



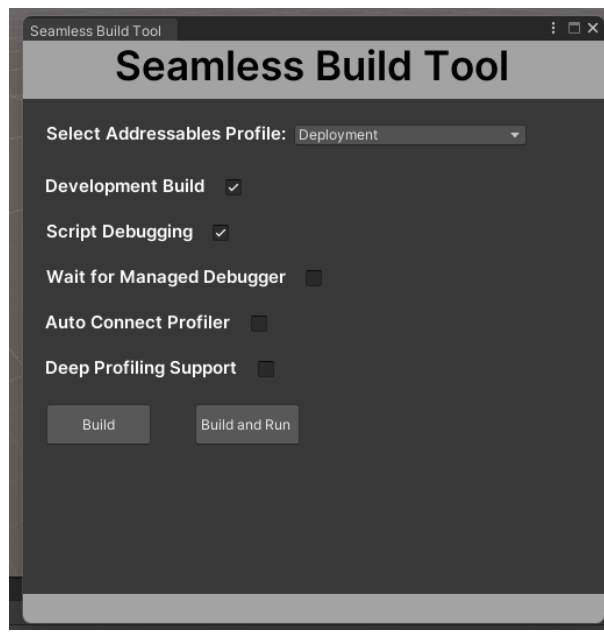
# **Arbelos Build Utility Package**

## **For Unity**

### **Guide**

By: [preet@arbelosinteractive.com](mailto:preet@arbelosinteractive.com)

## Package Features:



- The Seamless Build Tool automates the whole build process including the Addressables building and uploading process.
- The tool right now supports the five basic build options that are frequently used. Other settings must be set using the conventional project settings.
- The tool supports addressable profile switching between:

### **Deployment:**

- When using this profile, the player build will be made along with building and uploading addressables.

### **Editor Hosted:**

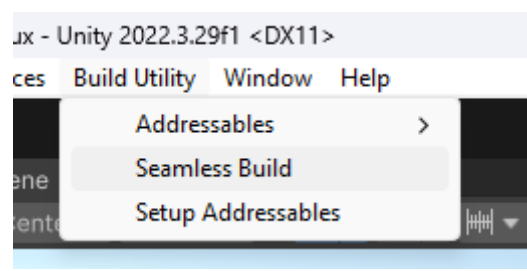
- This profile gives us the ability to skip uploading and downloading addressables in-game and use editor assets directly to quickly test local builds while developing a project. Note: Editor must be open for this.
- The package also includes a ready to use addressables downloader script

(AddressablesDownloader.cs) which can be attach in the game starting scene gameobject to download all the assets at the start of the game.

- You can also use the script as a base class to handle your own downloading ui logic by accessing parent variables like “number of assets downloaded”, “percentage of download completed” and there are also unity events that are called when an asset update is available, validation failed and Addressables successfully initialized (Listen to this unity event to load the next scene in the game).
- The addressables downloader script also does its own data validation to ensure all the assets are intact and successfully downloaded.

## **How to run Seamless Build Tool:**

- Under the menu items click: *Build Utility -> Seamless Build* to open the custom build tool.
- **NOTE: The tool will only work after all the necessary setup below is done.**



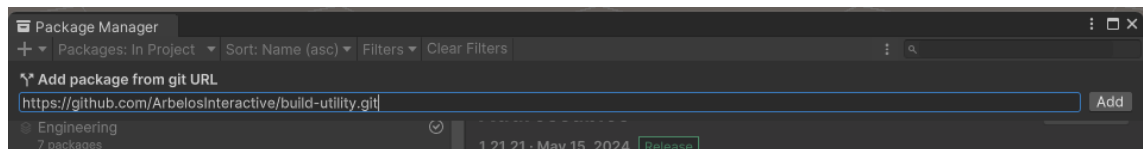
## Pre Requisites for the package:

- Unity Addressables Package must be pre-installed from the package manager.
- Azure Blob Storage Account must be setup with containers of different platforms named after the platforms (All lowercase). For example: "standalonewindows64", "standalonelinux64", "android", etc.
- Make sure to allow access to the containers.

## How to add the package in a unity project:

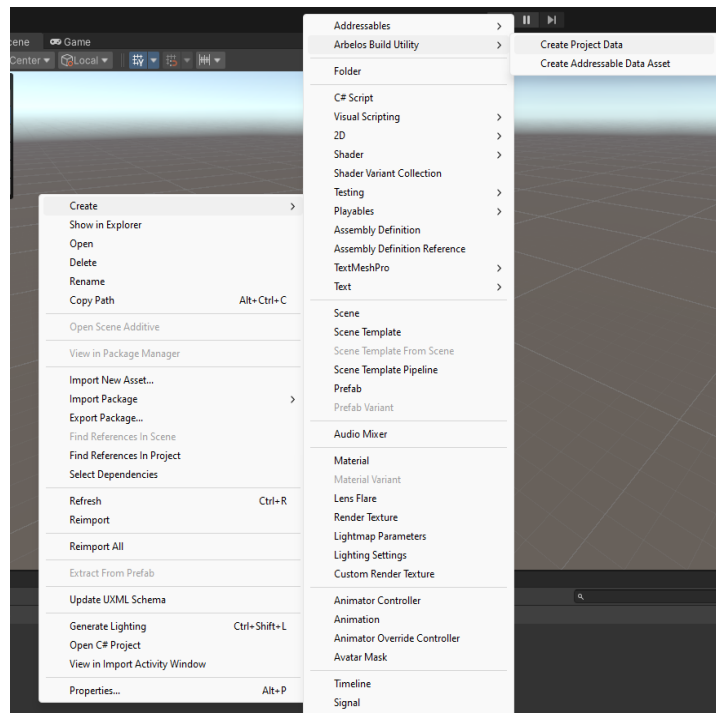
- The Arbelos build utility package can be imported through the package manager in your unity project and import a package by git URL and use the following URL:

<https://github.com/ArbelosInteractive/build-utility.git>



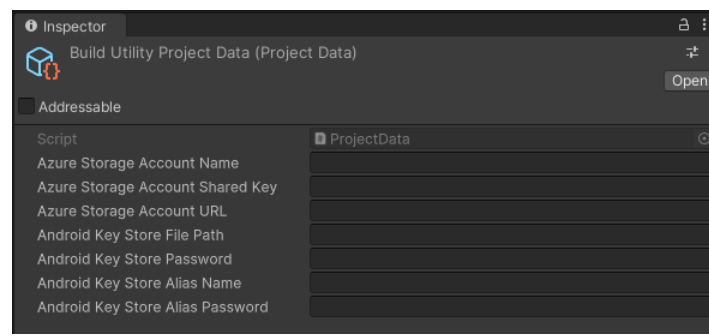
## Create and setup pre-required files:

- The very first step is to create the two required Scriptable Object Files under your project's 'Resources' folder.
- **NOTE:** There should only be one of each data files in the project resource directory and the names of the files should stay default.



## BuildUtilityProjectData.asset:

- In your project's Resources folder root, right click -> Create -> Arbelos Build Utility -> Create Project Data.
- The data file contains the following variables that needs to be set for the project's build pipeline:



- **Azure Storage Account Name:** The name of the azure storage account that has been setup already.
- **Azure Storage Account Shared Key:** Each azure storage account has two access keys that can be used for api calls. They can be found in the storage account settings under "*Security + networking -> Access Keys*". Enter one of the keys here.

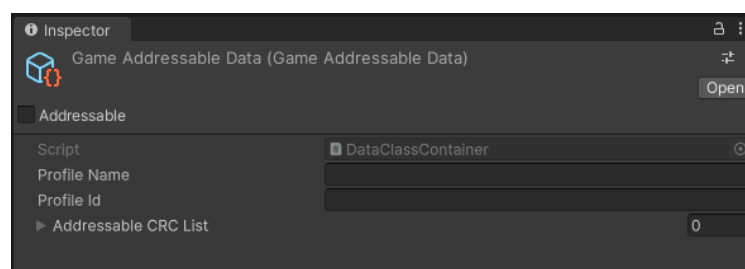
- **Azure Storage Account URL:** Each azure storage account has a specific URL and that is:

`https://[Azure Storage Account Name].blob.core.windows.net`

- **Android Keystore File Path:** If your project uses android builds, its keystore file's full path must be entered here, otherwise you can leave these 4 options empty.
- **Android Keystore Password:** Password for the keystore.
- **Android Keystore Alias Name:** The alias name assigned for the android keystore.
- **Android Keystore Alias Name:** The alias password assigned for the android keystore.

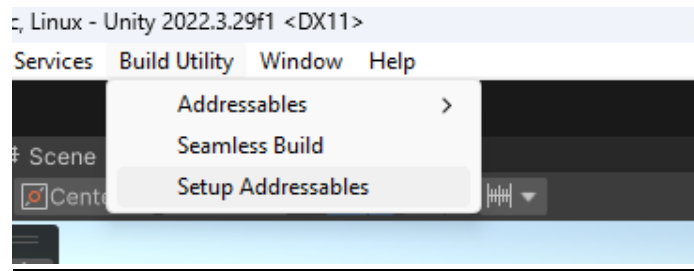
### **GameAddressableData.asset:**

- In your project's Resources folder root, right click -> Create -> Arbelos Build Utility -> Create Addressable Data Asset.
- This data file can be left empty.



## **Setup Addressable Settings:**

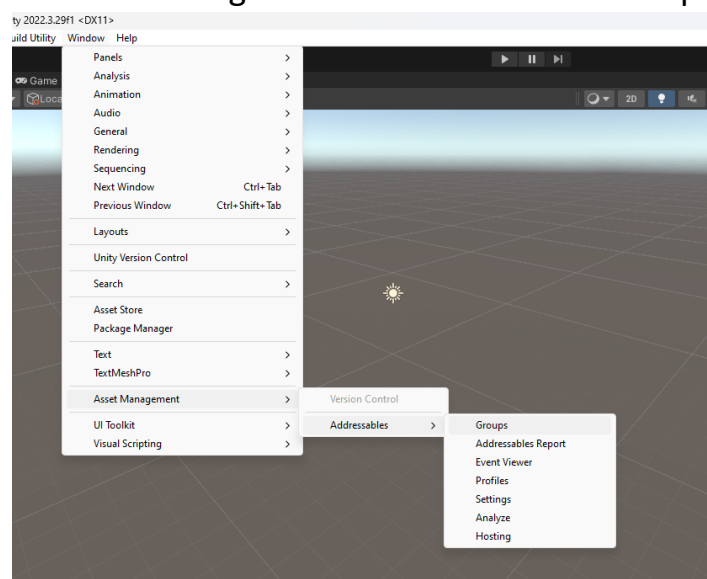
- If Addressables were previously setup, its recommended that you delete the "AddressableAssetsData" in the assets folder and do a clean setup.
- Under the menu items click: *Build Utility -> Setup Addressables*



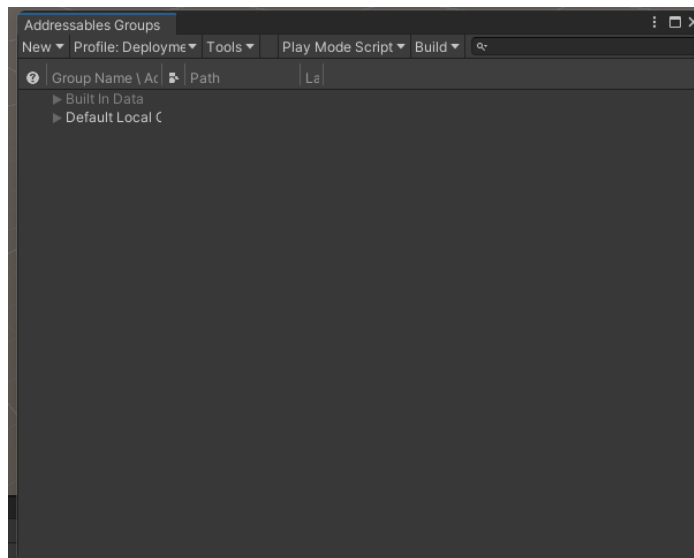
- This will create all the necessary profiles and services required for the tool for function properly.

## **Setup Addressable Asset Groups:**

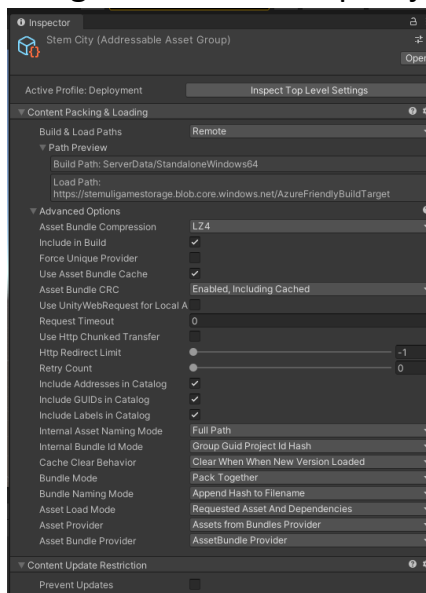
- For each project, addressable asset groups must be setup manually depending on the assets that needs to be Addressables for the particular project.
- Here's an example:
- Go to Window->AssetManagement->Addressables->Groups



You can create a new group using New->PackedAssets in the addressable groups window with a custom name or use the default one.

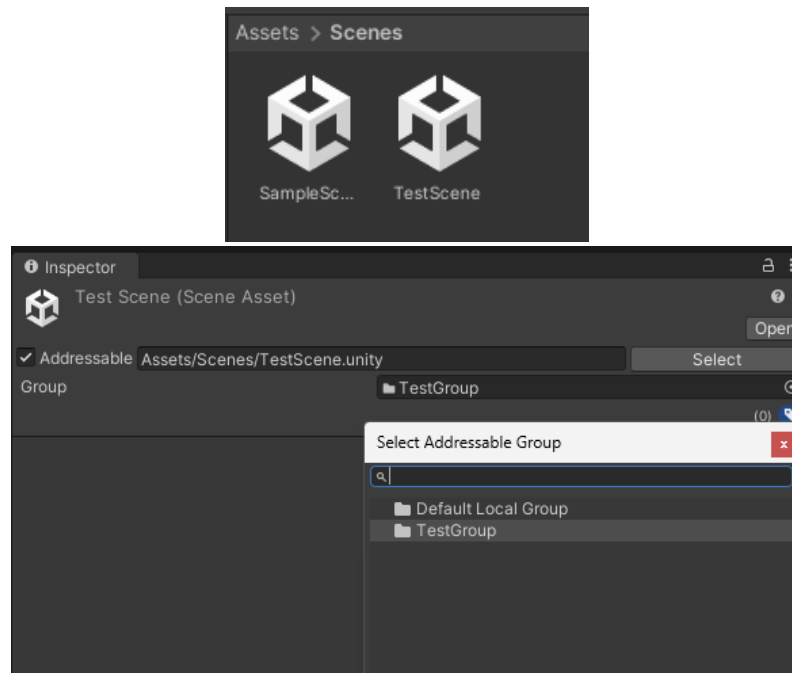


All the addressable groups used must have these similar settings. You can view the settings by clicking the Parent Group Object.

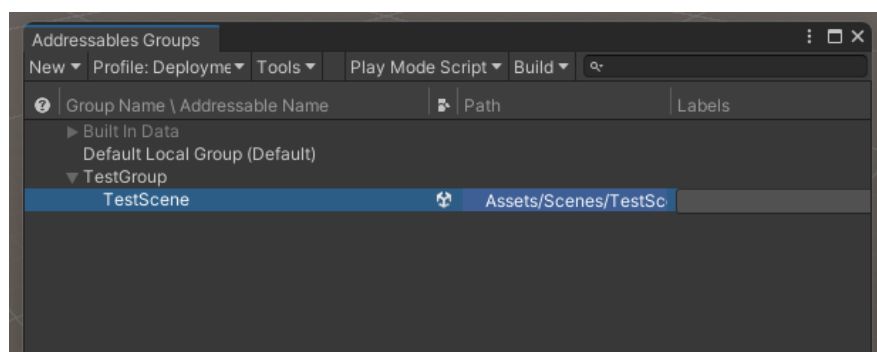


- Now, there's a "Test Scene" in the project that needs to be an addressable. Click on the test scene and in the inspector, check the addressable option for that asset and assign an addressable group to it.





- This should reflect back in the addressable groups window. Now select the scene and rename it (This name will be used to load the addressable assets. Eg: `Addressables.LoadSceneAsync("[TestScene]")`)



- This should be the end of the Addressables setup that must be done for the build tool and the automated flow to work properly.
- Feel free to contact me if you have any doubts.