# Streaming at Runtime

After create your bundles. The first question might comes up is how can I get them at runtime. This guide is a short walkthough for you to start streaming your own project.

## Url Setup

Before start writing your test code, you need to set up your download urls.
Your url can be any protocols by which WWW support.

There's a default url set in configuration panel for every platform.

```
$(DataPath)/../AssetBundle/$(Platform)
```

The $(xxx) is some thing similar to the environment variables for urls in Visual Studio. They will be replaced by the real path string when evaled at runtime.

Bundle Manager provides several environment variables.

- **DataPath** Application.dataPath
- **PersistentDataPath** Application.persistentDataPath
- **StreamingAssetsPath** Application.streamingAssetsPath
- **Platform** Name string of current target platform

**Url Redirect**

In some special case one download url for one platform is not enough. BundleManager provides a way to redirect your download target.
Make a new file named BMRedirect.txt and put it in to the download url. When DownloadManager initialize, it will try to get BMRedirect.txt first from the download url. If there's a BMRedirect.txt, DownloadManager's target url will follow the url in it.

## Server Setup

If you are testing in the editor and using the default url settings, you can skip this step, because the output url and the download url are pointing to the same position.

To test your app in other environment, you need to make sure the downloading target is ready. Copy all the files in the output folder into your download position. (In version 1.1.5 or above, the *.txt files are not necessary, but you may want to keep them for data checking purpose.)

If you are making a web game. Please make sure your cache headers are properly configured. Some wrong cache setting can lead your application alway get the old version of bundles.

If you are making a web game, and you deploy the application file separately from bundles on

different hosts, please make sure you have set up the crossdomain.xml.

## Coding

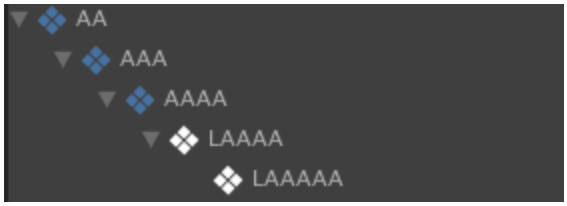
After all the things set up. We can start code to streaming your project.
We provides DownloadManager for runtime streaming work. It's very simple as using the WWWs. Here's a short snippet:

```
using UnityEngine;

public class TestDownloadManager : MonoBehaviour
{
   IEnumerator Start()
   {
      // You should only use the file name to access your bundle but not the full url.
      yield return StartCoroutine( DownloadManager.Instance.WaitDownload("cube.assetBundle") );

      WWW www = DownloadManager.Instance.GetWWW("cube.assetBundle");
      var bundle = www.assetBundle;

      GameObject cube = Instantiate(bundle.Load("cube")) as GameObject;
   }
}
```

The code above create a cube from bundle.

You don't need to care about how the bundle structured in the Bundle Tree View, or the priorities between bundles, DownloadManager will deal with them automatically.

For example, there's a bundle tree like below.




If you only want the last leaf bundle LAAAAA. You can just start one request.

```
yield return StartCoroutine( DownloadManager.Instance.WaitDownload("LAAAAA.assetBundle") );
```

The DownloadManager will prepare all the dependent bundles before LAAAAA.

If you get several bundle to download. Some of them you want get before others. But it's Annoying to handle the download routine one by one.
You can just set up the priorities of them. Then at runtime, you request them at once. The DownloadManager will make sure they are download by the priorities as you like.

For more information please check API Reference.