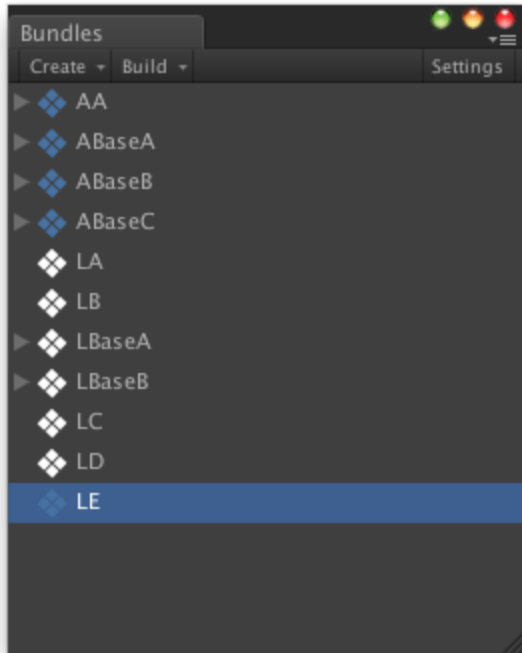# Learning The Interface of Bundle Manager

Bundle Manager is a Unity extension to provide an efficient solution for game streaming instead of hardcoding with Unity Build APIs.
This guide is an introduce to Bundle Manager's interface.

Manage Bundes

Use menu item **Window/Bundle Manager** to bring in Bundle Manager's main interface.

It looks similar to hierarchy window of Unity.

The main area of the window is a tree view of bundles. Different to the Hierarchy, this tree view show the dependencies of each bundles. The bundle on the root is depend by the bundle on the child.

There's two kind of bundles. The bundle with white icon is **Scene Bundle** which can only contains scene files**.** The bundle with blue icon is **Asset Bundle** which can contains everything except scenes files in Unity Project.

The tree view of bundles provides several operations same to Hierarchy Window. Drag bundles to change the dependencies between bundles. Rename the bundles by click it twice slowly. Delete the bundles by popup menu when right click. etc.

By default all bundles will be **build into one folder**. Plain file structure is fit for most projects.

But in some special case people want to put them into separate subfolders.
You can achieve this by add relative url to bundle name. For example, if you want put bundle A into subfolder F, you can rename A to F/A. Then bundle A can be build into subfolder F. And when loading it at runtime, you should use F/A as bundle's name but not just A.

Manage Bundle Content

Use the toolbar menu to create a new bundle. The type of bundle is defined at the moment you created it, there's no way to change it to another type of bundle. If you don't like the type you just created just delete it and create a new one.

Now you have the first bundle in bundle manager. You can check the content of the bundle from the inspector.
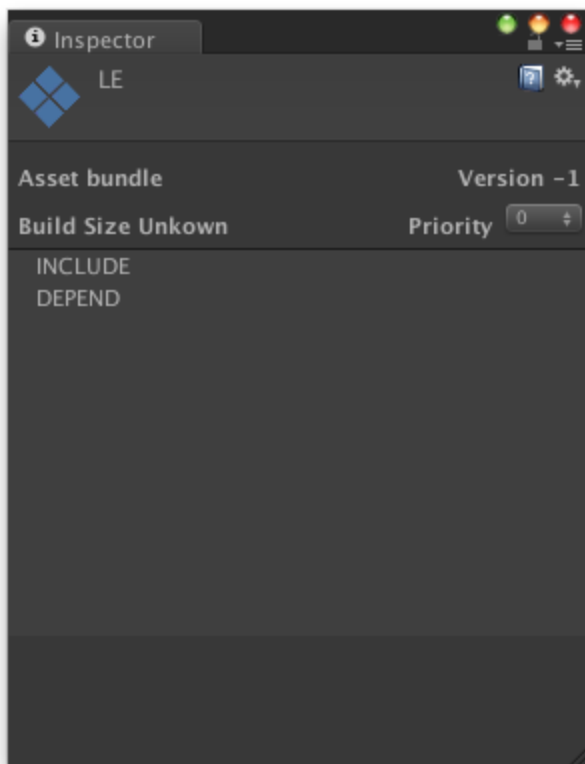
Top part of inspector shows the basic informations of the bundle.
● **Version:** The version number used for Unity cache system. Bundle Manager will increase the version number every time the bundle file get updated. The version informations is stored in BuildBundleData.txt. If you don't want to mess up the cache version on player's machine, be careful to update your BuildBundleData.txt.
● **Build Size**: Size of the bundle file built at last time.
● **Priority**: Default download priority of the bundle. Can be overwritten it at runtime.

The bottom part of inspector is the assets lists of the bundle.

INCLUDE is the assets you added into the bundle manually.
**DEPEND** is the depended assets of this bundle.

To **add assets to bundle** you can simply drag any assets or folders from project window into the item in the bundle hierarchy view.
Scene bundle receives paths contains only scene files, on the other hand asset bundle receives everything except scene files.
You can **remove included asset** by right click popup menu.

Manage Asset Dependencies

After you created several bundles. You might noticed some colorful indicators besides the assets like below.
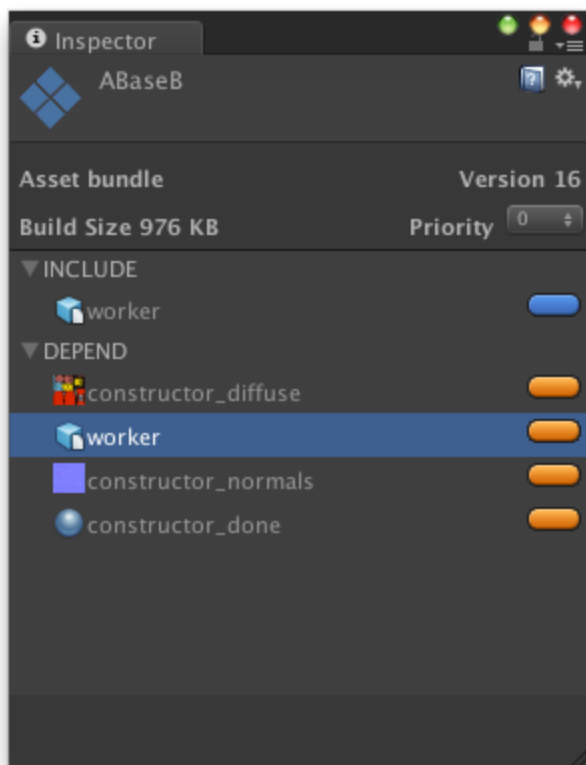
When the bundle have the same assets with other bundles. There will be a **share state indicator** shows on the asset item. These indicators will help you to locate the duplicate assets between your bundles.

⬭ is include or depend asset indicator. Means this asset exists in one of the parent bundles. When building the hierarchy, this bundle won't have a duplicated copy of this asset.

⬭ is depend asset indicator. Means this asset is referenced by some children bundles. The parent bundle will have a copy of this asset, and all bundles under this bundle will share this copy.

is include or depend asset indicator. Means this asset is duplicated with some other bundles without any kind of sharing. You should eliminate the red indicators in your project as much as you can to reduce the bundle size.

is depend asset indicator. Means this asset is fully or partly duplicated with some other bundles without any kind of sharing. You should eliminate the red indicators in your project as much as you can to reduce the bundle size.

Bundle Manager Configuration

Click the toolbar button **Settings** to show configurations of BundleManager.

There's several options for how to build and stream your bundles.
- **Compress:** When it's selected all bundle will build as compressed.
- **Deterministic:** In no deterministic mode, any bundle under the same root changed, the whole bundle tree will be rebuild. Deterministic bundles can help you decrease the build time by skipping the unchanged children bundle, but it will increase bundle loading time in runtime and the bundle size slightly.
- **Bundle Suffix:** The suffix of output bundle files.
- **Use Unity Cache:** Toggle this to open the Unity cache system for your game. Unit will cache your bundle by the version number. If you are running a web game, this option is not really necessary. You can rely on the browser's cache system which will work perfect if you know how to configure your http server's response headers.
- **Download Thread Limit:** This parameter defines how many download request you want to run at the same time.

- **Error Retry Time:** How many times you want to retry download when the request fails.
- **Use Editor Target:** When this toggle is checked, Build Target option will alway follow current active platform setting in Unity.
- **Build Target:** Current target platform to build bundles.
- **Output Path:** Where do you want to generate your bundle files. Different build target can have different output path.
- **Download Url:** The root url to download your bundles at runtime. When running in editor mode the download url will always be the current build target's output path.
- **Download from Output:** Option for test. When selected application will use the output url for downloading.
- **Export Config File:** This button allows you to export the Config file manually without rebuild the bundles.