**Loading and unloading objects from an AssetBundle**

**从AssetBundle 中加载和卸载对象**

Having created an AssetBundle object from the downloaded data, you can load the objects contained in it using three different methods:

根据下载的数据创建AssetBundle后，你可以通过三种方式从中加载对象：

* \* [AssetBundle.LoadAsset](http://docs.unity3d.com/540/Documentation/ScriptReference/AssetBundle.LoadAsset.html) will load an object using its name identifier as a parameter. The name is the one visible in the Project view. You can optionally pass an object type as an argument to the Load method to make sure the object loaded is of a specific type.

\* [AssetBundle.LoadAsset](http://docs.unity3d.com/540/Documentation/ScriptReference/AssetBundle.LoadAsset.html) 会通过标识名的方式作为参数来加载对象。名字就是Project试图中可见的那个。你可以可选择的传入一个对象类型作为参数来执行加载的命令来确保加载出来的对象是指定的类型。

* \* [AssetBundle.LoadAssetAsync](http://docs.unity3d.com/540/Documentation/ScriptReference/AssetBundle.LoadAssetAsync.html) works the same as the Load method described above but it will not block the main thread while the asset is loaded. This is useful when loading large assets or many assets at once to avoid pauses in your application.

\* [AssetBundle.LoadAssetAsync](http://docs.unity3d.com/540/Documentation/ScriptReference/AssetBundle.LoadAssetAsync.html) 和上面介绍的Load方法执行起来是一样的，但是他不会在加载资源的时候阻塞主线程。当你加载一个大的资源或者一次性加载很多资源时，使用这个方法可以有效地避免对你的应用程序造成卡顿。

* \* [AssetBundle.LoadAllAssets](http://docs.unity3d.com/540/Documentation/ScriptReference/AssetBundle.LoadAllAssets.html) will load all the objects contained in your AssetBundle. As with AssetBundle.Load, you can optionally filter objects by their type.

\* [AssetBundle.LoadAllAssets](http://docs.unity3d.com/540/Documentation/ScriptReference/AssetBundle.LoadAllAssets.html) 会加载你的AssetBundle中包含的所有对象。和AssetBundle.Load一样，你也可以可选择地通过对象的类型来过滤他们。

To unload assets you need to use [AssetBundle.Unload](http://docs.unity3d.com/540/Documentation/ScriptReference/AssetBundle.Unload.html). This method takes a boolean parameter which tells Unity whether to unload all data (including the loaded asset objects) or only the compressed data from the downloaded bundle. If your application is using some objects from the AssetBundle and you want to free some memory you can pass false to unload the compressed data from memory. If you want to completely unload everything from the AssetBundle you should pass true which will destroy the Assets loaded from the AssetBundle.

你需要使用[AssetBundle.Unload](http://docs.unity3d.com/540/Documentation/ScriptReference/AssetBundle.Unload.html)来卸载对象.这个方法使用了一个boolean类型的参数来告诉Unity是卸载所有的数据（包括已经加载的资源对象）还是仅仅是下载的资源包中的压缩数据。如果你的应用程序正在使用一些从AssetBundle中加载的对象而你又希望能够释放一些内存，那你可以传入false来从内存中释放那些压缩数据。如果你希望完全的卸载一切来源于AssetBundle的东西，你就应该传入true，这会摧毁那些从AssetBundle中加载的资源。

**Loading objects from an AssetBundles asynchronously**

**从AssetBundle中异步地加载对象**

You can use the [AssetBundle.LoadAssetAsync](http://docs.unity3d.com/540/Documentation/ScriptReference/AssetBundle.LoadAssetAsync.html) method to load objects Asynchronously and reduce the likelihood of having hiccups in your application.

你可以使用[AssetBundle.LoadAssetAsync](http://docs.unity3d.com/540/Documentation/ScriptReference/AssetBundle.LoadAssetAsync.html) 方法来异步的加载对象，以降低应用程序的卡顿感。

using UnityEngine;

// Note: This example does not check for errors. Please look at the example in the DownloadingAssetBundles section for more information

IEnumerator Start () {

while (!Caching.ready)

yield return null;

// Start a download of the given URL

WWW www = WWW.LoadFromCacheOrDownload (url, 1);

// Wait for download to complete

yield return www;

// Load and retrieve the AssetBundle

AssetBundle bundle = www.assetBundle;

// Load the object asynchronously

AssetBundleRequest request = bundle.LoadAssetAsync ("myObject", typeof(GameObject));

// Wait for completion

yield return request;

// Get the reference to the loaded object

GameObject obj = request.asset as GameObject;

// Unload the AssetBundles compressed contents to conserve memory

bundle.Unload(false);

// Frees the memory from the web stream

www.Dispose();

}