

Easy Content Delivery Network

1.2.0

Generated by Doxygen 1.10.0

1 Easy Content Delivery Network for Unity	1
1.1 Documentation	1
2 Namespace Index	3
2.1 Package List	3
3 Hierarchical Index	5
3.1 Class Hierarchy	5
4 Class Index	7
4.1 Class List	7
5 Namespace Documentation	9
5.1 SplenSoft Namespace Reference	9
5.2 SplenSoft.AssetBundles Namespace Reference	9
6 Class Documentation	11
6.1 SplenSoft.AssetBundles.AssetBundleData Class Reference	11
6.1.1 Detailed Description	12
6.1.2 Member Function Documentation	12
6.1.2.1 Flush()	12
6.2 SplenSoft.AssetBundles.AssetBundleManager Class Reference	12
6.2.1 Detailed Description	14
6.2.2 Member Function Documentation	14
6.2.2.1 DoesBucketExist()	14
6.2.2.2 GetAsset< T >()	14
6.2.2.3 GetAssetBundle()	15
6.2.2.4 GetAssetBundleNames() [1/2]	15
6.2.2.5 GetAssetBundleNames() [2/2]	15
6.2.2.6 GetDependencies()	16
6.2.2.7 GetManifest()	16
6.2.2.8 LoadSceneAsssetBundle()	16
6.2.2.9 RunCliCommands()	17
6.2.2.10 SetSelfInitializerTimeout()	17
6.2.2.11 TryGetAssetBundleData()	17
6.2.2.12 TryGetAssetBundleName()	18
6.2.2.13 TryGetAssetManifestRetrievalResult()	18
6.2.2.14 TryGetAssetRetrievalResult()	18
6.3 SplenSoft.AssetBundles.AssetBundleManagerSettings Class Reference	19
6.3.1 Detailed Description	19
6.3.2 Member Function Documentation	20
6.3.2.1 Get()	20
6.4 SplenSoft.AssetBundles.AssetBundleReferenceAttribute Class Reference	20
6.4.1 Detailed Description	20

6.5 SplenSoft.AssetBundles.AssetRequester Class Reference	20
6.5.1 Detailed Description	21
6.6 SplenSoft.AssetBundles.AssetRetrievalProgress Class Reference	21
6.6.1 Detailed Description	22
6.7 SplenSoft.AssetBundles.AssetRetrievalResult Class Reference	22
6.7.1 Detailed Description	22
6.8 SplenSoft.AssetBundles.AutoInstantiator Class Reference	22
6.8.1 Detailed Description	22
6.9 SplenSoft.AssetBundles.IPreprocessAssetBundle Interface Reference	22
6.9.1 Detailed Description	23
6.10 SplenSoft.AssetBundles.ManagedAssetAttribute Class Reference	23
6.10.1 Detailed Description	23
6.11 SplenSoft.AssetBundles.Requester< T > Class Template Reference	23
6.11.1 Detailed Description	24
6.11.2 Member Function Documentation	24
6.11.2.1 GetAsset()	24
6.11.2.2 GetAssetBundle()	24
6.11.2.3 TryGetAssetRetrievalResult()	24
6.12 SplenSoft.AssetBundles.SceneRequester Class Reference	25
6.12.1 Detailed Description	26
6.12.2 Member Function Documentation	26
6.12.2.1 DownloadAndLoadSceneAsync()	26
Index	27

Chapter 1

Easy Content Delivery Network for Unity

This is an addressables-free plugin that automatically handles the Unity Cloud Content Delivery API and packaging/deploying assetbundles while providing several tools to make asset management easy.

EZ-CDN handles the tedium of API connections, client line interfaces and cataloging assets for you. Just design your prefabs, flesh out your scenes and let us do all the heavy lifting.

You can [download the package on the Unity Asset Store](#).

1.1 Documentation

- [Why use it?](#)
- [Get Started](#)
- [Designing](#)
- [Publishing](#)
- [API Reference](#)
- [Limitations](#)

Chapter 2

Namespace Index

2.1 Package List

Here are the packages with brief descriptions (if available):

SplenSoft	9
SplenSoft.AssetBundles	9

Chapter 3

Hierarchical Index

3.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

SplnSoft.AssetBundles.AssetBundleData	11
SplnSoft.AssetBundles.AssetBundleManager	12
SplnSoft.AssetBundles.AssetBundleManagerSettings	19
SplnSoft.AssetBundles.AssetBundleReferenceAttribute	20
SplnSoft.AssetBundles.AssetRetrievalProgress	21
SplnSoft.AssetBundles.AssetRetrievalResult	22
SplnSoft.AssetBundles.AutoInstantiator	22
SplnSoft.AssetBundles.IPreprocessAssetBundle	22
SplnSoft.AssetBundles.ManagedAssetAttribute	23
SplnSoft.AssetBundles.Requester< T >	23
SplnSoft.AssetBundles.Requester< GameObject >	23
SplnSoft.AssetBundles.Requester< UnityEngine.Object >	23
SplnSoft.AssetBundles.AssetRequester	20
SplnSoft.AssetBundles.SceneRequester	25

Chapter 4

Class Index

4.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

SplenSoft.AssetBundles.AssetBundleData	
Data instantiated by AssetBundleManager on initialization. Each asset bundle in the AssetBundleManifest will have its own AssetBundleData entry	11
SplenSoft.AssetBundles.AssetBundleManager	
Static class that manages asset bundle retrieval and packaging	12
SplenSoft.AssetBundles.AssetBundleManagerSettings	
Controls the settings for EZ-CDN	19
SplenSoft.AssetBundles.AssetBundleReferenceAttribute	
Apply this attribute to a SerializeField string field to reference an asset bundle name with the mask of a normal Unity property drawer of the supplied type	20
SplenSoft.AssetBundles.AssetRequester	
MonoBehaviour component that can request an asset from the CDN. Accepts any UnityEngine.Object as an asset. Cannot instantiate or further manipulate the asset without the help of code. For codeless or more specific usages, try SceneRequester	20
SplenSoft.AssetBundles.AssetRetrievalProgress	
A trackable progress object that can be passed to AssetBundleManager.GetAsset and AssetBundleManager.GetAssetBundle through a Progress<T> object	21
SplenSoft.AssetBundles.AssetRetrievalResult	
Stores the results of the last attempt to download an asset bundle from the CDN	22
SplenSoft.AssetBundles.AutoInstantiator	
Automatically downloads and instantiates prefabs on app start	22
SplenSoft.AssetBundles.IPreprocessAssetBundle	
Used to add custom processing to an asset before it is packaged by EZ-CDN	22
SplenSoft.AssetBundles.ManagedAssetAttribute	
Add this attribute to a custom class to include it in the AssetBundleManager operations. Useful for custom UnityEngine.ScriptableObject. You can also add the class to the list of managed types in the SplenSoft -> Asset Bundles -> Settings	23
SplenSoft.AssetBundles.Requester< T >	
Base class for asset bundle requesters of various objects or scenes	23
SplenSoft.AssetBundles.SceneRequester	
MonoBehaviour component that can request a scene asset from the CDN. Accepts any UnityEditor.SceneAsset as an asset	25

Chapter 5

Namespace Documentation

5.1 SplenSoft Namespace Reference

5.2 SplenSoft.AssetBundles Namespace Reference

Classes

- class [AssetBundleData](#)
Data instantiated by AssetBundleManager on initialization. Each asset bundle in the AssetBundleManifest will have its own AssetBundleData entry.
- class [AssetBundleManager](#)
Static class that manages asset bundle retrieval and packaging.
- class [AssetBundleManagerSettings](#)
Controls the settings for EZ-CDN.
- class [AssetBundleReferenceAttribute](#)
Apply this attribute to a SerializeField string field to reference an asset bundle name with the mask of a normal Unity property drawer of the supplied type.
- class [AssetRequester](#)
MonoBehaviour component that can request an asset from the CDN. Accepts any UnityEngine.Object as an asset. Cannot instantiate or further manipulate the asset without the help of code. For codeless or more specific usages, try SceneRequester
- class [AssetRetrievalProgress](#)
A trackable progress object that can be passed to AssetBundleManager.GetAsset and AssetBundleManager.GetAsyncAssetBundle through a Progress<T> object.
- class [AssetRetrievalResult](#)
Stores the results of the last attempt to download an asset bundle from the CDN.
- class [AutoInstantiator](#)
Automatically downloads and instantiates prefabs on app start.
- interface [IPreprocessAssetBundle](#)
Used to add custom processing to an asset before it is packaged by EZ-CDN.
- class [ManagedAssetAttribute](#)
Add this attribute to a custom class to include it in the AssetBundleManager operations. Useful for custom UnityEngine.ScriptableObject. You can also add the class to the list of managed types in the [SplenSoft](#) -> Asset Bundles -> Settings.
- class [Requester](#)
Base class for asset bundle requesters of various objects or scenes.
- class [SceneRequester](#)
MonoBehaviour component that can request a scene asset from the CDN. Accepts any UnityEditor.SceneAsset as an asset.

Chapter 6

Class Documentation

6.1 SplenSoft.AssetBundles.AssetBundleData Class Reference

Data instantiated by AssetBundleManager on initialization. Each asset bundle in the AssetBundleManifest will have its own AssetBundleData entry.

Public Member Functions

- void **Flush** (bool unloadAllLoadedObjects=false)
Removes the AssetBundle and Asset from this cache so it can be garbage collected. Unloads the asset bundle from memory.

Properties

- List< string > **Dependencies** = new List<string>() [get, set]
List of asset bundle names that this asset bundle depends on. These should be loaded first, and will be loaded first automatically if the asset is requested through the AssetBundleManager
- Object **Asset** [get, set]
The actual asset that has been cached after successful retrieval. Can be used to instantiate the asset any time, as long as it's not null and as long as you know the type. Check Loaded to ensure the asset has been unpacked by the AssetBundleManager at least once to ensure this is not null.
- AssetBundle **AssetBundle** [get, set]
The asset bundle that has been cached after successful retrieval. Can be used to unpack its asset at any time, as long as its not null.
- bool **Loaded** [get, set]
Check this value to ensure the Asset is not null.
- bool **DownloadStarted** [get, set]
True if at least one attempt has been made to retrieve this asset.
- Hash128 **Hash** [get, set]
Used for caching.
- long **LastResponseCode** [get, set]
The last response code from a download attempt. Returns 0 if no download attempt was made, and can return 200 even if it was loaded from cache.

6.1.1 Detailed Description

Data instantiated by `AssetBundleManager` on initialization. Each asset bundle in the `AssetBundleManifest` will have its own `AssetBundleData` entry.

6.1.2 Member Function Documentation

6.1.2.1 Flush()

```
void SplenSoft.AssetBundles.AssetBundleData.Flush (
    bool unloadAllLoadedObjects = false )
```

Removes the `AssetBundle` and `Asset` from this cache so it can be garbage collected. Unloads the asset bundle from memory.

Parameters

<code>unloadAllLoadedObjects</code>	If true, all objects loaded from this asset bundle will also be unloaded. Use only if you're sure none of this asset bundle's assets are in the scene
-------------------------------------	---

The documentation for this class was generated from the following file:

- `C:/Development Projects/for-the-birds/Packages/com.splensoft.ezcdn/Runtime/AssetBundleData.cs`

6.2 SplenSoft.AssetBundles.AssetBundleManager Class Reference

Static class that manages asset bundle retrieval and packaging.

Static Public Member Functions

- static bool [TryGetAssetBundleData](#) (string assetBundleName, out [AssetBundleData](#) data)
Attempts to return `AssetBundleData` for an asset bundle. The `AssetBundleManager` must be initialized for the manifest data to be available.
- static bool [TryGetAssetRetrievalResult](#) (string assetBundleName, out [AssetRetrievalResult](#) result)
Gets status of last download attempt for an asset. Note: If an asset was pulled from the cache, it will return 200 with `UnityWebRequest.Result.Success`. To get the result of the initialization attempt (downloading the manifest), use `TryGetAssetManifestRetrievalResult`
- static bool [TryGetAssetManifestRetrievalResult](#) (out [AssetRetrievalResult](#) result)
Gets status of last download attempt of the project asset bundle manifest for this platform. Note: If the plugin is set to use editor assets and this is running in an editor, it will return 200 with `UnityWebRequest.Result.Success`.
- static async Task< [AssetBundleManifest](#) > [GetManifest](#) ()
Downloads and unpacks the `AssetBundleManifest` for this platform from the CDN.
- static void [SetSelfInitializerTimeout](#) (float timeInSeconds)
Sets a time, in seconds, that the Asset Bundle Manager will attempt to auto-initialize.
- static async void **Initialize** ()
Downloads the project asset bundle manifest for this platform and caches the dependency data. Called automatically on app start unless `AutoInitialize` is set to false on app awake.
- static async Task< bool > [DoesBucketExist](#) (string bucketId)

Uses client-facing Unity Cloud Content Delivery API to determine if a bucket exists. Requires an internet connection.

- static async Task< string[] > [GetAssetBundleNames](#) (string regexPattern)
If runtime, requires plugin to be initialized and Initialized must be true. Scans manifest for all asset bundle names. If running in the editor with Use Editor Assets checked in the settings, it will scan the project's AssetDatabase
- static async Task< string[] > [GetAssetBundleNames](#) (Type type)
If runtime, requires plugin to be initialized and Initialized must be true. Scans manifest for all asset bundle names. If running in the editor with Use Editor Assets checked in the settings, it will scan the project's AssetDatabase
- static async Task< T > [GetAsset< T >](#) (string name, IProgress< [AssetRetrievalProgress](#) > progress=null, Action< T > onSuccess=null, Action< [AssetRetrievalResult](#) > onFailure=null, bool waitForInitialize=true)
Downloads an asset bundle from the Unity Content Delivery cloud and unpacks the asset.
- static async Task< AssetBundle > [GetAssetBundle](#) (string name, IProgress< [AssetRetrievalProgress](#) > progress=null, Action< AssetBundle > onSuccess=null, Action< [AssetRetrievalResult](#) > onFailure=null, bool waitForInitialize=true)
Downloads an asset bundle from the Unity Content Delivery cloud.
- static async Task [LoadSceneAssetBundle](#) (string name, IProgress< [AssetRetrievalProgress](#) > progress=null, Action onSuccess=null, Action< [AssetRetrievalResult](#) > onFailure=null, bool waitForInitialize=true)
Downloads a scene asset bundle from the Unity Content Delivery cloud and immediately loads it.
- static async Task [GetDependencies](#) (string assetBundleName, IProgress< [AssetRetrievalProgress](#) > progress=null)
Downloads all of an AssetBundle's dependencies and keeps them cached. Good for loading ahead to make downloading a later asset bundle quicker.
- static bool [TryGetAssetBundleName](#) (UnityEngine.Object obj, out string assetBundleName)
Attempts to retrieve or generate asset bundle name. Will fail if asset type is not defined in EZCDN settings.
- static void [RunCliCommands](#) (List< string > commands, out List< string > output, bool verbose=true)
EDITOR ONLY: Will use the included Unity Cloud Content Delivery CLI executable for this Unity Editor platform to execute commands. Refer to the Unity CLI documentation for commands. Note: This will automatically log in with the provided API key in the plugin settings, so there's no need to provide an 'auth login' command. This will also automatically append the current environment id to each command, so there's no need to append -environment [id].

Properties

- static UnityEvent< string > **AssetRetrievalStarted** = new UnityEvent<string>() [get]
Fires when GetAsset is called.
- static UnityEvent< string > **AssetLoaded** = new UnityEvent<string>() [get]
Fires when GetAsset is finished and asset is fully loaded, ready to instantiate.
- static UnityEvent< string > **AssetBundleDownloadStarted** = new UnityEvent<string>() [get]
Fires when GetAssetBundle is called.
- static UnityEvent< string > **AssetBundleDownloadFinished** = new UnityEvent<string>() [get]
Fires when GetAssetBundle is finished and asset bundle has completed downloading. Note that this does not mean the asset has been loaded. Subscribe to AssetLoaded to know when the asset has been loaded.
- static UnityEvent< string > **SceneAssetRetrievalStarted** = new UnityEvent<string>() [get]
Fires when LoadSceneAssetBundle is called.
- static UnityEvent< string > **SceneAssetLoaded** = new UnityEvent<string>() [get]
Fires when LoadSceneAssetBundle is finished and scene is fully loaded.
- static bool **AutoInitialize** = true [get, set]
Set to false to disable the auto initialization (retrieval of AssetBundleManifest and caching dependencies). Note: You must set this to false immediately when the app starts to avoid the first initialization attempt. Best practice would be to use a gameobject that calls awake in the initial scene in the game.
- static bool **IsPackagingAssets** [get]
Returns true if the Asset Bundle Manager is currently processing and packaging assets on an AssetBundles -> Build command.

6.2.1 Detailed Description

Static class that manages asset bundle retrieval and packaging.

6.2.2 Member Function Documentation

6.2.2.1 DoesBucketExist()

```
static async Task< bool > SplenSoft.AssetBundles.AssetBundleManager.DoesBucketExist (
    string bucketId ) [static]
```

Uses client-facing Unity Cloud Content Delivery API

to determine if a bucket exists. Requires an internet connection.

Parameters

<i>bucketId</i>	Bucket ID, can be pulled from AssetBundleManagerSettings
-----------------	--

Returns

True if bucket exists on the Unity CDN

6.2.2.2 GetAsset< T >()

```
static async Task< T > SplenSoft.AssetBundles.AssetBundleManager.GetAsset< T > (
    string name,
    IProgress< AssetRetrievalProgress > progress = null,
    Action< T > onSuccess = null,
    Action< AssetRetrievalResult > onFailure = null,
    bool waitForInitialize = true ) [static]
```

Downloads an asset bundle from the Unity Content Delivery cloud and unpacks the asset.

Parameters

<i>name</i>	The name of the AssetBundle
<i>progress</i>	Trackable progress reporter
<i>waitForInitialize</i>	The task will wait for the Asset Bundle Manager to initialize (retrieve its manifest for dependencies) before requesting the asset. Strongly recommended to leave this as default (true) unless you have a very special case (like retrieving the manifest manually)
<i>onSuccess</i>	Action which is invoked on successful retrieval
<i>onFailure</i>	Action which is invoked on failed retrieval

Returns

A Task object with a result of type T

Type Constraints

T : UnityEngine.Object

6.2.2.3 GetAssetBundle()

```
static async Task< AssetBundle > SplenSoft.AssetBundles.AssetBundleManager.GetAssetBundle (
    string name,
    IProgress< AssetRetrievalProgress > progress = null,
    Action< AssetBundle > onSuccess = null,
    Action< AssetRetrievalResult > onFailure = null,
    bool waitForInitialize = true ) [static]
```

Downloads an asset bundle from the Unity Content Delivery cloud.

Parameters

<i>name</i>	The name of the AssetBundle
<i>progress</i>	Trackable progress reporter
<i>waitForInitialize</i>	The task will wait for the Asset Bundle Manager to initialize (retrieve its manifest for dependencies) before requesting the asset. Strongly recommended to leave this as default (true) unless you have a very special case (like retrieving the manifest manually)
<i>onSuccess</i>	Action which is invoked on successful retrieval
<i>onFailure</i>	Action which is invoked on failed retrieval

Returns

A Task object with an AssetBundle result

6.2.2.4 GetAssetBundleNames() [1/2]

```
static async Task< string[]> SplenSoft.AssetBundles.AssetBundleManager.GetAssetBundleNames (
    string regexPattern ) [static]
```

If runtime, requires plugin to be initialized and Initialized must be true. Scans manifest for all asset bundle names. If running in the editor with Use Editor Assets checked in the settings, it will scan the project's AssetDatabase

Parameters

<i>regexPattern</i>	A regex to match against the asset bundle names
---------------------	---

Returns

All asset bundle names that matched the regex

6.2.2.5 GetAssetBundleNames() [2/2]

```
static async Task< string[]> SplenSoft.AssetBundles.AssetBundleManager.GetAssetBundleNames (
    Type type ) [static]
```

If runtime, requires plugin to be initialized and `Initialized` must be true. Scans manifest for all asset bundle names. If running in the editor with Use Editor Assets checked in the settings, it will scan the project's AssetDatabase

Parameters

<i>type</i>	Translates a type to a string that follows the standard asset naming format of EZ-CDN: typename_guid
-------------	--

Returns

All asset bundle names that matched `typename_guid`

6.2.2.6 GetDependencies()

```
static async Task SplenSoft.AssetBundles.AssetBundleManager.GetDependencies (
    string assetBundleName,
    IProgress< AssetRetrievalProgress > progress = null ) [static]
```

Downloads all of an AssetBundle's dependencies and keeps them cached. Good for loading ahead to make downloading a later asset bundle quicker.

Parameters

<i>assetBundleName</i>	The name of the AssetBundle
<i>progress</i>	Trackable progress reporter

Returns

A Task object

6.2.2.7 GetManifest()

```
static async Task< AssetBundleManifest > SplenSoft.AssetBundles.AssetBundleManager.GetManifest
( ) [static]
```

Downloads and unpacks the AssetBundleManifest for this platform from the CDN.

Returns

The AssetBundleManifest for this platform

6.2.2.8 LoadSceneAssetBundle()

```
static async Task SplenSoft.AssetBundles.AssetBundleManager.LoadSceneAssetBundle (
    string name,
    IProgress< AssetRetrievalProgress > progress = null,
    Action onSuccess = null,
    Action< AssetRetrievalResult > onFailure = null,
    bool waitForInitialize = true ) [static]
```

Downloads a scene asset bundle from the Unity Content Delivery cloud and immediately loads it.

Parameters

<i>name</i>	The name of the AssetBundle
<i>progress</i>	Trackable progress reporter
<i>waitForInitialize</i>	The task will wait for the Asset Bundle Manager to initialize (retrieve its manifest for dependencies) before requesting the asset. Strongly recommended to leave this as default (true) unless you have a very special case (like retrieving the manifest manually)
<i>onSuccess</i>	Action which is invoked on successful retrieval
<i>onFailure</i>	Action which is invoked on failed retrieval

Returns

A Task object

6.2.2.9 RunCliCommands()

```
static void SplenSoft.AssetBundles.AssetBundleManager.RunCliCommands (
    List< string > commands,
    out List< string > output,
    bool verbose = true ) [static]
```

EDITOR ONLY: Will use the included Unity Cloud Content Delivery CLI executable for this Unity Editor platform to execute commands. Refer to the Unity CLI documentation

for commands. Note: This will automatically log in with the provided API key in the plugin settings, so there's no need to provide an 'auth login' command. This will also automatically append the current environment id to each command, so there's no need to append `--environment [id]`.

Parameters

<i>commands</i>	
<i>output</i>	Output from cmd.exe or /bin/bash after running the supplied commands
<i>verbose</i>	Will append '--verbose' after every command. Strongly recommended to leave as default (true)

6.2.2.10 SetSelfInitializerTimeout()

```
static void SplenSoft.AssetBundles.AssetBundleManager.SetSelfInitializerTimeout (
    float timeInSeconds ) [static]
```

Sets a time, in seconds, that the Asset Bundle Manager will attempt to auto-initialize.

Parameters

<i>timeInSeconds</i>	Should be a value ≥ 1
----------------------	----------------------------

6.2.2.11 TryGetAssetBundleData()

```
static bool SplenSoft.AssetBundles.AssetBundleManager.TryGetAssetBundleData (
```

```
string assetBundleName,
out AssetBundleData data ) [static]
```

Attempts to return AssetBundleData for an asset bundle. The AssetBundleManager must be initialized for the manifest data to be available.

Returns

True if the asset bundle data exists in the library

6.2.2.12 TryGetAssetBundleName()

```
static bool SplenSoft.AssetBundles.AssetBundleManager.TryGetAssetBundleName (
    UnityEngine.Object obj,
    out string assetBundleName ) [static]
```

Attempts to retrieve or generate asset bundle name. Will fail if asset type is not defined in EZCDN settings.

Returns

True if asset bundle name was found or generated

6.2.2.13 TryGetAssetManifestRetrievalResult()

```
static bool SplenSoft.AssetBundles.AssetBundleManager.TryGetAssetManifestRetrievalResult (
    out AssetRetrievalResult result ) [static]
```

Gets status of last download attempt of the project asset bundle manifest for this platform. Note: If the plugin is set to use editor assets and this is running in an editor, it will return 200 with UnityWebRequest.Result.Success.

Returns

True if at least one attempt was made to retrieve the manifest prior to this call

6.2.2.14 TryGetAssetRetrievalResult()

```
static bool SplenSoft.AssetBundles.AssetBundleManager.TryGetAssetRetrievalResult (
    string assetBundleName,
    out AssetRetrievalResult result ) [static]
```

Gets status of last download attempt for an asset. Note: If an asset was pulled from the cache, it will return 200 with UnityWebRequest.Result.Success. To get the result of the initialization attempt (downloading the manifest), use TryGetAssetManifestRetrievalResult

Returns

True if the asset existed in the manifest (and the plugin was properly initialized), and at least one attempt was made to retrieve the asset prior to this call

The documentation for this class was generated from the following files:

- C:/Development Projects/for-the-birds/Packages/com.splensoft.ezcdn/Runtime/AssetBundleManager.cs
- C:/Development Projects/for-the-birds/Packages/com.splensoft.ezcdn/Runtime/AssetBundleManager_↔ Editor.cs

6.3 SplenSoft.AssetBundles.AssetBundleManagerSettings Class Reference

Controls the settings for EZ-CDN.

Public Member Functions

- string **GetApiKey** ()
Sensitive info - stored in PlayerPrefs in the editor and ONLY used in the editor - never sent to a runtime build.
- string **GetAssetBundleURL** ()
Returns
A client-side API url for Unity CCD to retrieve assets

Static Public Member Functions

- static [AssetBundleManagerSettings Get](#) ()
Loads the existing settings from the Resources folder or creates new settings if the file does not exist.
- static void **EnsureDirectoriesExist** ()
Ensures proper folders exist in /Assets/ to store settings files.

Properties

- LogLevel **LogLevel** [get, set]
What level of debug logging to output to the console.
- int **Version** [get, set]
Uses a one-number Major semantic version. Breaking changes should increase the version. Games that were published with a lower major version will NOT download updates from any other version. A good rule of thumb is this: If your major version changes in your game, this should probably change to, as breaking code changes can affect asset bundles.
- bool **KeepLocalCopy** [get, set]
Includes a local copy of the latest assetbundles on Build. Will automatically trigger an assetbundle build before an app build and copy them into Assets/StreamingAssets/AssetBundles. If this is true, the app will use the local copy when there is no internet connection or when the cloud copy of the assetbundle is not available. (Cached assets are still used first)
- int **MaxConcurrentDownloads = 3** [get, set]
Unity CCD has a tendency to fail miserably when there is a lot of concurrent downloads from a single IP. See forum posts here: It is recommended to leave this at default (3), but advanced users can mess around with it.

6.3.1 Detailed Description

Controls the settings for EZ-CDN.

6.3.2 Member Function Documentation

6.3.2.1 Get()

```
static AssetBundleManagerSettings SplenSoft.AssetBundles.AssetBundleManagerSettings.Get ( )  
[static]
```

Loads the existing settings from the Resources folder or creates new settings if the file does not exist.

Returns

The documentation for this class was generated from the following file:

- C:/Development Projects/for-the-birds/Packages/com.splensoft.ezcdn/Runtime/AssetBundleManager↔
Settings.cs

6.4 SplenSoft.AssetBundles.AssetBundleReferenceAttribute Class Reference

Apply this attribute to a SerializeField string field to reference an asset bundle name with the mask of a normal Unity property drawer of the supplied type.

6.4.1 Detailed Description

Apply this attribute to a SerializeField string field to reference an asset bundle name with the mask of a normal Unity property drawer of the supplied type.

The documentation for this class was generated from the following file:

- C:/Development Projects/for-the-birds/Packages/com.splensoft.ezcdn/Runtime/AssetBundleReference↔
Attribute.cs

6.5 SplenSoft.AssetBundles.AssetRequester Class Reference

MonoBehaviour component that can request an asset from the CDN. Accepts any UnityEngine.Object as an asset. Cannot instantiate or further manipulate the asset without the help of code. For codeless or more specific usages, try SceneRequester

Additional Inherited Members

Public Member Functions inherited from

[SplenSoft.AssetBundles.Requester](#)< [UnityEngine.Object](#) >

- bool [TryGetAssetRetrievalResult](#) (out [AssetRetrievalResult](#) assetRetrievalResult)
Gets status of last download attempt for an asset. Note: If an asset was pulled from the cache, it will return response code 200 with [UnityWebRequest.Result.Success](#).
- async Task< [AssetBundle](#) > [GetAssetBundle](#) (Progress< [AssetRetrievalProgress](#) > progress=null)
Retrieves this components Asset from the CDN.
- async Task< T > [GetAsset](#) (Progress< [AssetRetrievalProgress](#) > progress=null, Action< T > onSuccess=null, Action< [AssetRetrievalResult](#) > onFailure=null, bool waitForManagerInit=true)
Retrieves an asset from the content delivery network.

Properties inherited from [SplenSoft.AssetBundles.Requester](#)< [UnityEngine.Object](#) >

- UnityEvent< float > **OnProgressUpdated** [get]
Fires when download / load progress changes. Includes a float between 0 and 1 representing progress.
- UnityEvent< long > **OnRetrievalFailed** [get]
Fires when an asset retrieval attempt did not work. Includes a long integer http status code
- UnityEvent **OnRetrievalSuccess** [get]
Fires when an asset retrieval attempt succeeded.

6.5.1 Detailed Description

MonoBehaviour component that can request an asset from the CDN. Accepts any [UnityEngine.Object](#) as an asset. Cannot instantiate or further manipulate the asset without the help of code. For codeless or more specific usages, try [SceneRequester](#)

The documentation for this class was generated from the following file:

- C:/Development Projects/for-the-birds/Packages/com.splensoft.ezcdn/Runtime/AssetRequester.cs

6.6 SplenSoft.AssetBundles.AssetRetrievalProgress Class Reference

A trackable progress object that can be passed to [AssetBundleManager.GetAsset](#) and [AssetBundleManager.GetAsyncAssetBundle](#) through a [Progress<T>](#) object.

Properties

- [AssetRetrievalStatus](#) **Status** [get]
Gets the status of the current retrieval operation.
- bool **IsDone** [get]
Returns true if Status == [AssetRetrievalStatus.Done](#)
- float **Progress** [get]
Tracks the progress of the retrieval operation between 0 (just started) to 1 (done)

6.6.1 Detailed Description

A trackable progress object that can be passed to `AssetBundleManager.GetAsset` and `AssetBundleManager.GetAssetBundle` through a `Progress<T>` object.

The documentation for this class was generated from the following file:

- `C:/Development Projects/for-the-birds/Packages/com.splensoft.ezcdn/Runtime/AssetRetrievalProgress.cs`

6.7 SplenSoft.AssetBundles.AssetRetrievalResult Class Reference

Stores the results of the last attempt to download an asset bundle from the CDN.

6.7.1 Detailed Description

Stores the results of the last attempt to download an asset bundle from the CDN.

The documentation for this class was generated from the following file:

- `C:/Development Projects/for-the-birds/Packages/com.splensoft.ezcdn/Runtime/AssetRetrievalResult.cs`

6.8 SplenSoft.AssetBundles.AutoInstantiator Class Reference

Automatically downloads and instantiates prefabs on app start.

6.8.1 Detailed Description

Automatically downloads and instantiates prefabs on app start.

The documentation for this class was generated from the following file:

- `C:/Development Projects/for-the-birds/Packages/com.splensoft.ezcdn/Runtime/AutoInstantiator.cs`

6.9 SplenSoft.AssetBundles.IPreprocessAssetBundle Interface Reference

Used to add custom processing to an asset before it is packaged by EZ-CDN.

6.9.1 Detailed Description

Used to add custom processing to an asset before it is packaged by EZ-CDN.

The documentation for this interface was generated from the following file:

- C:/Development Projects/for-the-birds/Packages/com.splensoft.ezcdn/Runtime/IPreprocessAssetBundle.cs

6.10 SplenSoft.AssetBundles.ManagedAssetAttribute Class Reference

Add this attribute to a custom class to include it in the AssetBundleManager operations. Useful for custom Unity↔ Engine.ScriptableObjects. You can also add the class to the list of managed types in the [SplenSoft](#) -> Asset Bundles -> Settings.

6.10.1 Detailed Description

Add this attribute to a custom class to include it in the AssetBundleManager operations. Useful for custom Unity↔ Engine.ScriptableObjects. You can also add the class to the list of managed types in the [SplenSoft](#) -> Asset Bundles -> Settings.

The documentation for this class was generated from the following file:

- C:/Development Projects/for-the-birds/Packages/com.splensoft.ezcdn/Runtime/ManagedAssetAttribute.cs

6.11 SplenSoft.AssetBundles.Requester< T > Class Template Reference

Base class for asset bundle requesters of various objects or scenes.

Public Member Functions

- bool [TryGetAssetRetrievalResult](#) (out [AssetRetrievalResult](#) assetRetrievalResult)
Gets status of last download attempt for an asset. Note: If an asset was pulled from the cache, it will return response code 200 with UnityWebRequest.Result.Success.
- async Task< AssetBundle > [GetAssetBundle](#) (Progress< [AssetRetrievalProgress](#) > progress=null)
Retrieves this components Asset from the CDN.
- async Task< T > [GetAsset](#) (Progress< [AssetRetrievalProgress](#) > progress=null, Action< T > on↔ Success=null, Action< [AssetRetrievalResult](#) > onFailure=null, bool waitForManagerInit=true)
Retrieves an asset from the content delivery network.

Properties

- UnityEvent< float > **OnProgressUpdated** = new() [get]
Fires when download / load progress changes. Includes a float between 0 and 1 representing progress.
- UnityEvent< long > **OnRetrievalFailed** = new() [get]
Fires when an asset retrieval attempt did not work. Includes a long integer http status code
- UnityEvent **OnRetrievalSuccess** = new() [get]
Fires when an asset retrieval attempt succeeded.

6.11.1 Detailed Description

Base class for asset bundle requesters of various objects or scenes.

Type Constraints

T : *UnityEngine.Object*

6.11.2 Member Function Documentation

6.11.2.1 GetAsset()

```
async Task< T > SplenSoft.AssetBundles.Requester< T >.GetAsset (
    Progress< AssetRetrievalProgress > progress = null,
    Action< T > onSuccess = null,
    Action< AssetRetrievalResult > onFailure = null,
    bool waitForManagerInit = true )
```

Retrieves an asset from the content delivery network.

Template Parameters

<i>T</i>	Any UnityEngine.Object derivative type, such as GameObject or Sprite
----------	--

6.11.2.2 GetAssetBundle()

```
async Task< AssetBundle > SplenSoft.AssetBundles.Requester< T >.GetAssetBundle (
    Progress< AssetRetrievalProgress > progress = null )
```

Retrieves this components Asset from the CDN.

Parameters

<i>progress</i>	Optional custom progress tracker
-----------------	----------------------------------

Returns

A Task object

6.11.2.3 TryGetAssetRetrievalResult()

```
bool SplenSoft.AssetBundles.Requester< T >.TryGetAssetRetrievalResult (
    out AssetRetrievalResult assetRetrievalResult )
```

Gets status of last download attempt for an asset. Note: If an asset was pulled from the cache, it will return response code 200 with UnityWebRequest.Result.Success.

Parameters

<code>assetRetrievalResult</code>	
-----------------------------------	--

Returns

True if at least one attempt was made to retrieve the manifest prior to this call

The documentation for this class was generated from the following file:

- C:/Development Projects/for-the-birds/Packages/com.splensoft.ezcdn/Runtime/Requester.cs

6.12 SplenSoft.AssetBundles.SceneRequester Class Reference

MonoBehaviour component that can request a scene asset from the CDN. Accepts any UnityEditor.SceneAsset as an asset.

Public Member Functions

- void **DownloadAndLoadScene** ()
UnityEvent listener for the inspector.
- async void **DownloadAndLoadSceneAsync** (Progress< [AssetRetrievalProgress](#) > progress=null, Action on↵ Success=null, Action< [AssetRetrievalResult](#) > onFailure=null, bool waitForManagerInit=true)
Downloads and loads the scene attached to this component.

Public Member Functions inherited from [SplenSoft.AssetBundles.Requester](#)< [UnityEngine.Object](#) >

- bool [TryGetAssetRetrievalResult](#) (out [AssetRetrievalResult](#) assetRetrievalResult)
Gets status of last download attempt for an asset. Note: If an asset was pulled from the cache, it will return response code 200 with UnityWebRequest.Result.Success.
- async Task< AssetBundle > [GetAssetBundle](#) (Progress< [AssetRetrievalProgress](#) > progress=null)
Retrieves this components Asset from the CDN.
- async Task< T > [GetAsset](#) (Progress< [AssetRetrievalProgress](#) > progress=null, Action< T > on↵ Success=null, Action< [AssetRetrievalResult](#) > onFailure=null, bool waitForManagerInit=true)
Retrieves an asset from the content delivery network.

Additional Inherited Members

Properties inherited from [SplenSoft.AssetBundles.Requester](#)< [UnityEngine.Object](#) >

- UnityEvent< float > **OnProgressUpdated** [get]
Fires when download / load progress changes. Includes a float between 0 and 1 representing progress.
- UnityEvent< long > **OnRetrievalFailed** [get]
Fires when an asset retrieval attempt did not work. Includes a long integer http status code
- UnityEvent **OnRetrievalSuccess** [get]
Fires when an asset retrieval attempt succeeded.

6.12.1 Detailed Description

MonoBehaviour component that can request a scene asset from the CDN. Accepts any UnityEditor.SceneAsset as an asset.

6.12.2 Member Function Documentation

6.12.2.1 DownloadAndLoadSceneAsync()

```
async void SplenSoft.AssetBundles.SceneRequester.DownloadAndLoadSceneAsync (
    Progress< AssetRetrievalProgress > progress = null,
    Action onSuccess = null,
    Action< AssetRetrievalResult > onFailure = null,
    bool waitForManagerInit = true )
```

Downloads and loads the scene attached to this component.

Parameters

<i>progress</i>	An optional Progress object to track download/load progress for a loading screen / bar
-----------------	--

The documentation for this class was generated from the following file:

- C:/Development Projects/for-the-birds/Packages/com.splensoft.ezcdn/Runtime/SceneRequester.cs

Index

DoesBucketExist
 SplenSoft.AssetBundles.AssetBundleManager, 14

DownloadAndLoadSceneAsync
 SplenSoft.AssetBundles.SceneRequester, 26

Easy Content Delivery Network for Unity, 1

Flush
 SplenSoft.AssetBundles.AssetBundleData, 12

Get
 SplenSoft.AssetBundles.AssetBundleManagerSettings, 20

GetAsset
 SplenSoft.AssetBundles.Requester< T >, 24

GetAsset< T >
 SplenSoft.AssetBundles.AssetBundleManager, 14

GetAssetBundle
 SplenSoft.AssetBundles.AssetBundleManager, 15
 SplenSoft.AssetBundles.Requester< T >, 24

GetAssetBundleNames
 SplenSoft.AssetBundles.AssetBundleManager, 15

GetDependencies
 SplenSoft.AssetBundles.AssetBundleManager, 16

GetManifest
 SplenSoft.AssetBundles.AssetBundleManager, 16

LoadSceneAssetBundle
 SplenSoft.AssetBundles.AssetBundleManager, 16

RunCliCommands
 SplenSoft.AssetBundles.AssetBundleManager, 17

SetSelfInitializerTimeout
 SplenSoft.AssetBundles.AssetBundleManager, 17

SplenSoft, 9

SplenSoft.AssetBundles, 9

SplenSoft.AssetBundles.AssetBundleData, 11
 Flush, 12

SplenSoft.AssetBundles.AssetBundleManager, 12
 DoesBucketExist, 14
 GetAsset< T >, 14
 GetAssetBundle, 15
 GetAssetBundleNames, 15
 GetDependencies, 16
 GetManifest, 16
 LoadSceneAssetBundle, 16
 RunCliCommands, 17
 SetSelfInitializerTimeout, 17
 TryGetAssetBundleData, 17
 TryGetAssetBundleName, 18
 TryGetAssetManifestRetrievalResult, 18
 TryGetAssetRetrievalResult, 18

SplenSoft.AssetBundles.AssetBundleManagerSettings, 19
 Get, 20

SplenSoft.AssetBundles.AssetBundleReferenceAttribute, 20

SplenSoft.AssetBundles.AssetRequester, 20

SplenSoft.AssetBundles.AssetRetrievalProgress, 21

SplenSoft.AssetBundles.AssetRetrievalResult, 22

SplenSoft.AssetBundles.AutoInstantiator, 22

SplenSoft.AssetBundles.IPreprocessAssetBundle, 22

SplenSoft.AssetBundles.ManagedAssetAttribute, 23

SplenSoft.AssetBundles.Requester< T >, 23
 GetAsset, 24
 GetAssetBundle, 24
 TryGetAssetRetrievalResult, 24

SplenSoft.AssetBundles.SceneRequester, 25
 DownloadAndLoadSceneAsync, 26

TryGetAssetBundleData
 SplenSoft.AssetBundles.AssetBundleManager, 17

TryGetAssetBundleName
 SplenSoft.AssetBundles.AssetBundleManager, 18

TryGetAssetManifestRetrievalResult
 SplenSoft.AssetBundles.AssetBundleManager, 18

TryGetAssetRetrievalResult
 SplenSoft.AssetBundles.AssetBundleManager, 18
 SplenSoft.AssetBundles.Requester< T >, 24