗https://docs.microsoft.com/en-us/dotnet/standard/base-types/custom-date-and-time-format-strings

You'll see an example of your current format on the disabled field below.

Git settings

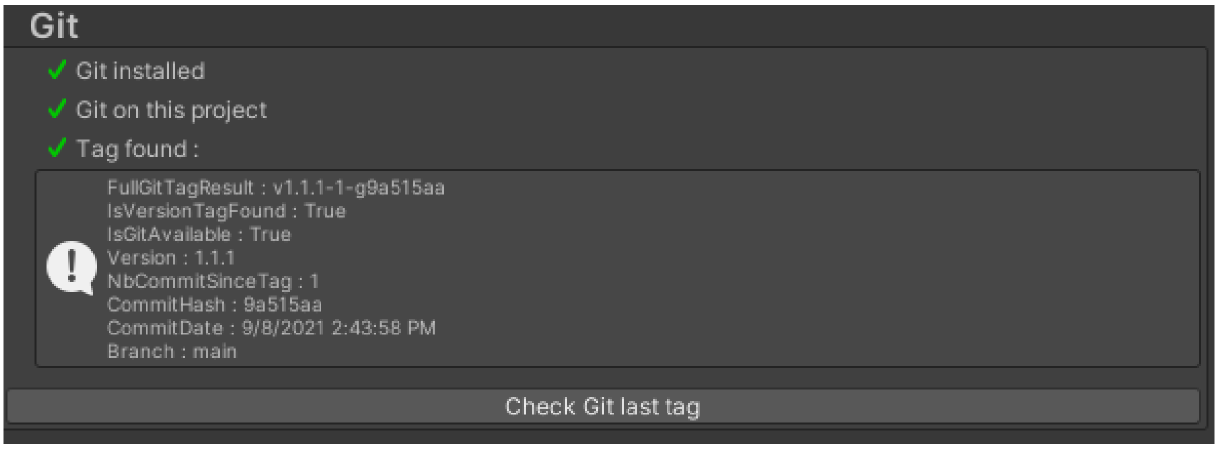
This section resumes if Git was found on your computer and if it's initialized for the current project. You should have two green checkmarks here. Otherwise, you should be sure that Git is installed and initialized for the current project. See the Git documentation for any further information about installing Git. For users on windows, be sure to have the Git path configured on your environment variable (don't forget to restart your computer after adding it). More information can be found on Google. If you want to check if the path is correctly installed just type "git version" on a command prompt and you should get this kind of output:



```
Microsoft Windows [version 10.0.19041.1165
(c) Microsoft Corporation. Tous droits rés

C:\Users\Dan>git version
git version 2.32.0.windows.1
```

You can use the "Check Git last tag" button to check the git availability and the last tag. You'll get a full report looking like this:



```
Git
✓ Git installed
✓ Git on this project
✓ Tag found :
    FullGitTagResult : v1.1.1-1-g9a515aa
    IsVersionTagFound : True
    IsGitAvailable : True
    Version : 1.1.1
    NbCommitSinceTag : 1
    CommitHash : 9a515aa
    CommitDate : 9/8/2021 2:43:58 PM
    Branch : main
```

i Note: Settings are saved as a *ScriptableObject* in the *Assets/Editor/VersionFromGitSettings.asset* file. Deleting this file will reset the settings and a new file will be created at the same location.

API usage

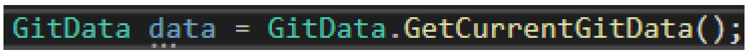
If you want to handle the formatting of the version yourself, you can disable the "Automatic On Build" and creating your own "IPreprocessBuildWithReport" script to handle the version. You could retrieve all git information and format it as you wish with your custom logic.

GitData

This class can be created by using static method:

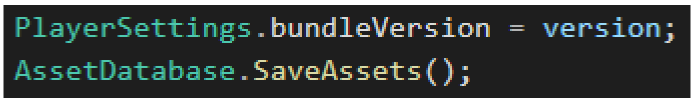
1. GitData.GetCurrentGitData(): Will have the last tag information and the current branch name.
2. GitData.GetExample(): Will give you an example of data who don't have any link with your project. You should not use this one.

You retrieve the current GitData information of your project like this:



```
GitData data = GitData.GetCurrentGitData();
```

This is the common way to assign the game version by script:



```
PlayerSettings.bundleVersion = version;
AssetDatabase.SaveAssets();
```

Custom IPreprocessBuildWithReport example

Here's an example/model how you can handle automatic versioning with your own logic for formatting it:

```
using NotInvited.VersionFromGit.Editor.Git;

public class ExampleVersionOnBuild : IPreprocessBuildWithReport
{
   public int callbackOrder { get; }
   public void OnPreprocessBuild(BuildReport report)
   {
       GitData data = GitData.GetCurrentGitData();

       PlayerSettings.bundleVersion = $"{data.Version.ToString()}-{data.CommitHash}-
       Preview-Android";
       AssetDatabase.SaveAssets();
    }
}
```

Support

📧 You can write to notinvitedgames@gmail.com if you have any question or suggestion. We are always looking for improving our asset and will taking in consideration any good idea!

💖 Thank you for buying this asset. If you are happy with it, feel free to rate us on the asset store 📎https://assetstore.unity.com/packages/slug/203089#reviews

V1.0.0