AssetBundleManager

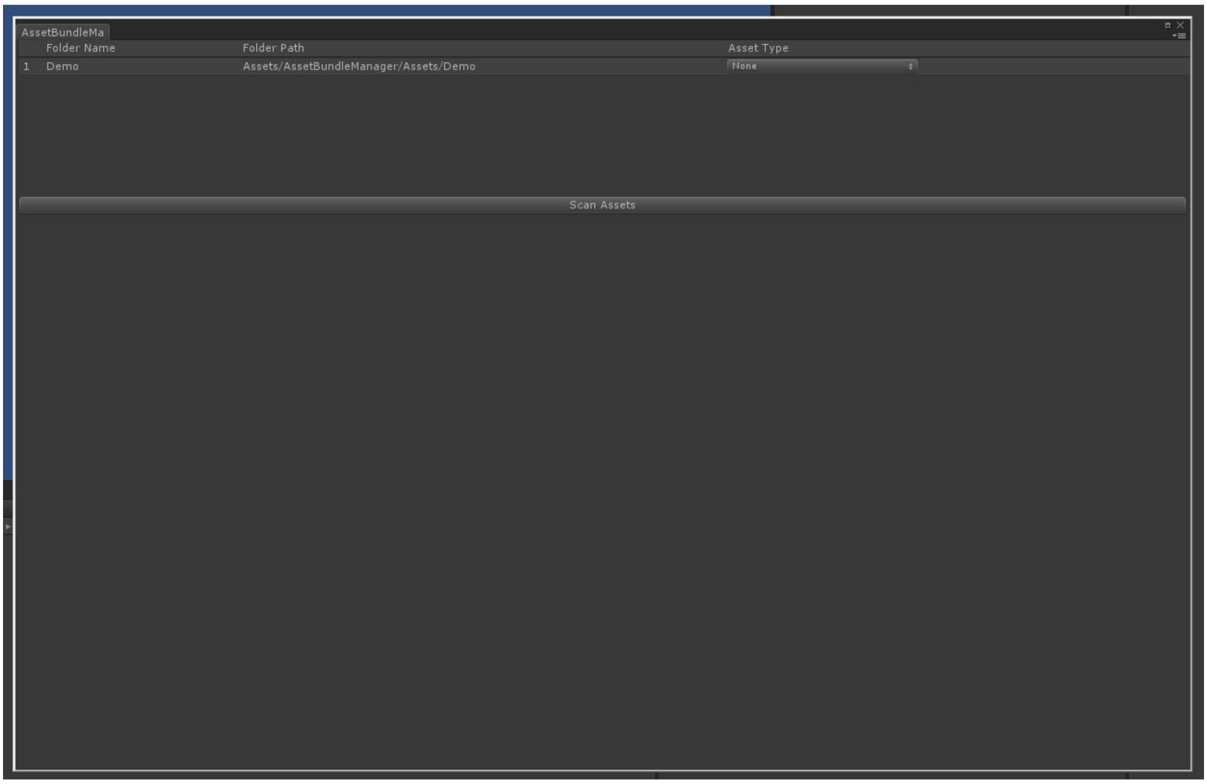
Easy-to-use tool for asset bundle

Features:

- One click scan-and-build assets bundles
- Load bundles in game with just one line of code
- Support platforms:
    - Android
    - iOS
    - Standalone (Windows, Linux, Mac OS X)

Usage Guideline:

1. Build Asset Bundles
    a. After importing “AssetBundleManager”, create folders under “/Assets/AssetBundleManager/Assets/”. You may name the folder as “Textures” or “Prefabs” or something meaningful to your game.
    b. Move the assets that going to be bundled into the folders just created
    c. Switch the build platform of Unity to which you want to build for, currently we support Android and iOS. Please note that bundles needed to be built for each platform once.
    d. In Unity editor’s top menu bar, open the asset bundle builder by “AssertBundleManager” → “Builder”
    e. Here is the builder window for asset bundle, you will see a list of folders created under “/Assets/AssetBundleManager/Assets/”.



	Folder Name	Folder Path	Asset Type
1	Demo	Assets/AssetBundleManager/Assets/Demo	None

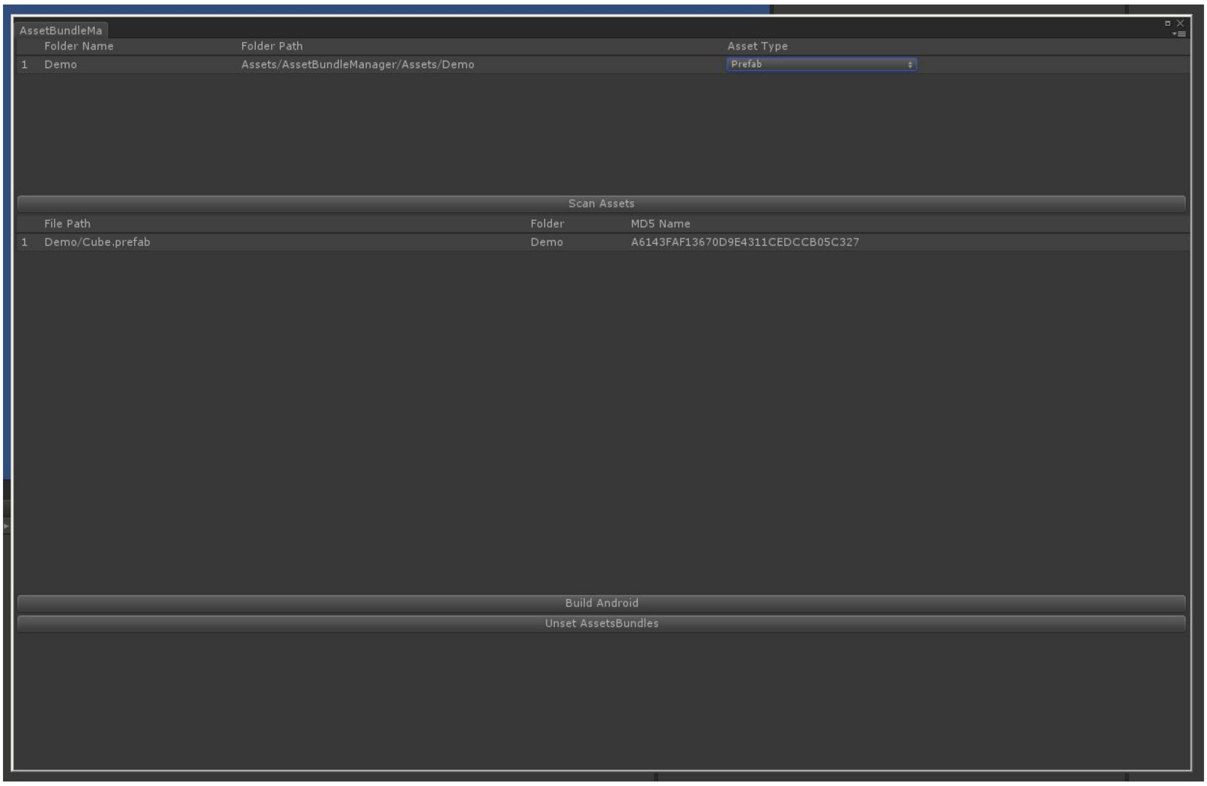
    f. Each folder will be scanned for an “Asset Type”, which can be:

i. “All”: All files under the folder will be listed out and will be bundled, like Material, Texture2D, Sprite, GameObject, etc

ii. “Prefab”: Only prefabs will be listed. For example, you have created a prefab of a character that contains a mesh, a material and a texture. All these files are placed under that folder, but only the prefab is needed to be bundled

iii. “None”: Nothing under this folder need to be bundled

g. After setting up the “Asset Type” of all folders, click the “Scan Assets” button. The builder will scan for each folder with its “Asset Type”. A list of assets will be shown after the scan completes



	Folder Name	Folder Path	Asset Type
1	Demo	Assets/AssetBundleManager/Assets/Demo	Prefab

	File Path	Folder	MD5 Name
1	Demo/Cube.prefab	Demo	A6143FAF13670D9E4311CEDCCB05C327

h. Now you may click ”Build Android” button the build asset bundles for those assets in the list. Once bundles are built, those assets will no longer embed with the game executables and need to be downloaded through codes, we will discuss about it later. Another button “Unset AssetsBundles” will delete all bundles built before, and those assets will now embedded with the game executables

i. After building the bundles, they can be found under “/Assets/AssetBundleManager/Bundles/” in a folder named with the platform (“Android”/”iOS”). Upload the whole folder **including the platform-named folder** to the network where can be accessed through http / https / ftp. Please note that ftp protocol support is limited to anonymous downloads only

j. Finally here is the last step, in Unity editor’s top menu bar, open setting file by “AssertBundleManager” → “Settings”, and then in the Inspector window, put the url of the asset bundle just uploaded. No need to include platform-named folder in the url

2. Download and use asset bundles in game
    a. Only one single function needed to call!

Method:
public ABLoadResult<T> Load<T>(string objName) where T : UnityEngine.Object

Parameters:

- objName: The path of the object going to load, for example, in order to load a prefab in file path “/Assets/AssetBundleManager/Assets/Prefabs/Object.prefab”, the objName will be “Prefabs/Object”

    b. After the method called, ABLoadResult<T> will be given, for loading prefabs, T will be GameObject. T can also be Texture2D or Sprite or Material etc, depends on what asset is being loaded
    c. The asset will start loading as soon as the method called. When the asset is ready, ABLoadResult<T>.isDone will become true. It is suggested that using a coroutine and wait for the asset is ready through a while loop and yield return null.
    d. The asset will be stored in ABLoadResult<T>.data, you may now use it whatever you want, just like have been loaded from resources folder.
    e. Downloaded assets are automatically saved in local device, calling Load<T> method again with same objName will be loaded from local cache.
    f. Optional: as network in client devices is not stable and always available, it is suggested that check network status before start loading (Application.internetReachability) and listen to delegate ABManager.OnConfDownloadFail and ABManager.OnABDownloadFail. Please refer to the documentation for the usage of these two delegates.

Example:

1. There is an example together with the package which load a prefab (a cube) to a scene. The prefab is placed in “/Assets/AssetBundleManager/Assets/Demo” together with its mesh and material.
2. Build the demo scene “/Assets/AssetBundleManager/Demo/AssetBundleManagerDemo.unity” to test on your device.

Support:
Please feel free to contact me via email: me@zicaleo.li

Copyright Notice:
AssetBundleManager uses **LitJson** C# library
https://lbv.github.io/litjson/

Version number: v1.1.0