Build Versions Documentation

Contributors	Valid for Version	Last Updated
Jonathan Carter	1.0.1 & Higher	11/12/2021

Package Information

The package has the following folders & files, all listed with an asterisk and coloured **green** are required files for the asset to work. Those without are in red are not needed for the functionality but are required for some cosmetic features.

▼ **Carter Games/Build Versions/Editor**

- ***BuildInformationEditor.cs**
- ***BuildVersionsOptionsEditor.cs**
- ***BuildVersionsEditorWindow.cs**
- ***BuildVersionsManager.cs**

▼ **Carter Games/Build Versions/Resources**

- LogoBV.png
- Build Information Icon.png
- Build Options Icon.png

▼ **Carter Games/Build Versions/Scripts**

- ***BuildIncrementTime.cs**
- ***BuildInformation.cs**
- ***BuildVersionOptions.cs**

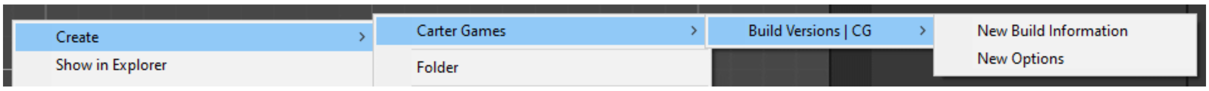
Change Log: Shows the changes from previous versions of the asset.

Documentation: A pdf of the documentation to help you use the asset.

Contents

Setup

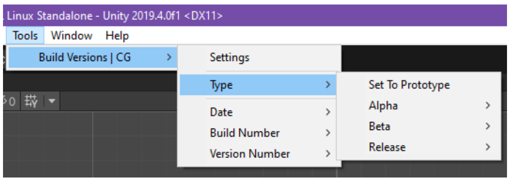
This asset is designed to help automate a normally mundane task that most developers forget to change when making builds. The asset requires no setup on your part as it will do this itself when you first make a build. If you want to setup the asset yourself, just create an instance of Build Information & Build Versions Options and store them somewhere in your project. These can be created via the Create Asset menu via the path showed below:

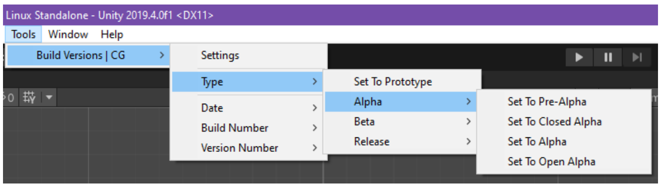


From here all you need to do is configure the asset to work as you wish. You can change the settings in the **options scriptable object** directly or via the convenient **settings editor window**.

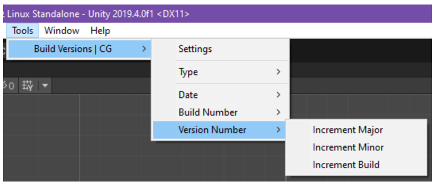
Changing Types

This is the only element of the asset that is not automated, as we can't read minds 😂 You can select a type of build via the tools menu. We've listed most common build types which we feel covers most of the types you will require. Should you want a different one, you can simply write it into the **Build Information** scriptable object.





Player Settings Systematic Versioning



When enabled the asset also updates the player settings for which ever platform you are currently on as well. The current version will not be transferred when you change platform. For the version to be correct on the new platform, you will need to set the version in the player settings to the last build's version to keep the version in sync.

The version number in the settings **MUST** follow the x.x.x or major/minor/build style formatting which is common in the industry. The asset only support 3 numbers separated by a dot (.), any other combinations will cause issues or errors.

By default the asset will only increment the build number by 1 each build made. Should you wish to update the major or minor number you can do so manually or via the tools menu options.

Scripting

Build Information

Properties

▼ Click To Expand

BuildNumber

Gets the current build number, this is the unique number not the player settings number.

Returns: Int

BuildDate

Gets the date that the last build was made as a string.

Returns: String

BuildType

Gets the "type" the build is set as

Returns: String

SystematicVersionNumber

Gets the systematic version in the player settings

Returns: String

UI Usage Example

You can use the build information asset to display the current build information in your game should you wish. Below is an example of a TextMeshPro UI display to show the current build number & date.

```
[SerializeField] private BuildInformation buildInformation;
private TMP_Text displayText;

private void Awake()
{
    displayText = GetComponent<TMP_Text>();
    displayText.text = $"Build: {buildInformation.BuildNumber} ({buildInformation.BuildDate})";
}
```

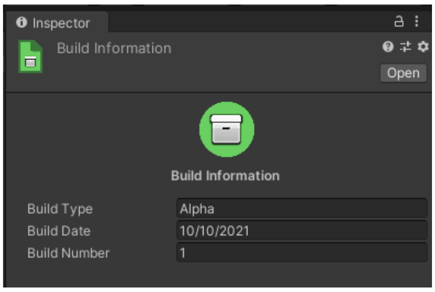
Below is another example in a different formatting and coding style.

```
private readonly string DatePrefix = "Date:";
private readonly string BuildPrefix = "Build:";

private TMP_Text buildText;

public BuildInformation buildInformation;

private void Awake()
{
    buildText = GetComponent<TMP_Text>();
    UpdateBuildText();
}

private void UpdateBuildText()
{
    var _string = string.Empty;

    _string += $"{DatePrefix} {buildInformation.BuildDate}\n";
    _string += $"{BuildPrefix} {buildInformation.BuildNumber} ({buildInformation.BuildType})";

    buildText.text = _string;
}
```

Inspector Breakdowns

Here are detailed explanations of each inspector and what each option is...

Build Information



Build Type: Alpha
Build Date: 10/10/2021
Build Number: 1

Summary

The build information scriptable object is the object that stores the data for the current build. This is what gets updated each build automatically with the latest information.

Breakdown

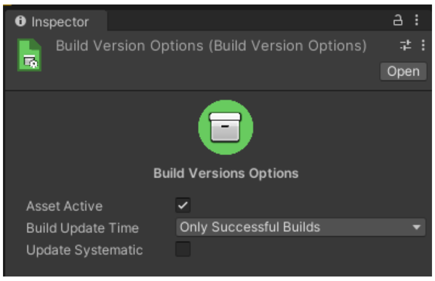
- Build Type - The type of build to display. This is manually set by you with the options in the Tools menu. The intention of this is to let users know if a build is a pre-release style of build such as an alpha or beta, or a full release.
- Build Date - This holds the date from the machine that made the build's system time. This gets updated automatically each time you make a build though you can set it manually if it doesn't update for some reason using the Tools menu.
- Build Number - This is the unique number for the build that is just a incremented automatically each time you make a build. It will also automatically adjust itself +/- 1 when changing the build number update time.

Build Versions Options

Summary

The build version options is the settings data object for the asset. Here you can change the settings that the asset uses when updating the build information.

Breakdown



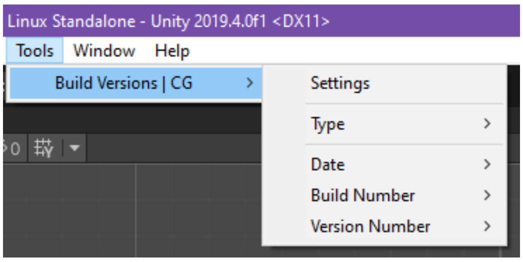
Build Versions Options

Asset Active ☑

Build Update Time: Only Successful Builds

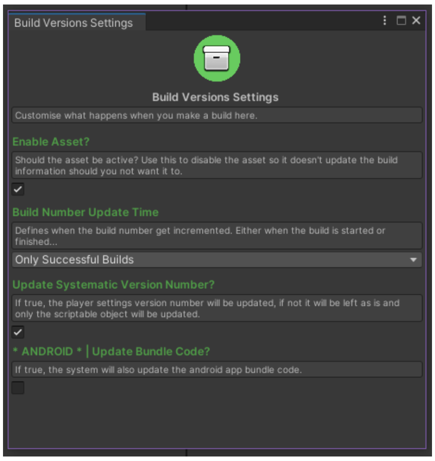
Update Systematic □

These values can all be edited in the settings window access via the path shown on the image below. We recommend using the editor window instead of this, however this will show the options that work across any platform.



- Asset Active - Should the asset be active? use this to toggle the asset on or off.
- Build Number Update Time - This changes when the build information is updated. You can choose between 2 options:
    - Any Build - Will update the build number as soon as you start a build, any will update the information even if the build fails along the way for some reason.
    - Only Successful Builds - Will update the build information only when a build is successfully made.
- Update Systematic Version Number - Toggles whether or not to change the x.x.x version number in the player settings when a build is made. This will only effect the last digit (the build/patch). If you want to change the major or minor you will need to edit it manually or via the Tools menu.

Settings Window



Build Versions Settings

Customise what happens when you make a build here.

Enable Asset?

Should the asset be active? Use this to disable the asset so it doesn't update the build information should you not want it to.

☑

Build Number Update Time

Defines when the build number get incremented. Either when the build is started or finished...

Only Successful Builds

Update Systematic Version Number?

If true, the player settings version number will be updated, if not it will be left as is and only the scriptable object will be updated.

☑

* ANDROID * | Update Bundle Code?

If true, the system will also update the android app bundle code.

□

Summary

The settings window allow you to configure the build version options data in a nicer format. This will also display extra options that are platform specific.

Breakdown

- Enable Asset? - This toggles whether or not the asset will function when you make a build.
- Build Number Update Time - This changes when the build information is updated. You can choose between 2 options:
    - Any Build - Will update the build number as soon as you start a build, any will update the information even if the build fails along the way for some reason.
    - Only Successful Builds - Will update the build information only when a build is successfully made.
- Update Systematic Version Number - Toggles whether or not to change the x.x.x version number in the player settings when a build is made. This will only effect the last digit (the build/patch). If you want to change the major or minor you will need to edit it manually or via the Tools menu.

- Update Bundle Code - This only appears if the build platform is set to Android for the project. Having this set to true will update the bundle code number for the build.

F.A.Q

How it works

The asset works by listening in to the pre & post build processes. These are automatically called by unity without needing any additional work. The asset updated the build information scriptable object at this time which keeps it up to date with the current builds.

The version number didn't update

If this happens, please let us know so we can investigate the issue. Do note however that the scriptable object values may not update if you have object open in the inspector. Instead you will find that you will need to close and open the object again to see the updated values. The asset can be toggled on and off, so please be sure that the Asset Active toggle is enabled for the version number to update.

The version number skipped a number

This can happen if you switch the build update time mid project. We currently don't have an automated fix for it but we are working on it for a future update. To mitigate this issue, simply manually change the build number down by 1 when switching to the Any Build type and up by one when switching to the Only Successful Builds type.

Got an issue that isn't listed?

Please get in touch with us so we can do our best to help you out. Email: support@carter.games