Better Resources

Generated by Doxygen 1.9.1

1 Better Resources

Better Resources is a library that offers a better API to interact with Unity3D resources. Most notably, it extends the existing `Resources` with the capability to query all resource paths in your project and imported packages, both at **design-and-run-time**!

1.1 Getting Started

Better Resources exposes the complete UnityEngine `Resources` API, which can be used straight out of the box. However, in order to query all resources in a project (including packages) a ResourcesCache is needed.

A ResourcesCache can be generated through the different overloads of `BetterResourcesEditor`. This file can be generated **in the Unity3D editor** from the `Tools/HamerSoft/BetterResources/Generate`

`Cache` menu in the toolbar or through the available API. BetterResources also comes with a pre-made pre-build hook to generate a ResourcesCache before a build is compiled. See `enabling the built-in pre-build hook` for more information.

Once a cache is generated and saved in a Resources root folder like: `<Project>/Assets/Resources/ResourcesCache` it can be used in a build to initialize BetterResources to enable queries.

1.1.1 Documentation

For more detailed docs please review the `GH-Pages`.

1.2 Initialization

BetterResources requires an initialization step (on the main thread) in order to enable querying the available resources at run-time. This step is needed to load a ResourcesCache, which was made in the editor, into memory.

BetterResources exposes 3 flavors of initialization for a fitting coding style:

Initialization	Description
BetterResources.Initialize	Blocking initialization on the main thread.
BetterResources.InitializeAsync	Non-blocking initialization using async/await.
BetterResources.InitializeRoutine	Non-blocking initialization using a Unity3D coroutine.

Once the initialization process has finished the ResourcesCache is loaded into memory and queries are enabled. *Note: BetterResources exposes a (static)* `Initialized` *event with a boolean flag to check if the initialization was successful.*

1.3 Queries

The core piece of functionality BetterResources exposes over the default `Resources` API Unity3D provides is: **Being able to query available resources at design-and-run-time**. BetterResources exposes a nice `QueryBuilder` object to create fine-grained lookups for resources.

For example: *Let's say we want to find all resources in your custom made package in some root* `Resources` *folder with a* `Camera` *component, but without an* `AudioListener` *component (because the rule of two doesn't work well here :D).*

```
using var query = BetterResources.Query()
          .ByPackage("com.my-org.package")
          .AtRoot()
          .WithSomeComponents(typeof(Camera))
          .WithoutAllComponents(typeof(AudioListener));

var results = query.GetResults();
// do something with results
```

∗*The QueryBuilder supports a lot more filters so be sure to check them out, and if you have a suggestion for a useful filter that is missing please create a ticket, or better yet, create a pull-request.*

The QueryBuilder implements an IDisposable interface so it's best to use it in a `using` *statement for resource optimization.*

1.4 Misc

1.4.1 Root & Nested Resources folders

In contrary to common believe, Unity3D does support nested resources folders. So, for example: A path like: `<Project>/Assets/Resources/RandomFolder/Resources/MyAsset.asset` is supported by Unity3D. Another fun fact is that `MyAsset.asset` can be loaded through the built-in Resources Api using these two paths:

```
// Like this
Resources.Load("MyAsset");
// or
Resources.Load("RandomFolder/Resources/MyAsset");
```

This is because each occurence of a directory called `Resources` is treated as a root to load from.

1.4.2 Non-Root Resources

The Resources folder also doesn't have to be in the root of the Unity3D project or package. It's perfectly viable to have a `Resources` folder located at `<Project>/Assets/SomeFolder/AnotherFolder/Resources`. This Resources folder, will be treated as a root folder like no other.

1.4.3 Enabling the built-in pre-build hook

Better Resources has a built-in pre-build hook that can be enabled to generate a ResourceCache just before a build is made. The hook is executed when a build is triggered through the Unity3D `BuildSettings` The pre-build hook can be enabled in different ways:

1.4.3.1 Custom Define in Code Add the custom `BETTERRESOURCES_PRE_BUILD` define at the top of some file.

```
#define BETTERRESOURCES_PRE_BUILD
public class MyRandomClass
{

}
```

You could also wrap this define in line a `UNITY_ANDROID` define if you just want to use the hook before making an Android build for example.

1.4.3.2 Custom Define in PlayerSettings Add the custom `BETTERRESOURCES_PRE_BUILD` define to the `PlayerSettings` in the `other` tab (per platform) like so:

1.4.3.3 Custom Define through Unity3D API A custom define can also be set through the Unity3D `PlayerDefineSymbols` API. Just remember to do it for all platforms needed, just like the other alternatives.

1.4.3.4 Custom implementation A totally custom implementation is always possible. Inspiration can be taken from `PreBuildHook`. The `BetterResourcesEditor` exposes a number of overloads to generate a cache. These could be called through a custom pre-build hook, CloudBuild `Pre-Export` method or a custom CI implementation like GitHub Actions. _Just remember that most, if not all, of the pre-and-post build handlers in Unity run synchronously!_

1.5 Editor Integration

Better Resources also comes with a AssetPostProcessor. This PostProcessor will detect if there's any changes being made to files in `Resources` directories. Once a new asset is added, deleted or moved to the `Resources` folder(s) it will kick in and generate a new cache and initialize BetterResources for use. It's main use is for better integration with custom Editor plugins that might use BetterResources to find resources. By using the AssetPost↩ Processor a new cache is generated and initialization is done automatically.

The AssetPostProcessor can be enabled by adding the scripting define: `BETTERRESOURCES_AUTO_↩ GENERATE`. ∗*The process for adding this custom define is identical to adding it for the pre-build hook.*

1.6 Custom Define Summary

A summary of the custom defines for Better Resources is the following:

Define	Description
BETTERRESOURCES_PRE_BUILD	Enables the pre-made pre-build hook to generate the cache once a build is made.
BETTERRESOURCES_AUTO_GENERATE	Enables Editor integration, this is useful for writing Editor Plugins that need access to BetterResources.
BETTERRESOURCES_LOG	Enables logging. (Logging is a bit incomplete)

1.7 Acknowledgements

Better Resources is a tribute to another amazing library called `BetterStreamingAssets`, hence the name :)

```
"Better Streaming Assets is a plugin that lets you access Streaming Assets
directly in an uniform and thread-safe way, with tiny overhead.  Mostly
beneficial for Android projects, where the alternatives are to use archaic
and hugely inefficient WWW or embed data in Asset Bundles.  API is based on
Syste.IO.File and System.IO.Directory classes."
```

2 <a href="https://github.com/HamerSoft/better-resources/compare/v0.1.0...v0.2.0">0.2.0</a> (2023-12-02)

2.0.1 Features

- add Load byList and PostProcessor (`32135b2`)

2.1 <a href="https://github.com/HamerSoft/better-resources/compare/73f7c61b992782aec1386ea0659f9b2f5b1a0425...v0.1.0">0.1.↩ 0</a> (2023-11-26)

2.1.0.1 Features

- restructure dir to support dll compilation (`73f7c61`)

3 LICENSE

4 Namespace Index

4.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

5 Class Index

5.1 Class List

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

6 Namespace Documentation

6.1 HamerSoft Namespace Reference

6.2 HamerSoft.BetterResources Namespace Reference

Classes

- class BetterResources

  A Better API to load resources from the Unity3D Resources folders.
- class QueryBuilder

  A Query Builder to Search the Unity3D Resources

6.3 HamerSoft.BetterResources.Awaiters Namespace Reference

Classes

- struct ListRequestAwaiter

  Awaiter for UnityEditor.PackageManager.Requests.ListRequest
- class AwaiterExtensions

  Extensions for custom awaiters to use with async keyword
- struct ResourceRequestAwaiter

  Awaiter for UnityEngine.ResourceRequest

6.4 HamerSoft.BetterResources.Dto Namespace Reference

6.5 HamerSoft.BetterResources.Editor Namespace Reference

Classes

- class AwaiterExtensionsEditor

  Extensions for custom awaiters to use with async keyword for Editor namespace specific logic
- class BetterResourcesEditor

  This class exposes a couple of variants to generate the a ResourceCache.

6.6 HamerSoft.BetterResources.Extensions Namespace Reference

Classes

- class TaskExtensions

  Extensions to use tasks as coroutines

6.7 HamerSoft.BetterResources.Samples Namespace Reference

6.8 HamerSoft.BetterResources.Samples.CacheGeneration Namespace Reference

6.9 HamerSoft.BetterResources.Samples.QueryBuilder Namespace Reference

6.10 HamerSoft.BetterResources.Tests Namespace Reference

6.11 HamerSoft.BetterResources.Tests.Initialization Namespace Reference

7 Class Documentation

7.1 HamerSoft.BetterResources.Awaiters.AwaiterExtensions Class Reference

Extensions for custom awaiters to use with async keyword

Static Public Member Functions

- static ResourceRequestAwaiter GetAwaiter (this ResourceRequest asyncOp)

  Get an awaiter for a ResourceRequest

7.1.1 Detailed Description

Extensions for custom awaiters to use with async keyword

7.1.2 Member Function Documentation

7.1.2.1 GetAwaiter()

```
static ResourceRequestAwaiter HamerSoft.BetterResources.Awaiters.Awaiter↩
Extensions.GetAwaiter (
            this ResourceRequest asyncOp )  [inline], [static]
```

Get an awaiter for a ResourceRequest

Parameters

asyncOp	the resource request

Returns

Awaitable object

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

- Runtime/Awaiters/AwaiterExtensions.cs

7.2 HamerSoft.BetterResources.Editor.AwaiterExtensionsEditor Class Reference

Extensions for custom awaiters to use with async keyword for Editor namespace specific logic

Static Public Member Functions

- static ListRequestAwaiter GetAwaiter (this ListRequest asyncOp)

  Get an awaiter for a ListRequest from the UnityEditor.PackageManager

7.2.1 Detailed Description

Extensions for custom awaiters to use with async keyword for Editor namespace specific logic

7.2.2 Member Function Documentation

7.2.2.1 GetAwaiter()

```
static ListRequestAwaiter HamerSoft.BetterResources.Editor.AwaiterExtensions↩
Editor.GetAwaiter (
            this ListRequest asyncOp )  [inline], [static]
```

Get an awaiter for a ListRequest from the UnityEditor.PackageManager

Parameters

asyncOp	The ListRequest

Returns

Awaitable object

Sometimes it takes a very long time for this to return!

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

- Editor/Awaiters/AwaiterExtensionsEditor.cs

7.3 HamerSoft.BetterResources.BetterResources Class Reference

A Better API to load resources from the Unity3D Resources folders.

Static Public Member Functions

- static void Initialize (string directory=null)

  Initialize BetterResources
  See also

  InitializeAsync

  InitializeRoutine
- static async Task InitializeAsync (string directory=null, CancellationToken token=default)

  Initialize BetterResources Async
  See also

  Initialize, InitializeRoutine
- static IEnumerator InitializeRoutine (string directory=null, CancellationToken token=default)

  Initialize BetterResources in a Coroutine
  See also

  Initialize, InitializeAsync
- static Object Load (string path, Type type=null)

  Loads the asset of the requested type stored at path in a Resources folder using a parameter type filter of type.
- static Object Load (ResourceAsset resourceAsset, Type type=null)

  Loads the asset of the requested type stored at path in a Resources folder using a parameter type filter of type.
- static Object[ ] Load (IEnumerable< ResourceAsset > resourceAssets, Type type=null)

  Loads the multiple assets of the requested type stored at path in a Resources folder using a parameter type filter of type.
- static T Load< T > (string path)

  Loads the asset of the requested type stored at path in a Resources folder using a generic parameter type filter of type T.
- static T Load< T > (ResourceAsset resourceAsset)

  Loads the asset of the requested type stored at path in a Resources folder using a generic parameter type filter of type T.
- static T[ ] Load< T > (IEnumerable< ResourceAsset > resourceAssets)

  Loads a collection of assets of the requested type stored at path in a Resources folder using a generic parameter type filter of type T.
- static async Task< T > LoadAsync< T > (string path)

  Asynchronously loads an asset stored at path in a Resources folder.
- static async Task< T > LoadAsync< T > (ResourceAsset resourceAsset)

  Asynchronously loads an asset stored at path in a Resources folder.
- static async Task< T[ ]> LoadAsync< T > (IEnumerable< ResourceAsset > resourceAssets)

  Asynchronously loads a collection of asset stored at path in a Resources folder.
- static async Task< Object > LoadAsync (string path, Type type=null)

  Asynchronously loads an asset stored at path in a Resources folder.
- static async Task< Object > LoadAsync (ResourceAsset resourceAsset, Type type=null)

  Asynchronously loads an asset stored at path in a Resources folder.
- static async Task< Object[ ]> LoadAsync (IEnumerable< ResourceAsset > resourceAssets, Type type=null)

  Asynchronously loads a collection of assets stored at path in a Resources folder.
- static T GetBuiltinResource< T > (string path)

Get a built-in Resource
- static Object GetBuiltinResource (Type type, string path)

  Get a built-in Resource
- static void UnloadAsset (Object asset)

  Unload an asset from memory
- static void UnloadUnusedAssets ()

  Unload all unused Assets
- static Object[ ] FindObjectsOfTypeAll (Type type)

  Returns a list of all objects of Type type.
- static T[ ] FindObjectsOfTypeAll< T > ()

  Returns a list of all objects of Type type.
- static Object InstanceIdToObject (int instanceID)

  Translates an instance ID to an object reference
- static void InstanceIdToObjectList (NativeArray< int > instanceIDs, List< Object > objects)

  Translates instance IDs to object references
- static QueryBuilder Query ()

  Create a new instance of a query builder to search the available resources

Properties

- static bool IsInitialized `[get]`

  A flag indicating if BetterResources is Initialized
- static bool IsValid `[get]`

  A flag indicating if BetterResources is successfully Initialized

Events

- static Action< bool > Initialized

  Event fired when BetterResources is Initialized!

7.3.1 Detailed Description

A Better API to load resources from the Unity3D Resources folders.

Make sure to

See also

Initialize, InitializeAsync, InitializeRoutine

| | in the Editor / Pre-Build before using the Query API.

7.3.2 Member Function Documentation

7.3.2.1 FindObjectsOfTypeAll()

```
static Object [ ] HamerSoft.BetterResources.BetterResources.↩
FindObjectsOfTypeAll (
            Type type )  [inline], [static]
```

Returns a list of all objects of Type type.

Parameters

type	type to search for.

Pro-Tip: use Query

Returns

Array of objects that match the type

7.3.2.2 FindObjectsOfTypeAll< T >() `static T [] HamerSoft.BetterResources.BetterResources.FindObjectsOfTypeAll T > ( ) [inline], [static]`

Returns a list of all objects of Type type.

Template Parameters

T	type to search for.

Pro-Tip: use Query

Returns

Array of objects that match the type

Type Constraints

T* : *Object

7.3.2.3 GetBuiltinResource()

```
static Object HamerSoft.BetterResources.BetterResources.Get←
BuiltinResource (
             Type type,
             string path )  [inline], [static]
```

Get a built-in Resource

Parameters

path	Path of the resource

Some resources require file extensions like: Sphere.fbx

Parameters

type	Type filter

Returns

Built-in resource of optional type

7.3.2.4 GetBuiltinResource< T >() `static T HamerSoft.BetterResources.BetterResources.GetBuiltinResource< T > ( string path ) [inline], [static]`

Get a built-in Resource

Parameters

path	Path of the resource

Some resources require file extensions like: Sphere.fbx

Template Parameters

T	Generic type filter

Returns

Built-in resource of Type T

Type Constraints

T* : *Object

7.3.2.5 Initialize() `static void HamerSoft.BetterResources.BetterResources.Initialize ( string directory = null ) [inline], [static]`

Initialize BetterResources

See also

InitializeAsync

InitializeRoutine

Parameters

directory	Optional directory (local to Resources) where the ResourcesCache exists, if null is used, the ResourceCache is loaded from root.

GenerateCache allows for an optional directory too, these must match!

7.3.2.6 InitializeAsync()

```
static async Task HamerSoft.BetterResources.BetterResources.InitializeAsync (
            string directory = null,
            CancellationToken token = default )  [inline], [static]
```

Initialize BetterResources Async

See also

Initialize, InitializeRoutine

Parameters

directory	Optional directory (local to Resources) where the ResourcesCache exists, if null is used, the ResourceCache is loaded from root.
token	Optional token to cancel the ongoing initialization process.

GenerateCache allows for an optional directory too, these must match!

7.3.2.7 InitializeRoutine()

```
static IEnumerator HamerSoft.BetterResources.BetterResources.InitializeRoutine (
            string directory = null,
            CancellationToken token = default )  [inline], [static]
```

Initialize BetterResources in a Coroutine

See also

Initialize, InitializeAsync

Parameters

directory	Optional directory (local to Resources) where the ResourcesCache exists, if null is used, the ResourceCache is loaded from root.
token	Optional token to cancel the ongoing initialization process. (The token os used since there is no nice way of cancelling the IEnumerator in this particular case)

GenerateCache allows for an optional directory too, these must match!

7.3.2.8 InstanceIdToObject()

```
static Object HamerSoft.BetterResources.BetterResources.InstanceIdToObject (
            int instanceID )  [static]
```

Translates an instance ID to an object reference

Parameters

instanceID	Instance ID of an Object.

Returns

Resolved reference or null if the instance ID didn't match anything.

7.3.2.9 InstanceIdToObjectList() `static void HamerSoft.BetterResources.BetterResources.Instance↩ IdToObjectList (`

```
            NativeArray< int > instanceIDs,
            List< Object > objects )  [static]
```

Translates instance IDs to object references

Parameters

instanceIDs	array of instance IDs
objects	List of Object

A List can be used as an argument since it keeps references

7.3.2.10 Load() [1/3] `static Object [] HamerSoft.BetterResources.BetterResources.Load (`

```
            IEnumerable< ResourceAsset > resourceAssets,
            Type type = null )  [inline], [static]
```

Loads the multiple assets of the requested type stored at path in a Resources folder using a parameter type filter of type.

Parameters

resourceAssets	ResourceAssets, found through Query
type	Optional type filter

Type can be of any in the ResourceAsset.Components

Returns

A collection of objects at the path of the ResourceAsset, by any type filter that matches

7.3.2.11 Load() [2/3] `static Object HamerSoft.BetterResources.BetterResources.Load (`

```
            ResourceAsset resourceAsset,
            Type type = null )  [inline], [static]
```

Loads the asset of the requested type stored at path in a Resources folder using a parameter type filter of type.

Parameters

resourceAsset	ResourceAsset, found through Query
type	Optional type filter

Type can be of any in the ResourceAsset.Components

Returns

An object at the path of the ResourceAsset, by any type filter that matches

7.3.2.12 Load() [3/3]

```
static Object HamerSoft.BetterResources.BetterResources.Load (
            string path,
            Type type = null )  [inline], [static]
```

Loads the asset of the requested type stored at path in a Resources folder using a parameter type filter of type.

Parameters

path	path local to Resources
type	optional type filter

Returns

An object at the requested path, of type, or null

7.3.2.13 Load< T >() [1/3]

```
static T [] HamerSoft.BetterResources.BetterResources.Load< T > (
            IEnumerable< ResourceAsset > resourceAssets )  [inline], [static]
```

Loads a collection of assets of the requested type stored at path in a Resources folder using a generic parameter type filter of type T.

Parameters

resourceAssets	ResourceAssets, found through Query

Template Parameters

T	Generic Type filter

Returns

An collection of objects of the requested generic parameter type

Type Constraints

T* : *Object

7.3.2.14 Load< T >() [2/3] static T HamerSoft.BetterResources.BetterResources.Load< T > (ResourceAsset *resourceAsset*) [inline], [static]

Loads the asset of the requested type stored at path in a Resources folder using a generic parameter type filter of type T.

Parameters

resourceAsset	ResourceAsset, found through Query

Template Parameters

T	Generic Type filter

Returns

An object of the requested generic parameter type

Type Constraints

T* : *Object

7.3.2.15 Load< T >() [3/3] static T HamerSoft.BetterResources.BetterResources.Load< T > (string *path*) [inline], [static]

Loads the asset of the requested type stored at path in a Resources folder using a generic parameter type filter of type T.

Parameters

path	path local to Resources

Template Parameters

T	Generic Type Filter

Returns

An object of the requested generic parameter type

Type Constraints

T* : *Object

7.3.2.16 LoadAsync() [1/3] static async Task<Object[ ]> HamerSoft.BetterResources.Better↩ Resources.LoadAsync (

```
            IEnumerable< ResourceAsset > resourceAssets,
            Type type = null ) [inline], [static]
```

Asynchronously loads a collection of assets stored at path in a Resources folder.

Parameters

resourceAssets	ResourceAssets, found through Query

When using the empty string (i.e., ""), the function will load the entire contents of the Resources folder.

Parameters

type	Type Filter

Returns

A collection of objects of the requested type parameter type

7.3.2.17 LoadAsync() [2/3] static async Task<Object> HamerSoft.BetterResources.BetterResources.↩ LoadAsync (

```
            ResourceAsset resourceAsset,
            Type type = null ) [inline], [static]
```

Asynchronously loads an asset stored at path in a Resources folder.

Parameters

resourceAsset	ResourceAsset, found through Query

When using the empty string (i.e., ""), the function will load the entire contents of the Resources folder.

Parameters

type	Type Filter

Returns

An object of the requested type parameter type

7.3.2.18 LoadAsync() [3/3] static async Task<Object> HamerSoft.BetterResources.BetterResources.↩ LoadAsync (

```
            string path,
            Type type = null )  [inline], [static]
```

Asynchronously loads an asset stored at path in a Resources folder.

Parameters

path	pathname of the target folder

When using the empty string (i.e., ""), the function will load the entire contents of the Resources folder.

Parameters

type	Type Filter

Returns

An object of the requested type parameter type

7.3.2.19 LoadAsync< T >() [1/3]

```
static async Task<T[]> HamerSoft.BetterResources.BetterResources.LoadAsync<
T > (
            IEnumerable< ResourceAsset > resourceAssets )  [inline], [static]
```

Asynchronously loads a collection of asset stored at path in a Resources folder.

Parameters

resourceAssets	ResourceAssets, found through Query

Template Parameters

T	Generic Type Filter

Type filter T can be of any Type in ResourceAssets.Components

Returns

A collection of objects of the requested generic parameter type

Type Constraints

T : Object

7.3.2.20 LoadAsync< T >() [2/3]

```
static async Task<T> HamerSoft.BetterResources.BetterResources.LoadAsync<
T > (
            ResourceAsset resourceAsset )  [inline], [static]
```

Asynchronously loads an asset stored at path in a Resources folder.

Parameters

resourceAsset	ResourceAsset, found through Query

Template Parameters

T	Generic Type Filter

Type filter T can be of any Type in ResourceAsset.Components

Returns

An object of the requested generic parameter type

Type Constraints

T* : *Object

7.3.2.21 LoadAsync< T >() [3/3]

```
static async Task<T> HamerSoft.BetterResources.BetterResources.LoadAsync<
T > (
            string path ) [inline], [static]
```

Asynchronously loads an asset stored at path in a Resources folder.

Parameters

path	pathname of the target folder

When using the empty string (i.e., ""), the function will load the entire contents of the Resources folder.

Template Parameters

T	Generic Type Filter

Returns

An object of the requested generic parameter type

Type Constraints

T* : *Object

7.3.2.22 Query() `static QueryBuilder HamerSoft.BetterResources.BetterResources.Query ( ) [inline], [static]`

Create a new instance of a query builder to search the available resources

QueryBuilder is disposable, so best to use it in a using statement

Returns

QueryBuilder instance

7.3.2.23 UnloadAsset() `static void HamerSoft.BetterResources.BetterResources.UnloadAsset ( Object asset ) [inline], [static]`

Unload an asset from memory

This will also destroy all existing references!

Parameters

asset	The object to destroy

7.3.2.24 UnloadUnusedAssets() `static void HamerSoft.BetterResources.BetterResources.Unload↩UnusedAssets ( ) [inline], [static]`

Unload all unused Assets

7.3.3 Property Documentation

7.3.3.1 IsInitialized `bool HamerSoft.BetterResources.BetterResources.IsInitialized [static], [get]`

A flag indicating if BetterResources is Initialized

7.3.3.2 IsValid `bool HamerSoft.BetterResources.BetterResources.IsValid [static], [get]`

A flag indicating if BetterResources is successfully Initialized

Initialization might be invalid when there is no ResourceCache found at the initialization directory or when the ResourceCache contains an invalid JSON.

7.3.4 Event Documentation

7.3.4.1 Initialized `Action<bool> HamerSoft.BetterResources.BetterResources.Initialized [static]`

Event fired when BetterResources is Initialized!

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

- Runtime/BetterResources.cs

7.4 HamerSoft.BetterResources.Editor.BetterResourcesEditor Class Reference

This class exposes a couple of variants to generate the a ResourceCache.

Static Public Member Functions

- static async void GenerateCacheAsync ()

  Generate the Cache in order to BetterResources.Query
- static async Task GenerateCacheAsync (CancellationToken token)

  Generate the Cache in order to BetterResources.Query
- static void GenerateCache ()

  Generate the Cache in order to BetterResources.Query

Events

- static Action CacheGenerated

  Event fired when a new cache has been generated

7.4.1 Detailed Description

This class exposes a couple of variants to generate the a ResourceCache.

GenerateCacheAsync is used by the HamerSoft/BetterResources/Generate Cache menu item in the editor

GenerateCache is used by the Pre-Build hook.

7.4.2 Member Function Documentation

7.4.2.1 GenerateCache() `static void HamerSoft.BetterResources.Editor.BetterResourcesEditor.GenerateCache ( ) [inline], [static]`

Generate the Cache in order to BetterResources.Query

You can use the built-in load functions without cache.

This function can be triggered through CI or in the UnityCloud build just like PreBuildHook.

7.4.2.2 GenerateCacheAsync() [1/2] `static async void HamerSoft.BetterResources.Editor.BetterResourcesEditor.GenerateCacheAsync ( ) [inline], [static]`

Generate the Cache in order to BetterResources.Query

You can use the built-in load functions without cache.

This function can be triggered through CI and a Menu in the Unity3D editor toolbar at HamerSoft/BetterResources/Generate Cache.

7.4.2.3 GenerateCacheAsync() [2/2] `static async Task HamerSoft.BetterResources.Editor.BetterResourcesEditor.GenerateCacheAsync ( CancellationToken token ) [inline], [static]`

Generate the Cache in order to BetterResources.Query

You can use the built-in load functions without cache.

This function can be triggered through CI.

Parameters

token	Optional token to cancel the cache generation process

7.4.3 Event Documentation

7.4.3.1 CacheGenerated `Action HamerSoft.BetterResources.Editor.BetterResourcesEditor.CacheGenerated [static]`

Event fired when a new cache has been generated

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

- Editor/BetterResourcesEditor.cs

7.5 HamerSoft.BetterResources.Awaiters.ListRequestAwaiter Struct Reference

Awaiter for UnityEditor.PackageManager.Requests.ListRequest

Inherits INotifyCompletion.

Public Member Functions

- ListRequestAwaiter (ListRequest asyncOperation)

  Initialized a new instance of ListRequestAwaiter
- void OnCompleted (Action continuation)
- PackageCollection GetResult ()

  Get the return value from the ListRequest

Properties

- bool IsCompleted `[get]`

  A flag indicating if the ListRequest is complete

7.5.1 Detailed Description

Awaiter for UnityEditor.PackageManager.Requests.ListRequest

7.5.2 Constructor & Destructor Documentation

7.5.2.1 ListRequestAwaiter()

```
HamerSoft.BetterResources.Awaiters.ListRequestAwaiter.ListRequest↵
Awaiter (
            ListRequest asyncOperation )   [inline]
```

Initialized a new instance of ListRequestAwaiter

Parameters

asyncOperation	The ListRequest returned by the PackageManager

7.5.3 Member Function Documentation

7.5.3.1 GetResult()

```
PackageCollection HamerSoft.BetterResources.Awaiters.ListRequestAwaiter.↵
GetResult ( )
```

Get the return value from the ListRequest

Returns

PackageCollection object

7.5.4 Property Documentation

7.5.4.1 IsCompleted `bool HamerSoft.BetterResources.Awaiters.ListRequestAwaiter.IsCompleted [get]`

A flag indicating if the ListRequest is complete

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

- Editor/Awaiters/ListRequestAwaiter.cs

7.6 HamerSoft.BetterResources.QueryBuilder Class Reference

A Query Builder to Search the Unity3D Resources

Inherits IDisposable.

Public Member Functions

- ResourceAsset GetResult< T > ()

  Get a single ResourceAsset result that has a component that matches type T or is derived from T and match the other filters
- IEnumerable< ResourceAsset > GetResults< T > ()

  Get ResourceAsset results that have a component that matches type T or is derived from T and match the other filters
- ResourceAsset GetResult ()

  Get a single ResourceAsset result that match the filters
- IEnumerable< ResourceAsset > GetResults ()

  Get ResourceAsset results that match the filters
- QueryBuilder ByName (string nameFilter, StringComparison comparison=StringComparison.CurrentCulture)

  Add a name filter
- QueryBuilder ByGuid (string guidFilter)

  Add a GUID filter
- QueryBuilder ByGuid (Guid guid)

  Add a GUID filter
- QueryBuilder ByNameSubString (string nameFilter, StringComparison comparison=StringComparison.↵ CurrentCulture)

  Add a name substring filter
- QueryBuilder AtRoot ()

  Add a filter to only find assets that are in the Root Resources folders
- QueryBuilder ByPath (string pathFilter, StringComparison comparison=StringComparison.CurrentCulture)

  Add a filter for an exact directory
- QueryBuilder ByPathSubString (string pathFilter, StringComparison comparison=StringComparison.↵ CurrentCulture)

  Add a path substring filter
- QueryBuilder ByPackage (string packageFilter, StringComparison comparison=StringComparison.Current↵ Culture)

  Add a filter for an exact package name e.g. com.hamersoft.betterresources
- QueryBuilder ByPackageSubString (string packageFilter, StringComparison comparison=String↵ Comparison.CurrentCulture)

  Add a filter for a substring in a package name
- QueryBuilder InPackage (bool inPackage)

  Add a filter to only find or exclude assets in packages
- QueryBuilder WithAllComponents (params Type[ ] components)

Add a filter where ALL given components must be present on the target object(s)

- QueryBuilder WithSomeComponents (params Type[ ] components)

  Add a filter where some of the given components should be present on the target object(s)
- QueryBuilder WithoutAnyComponents (params Type[ ] components)

  Add a filter where Some of the given components must be present on the target object(s)
- QueryBuilder WithoutAllComponents (params Type[ ] components)

  Add a filter where NONE of the given components must be present on the target object(s)

7.6.1 Detailed Description

A Query Builder to Search the Unity3D Resources

This requires a Cache is loaded during BetterResources.Initialize | InitializeAsync | InitializeRoutine.

QueryBuilder implements IDisposable, so use a using statement!

You can no longer change a query after you call GetResult | GetResult<T> | GetResults | GetResults<T>.

7.6.2 Member Function Documentation

7.6.2.1 AtRoot() `QueryBuilder HamerSoft.BetterResources.QueryBuilder.AtRoot ( ) [inline]`

Add a filter to only find assets that are in the Root Resources folders

Any Resources folder is seen as root, also when Resource folders are nested!

Setting a root filter cancels out the ByPath, ByPathSubString and vice-versa.

Returns

The same instance of QueryBuilder but with root Filter added.

7.6.2.2 ByGuid() [1/2] `QueryBuilder HamerSoft.BetterResources.QueryBuilder.ByGuid ( Guid guid ) [inline]`

Add a GUID filter

Parameters

guid	GUID object as given by the UnityEditor.AssetDatabase

Returns

The same instance of QueryBuilder but with GUID Filter added.

7.6.2.3 ByGuid() [2/2]

```
QueryBuilder HamerSoft.BetterResources.QueryBuilder.ByGuid (
        string guidFilter ) [inline]
```

Add a GUID filter

Parameters

guidFilter	String GUID as given by the UnityEditor.AssetDatabase

Returns

The same instance of QueryBuilder but with GUID Filter added.

7.6.2.4 ByName()

```
QueryBuilder HamerSoft.BetterResources.QueryBuilder.ByName (
        string nameFilter,
        StringComparison comparison = StringComparison.CurrentCulture ) [inline]
```

Add a name filter

Parameters

nameFilter	Exact name of the object of interest
comparison	Comparison culture for exact filtering

Default comparison is CurrentCulture

Setting a name filter cancels out the ByNameSubString and vice-versa

Returns

The same instance of QueryBuilder but with Name Filter added.

7.6.2.5 ByNameSubString()

```
QueryBuilder HamerSoft.BetterResources.QueryBuilder.ByNameSubString
(
        string nameFilter,
        StringComparison comparison = StringComparison.CurrentCulture ) [inline]
```

Add a name substring filter

Parameters

nameFilter	Substring of the name of the object of interest.
comparison	Comparison culture for exact filtering.

Default comparison is CurrentCulture.

Setting a substring name filter cancels out the ByName and vice-versa.

Returns

The same instance of QueryBuilder but with NameSubString Filter added.

7.6.2.6 ByPackage()

```
QueryBuilder HamerSoft.BetterResources.QueryBuilder.ByPackage (
            string packageFilter,
            StringComparison comparison = StringComparison.CurrentCulture )  [inline]
```

Add a filter for an exact package name e.g. com.hamersoft.betterresources

Parameters

packageFilter	Package name loaded through the Unity3D PackageManager
comparison	Comparison culture for exact filtering

Default comparison is CurrentCulture

Setting a package filter cancels out the ByPackageSubString and vice-versa.

Returns

The same instance of QueryBuilder but with package Filter added.

7.6.2.7 ByPackageSubString()

```
QueryBuilder HamerSoft.BetterResources.QueryBuilder.ByPackage↩
SubString (
            string packageFilter,
            StringComparison comparison = StringComparison.CurrentCulture )  [inline]
```

Add a filter for a substring in a package name

Parameters

packageFilter	Package name substring loaded through the Unity3D PackageManager
comparison	Comparison culture for exact filtering

Default comparison is CurrentCulture

Setting a package substring filter cancels out the ByPackage and vice-versa.

Returns

The same instance of QueryBuilder but with package substring Filter added.

7.6.2.8 ByPath() `QueryBuilder HamerSoft.BetterResources.QueryBuilder.ByPath ( string pathFilter, StringComparison comparison = StringComparison.CurrentCulture ) [inline]`

Add a filter for an exact directory

Parameters

pathFilter	Directory local to resources
comparison	Comparison culture for exact filtering

Default comparison is CurrentCulture

Setting a path filter cancels out the AtRoot, ByPathSubString and vice-versa.

Returns

The same instance of QueryBuilder but with path Filter added.

7.6.2.9 ByPathSubString() `QueryBuilder HamerSoft.BetterResources.QueryBuilder.ByPathSubString ( string pathFilter, StringComparison comparison = StringComparison.CurrentCulture ) [inline]`

Add a path substring filter

Parameters

pathFilter	Substring of the path of the object(s) of interest.
comparison	Comparison culture for exact filtering.

Default comparison is CurrentCulture.

Setting a substring name filter cancels out the ByPath, AtRoot and vice-versa.

Returns

The same instance of QueryBuilder but with PathSubString Filter added.

7.6.2.10 GetResult() `ResourceAsset HamerSoft.BetterResources.QueryBuilder.GetResult ( ) [inline]`

Get a single ResourceAsset result that match the filters

You can no longer change a query after you call GetResult | GetResult<T> | GetResults | GetResults<T>.

Returns

A ResourceAsset that matches all filters

7.6.2.11 GetResult< T >() `ResourceAsset HamerSoft.BetterResources.QueryBuilder.GetResult< T > ( ) [inline]`

Get a single ResourceAsset result that has a component that matches type T or is derived from T and match the other filters

Template Parameters

T	Generic type filter

You can no longer change a query after you call GetResult | GetResult<T> | GetResults | GetResults<T>.

Returns

A ResourceAsset that matches all filter and type filter T

Type Constraints

T* : *UnityEngine.Object

7.6.2.12 GetResults() `IEnumerable<ResourceAsset> HamerSoft.BetterResources.QueryBuilder.Get↩ Results ( ) [inline]`

Get ResourceAsset results that match the filters

You can no longer change a query after you call GetResult | GetResult<T> | GetResults | GetResults<T>.

Returns

A collection of ResourceAsset that match all filters

7.6.2.13 GetResults< T >() `IEnumerable<ResourceAsset> HamerSoft.BetterResources.QueryBuilder.GetResults< T > ( ) [inline]`

Get ResourceAsset results that have a component that matches type T or is derived from T and match the other filters

Template Parameters

T	Generic type filter

You can no longer change a query after you call GetResult | GetResult<T> | GetResults | GetResults<T>.

Returns

A collection of ResourceAsset that match all filter and type filter T

Type Constraints

T* : *UnityEngine.Object

7.6.2.14 InPackage() `QueryBuilder HamerSoft.BetterResources.QueryBuilder.InPackage ( bool inPackage ) [inline]`

Add a filter to only find or exclude assets in packages

Parameters

inPackage	In Package flag

Returns

The same instance of QueryBuilder but with path Filter added.

7.6.2.15 WithAllComponents() `QueryBuilder HamerSoft.BetterResources.QueryBuilder.WithAllComponents ( params Type[ ] components ) [inline]`

Add a filter where ALL given components must be present on the target object(s)

Parameters

components	Collection of components to filter for

Reminder: Most Prefabs will have a UnityEngine.Transform or UnityEngine.RectTransform.

Returns

The same instance of QueryBuilder but with AllComponents Filter added.

7.6.2.16 WithoutAllComponents() `QueryBuilder HamerSoft.BetterResources.QueryBuilder.WithoutAllComponents ( params Type[ ] components ) [inline]`

Add a filter where NONE of the given components must be present on the target object(s)

Parameters

components	Collection of components to filter for

Returns

The same instance of QueryBuilder but with WithoutAllComponents Filter added.

7.6.2.17 WithoutAnyComponents() `QueryBuilder HamerSoft.BetterResources.QueryBuilder.WithoutAnyComponents ( params Type[ ] components ) [inline]`

Add a filter where Some of the given components must be present on the target object(s)

Parameters

components	Collection of components to filter for

Returns

The same instance of QueryBuilder but with WithoutAnyComponents Filter added.

7.6.2.18 WithSomeComponents() `QueryBuilder HamerSoft.BetterResources.QueryBuilder.WithSomeComponents ( params Type[ ] components ) [inline]`

Add a filter where some of the given components should be present on the target object(s)

Parameters

components	Collection of components to filter for

Returns

The same instance of QueryBuilder but with SomeComponents Filter added.

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

- Runtime/QueryBuilder.cs

7.7 HamerSoft.BetterResources.Awaiters.ResourceRequestAwaiter Struct Reference

Awaiter for UnityEngine.ResourceRequest

Inherits INotifyCompletion.

Public Member Functions

- ResourceRequestAwaiter (ResourceRequest asyncOperation)

  Initializes a new instance of a ResourceRequestAwaiter
- void OnCompleted (Action continuation)
- Object GetResult ()

  Get the result Object of the ResourceRequest

Properties

- bool IsCompleted `[get]`

  a flag indicating the operation is complete

7.7.1 Detailed Description

Awaiter for UnityEngine.ResourceRequest

7.7.2 Constructor & Destructor Documentation

7.7.2.1 ResourceRequestAwaiter()

```
HamerSoft.BetterResources.Awaiters.ResourceRequestAwaiter.↩
ResourceRequestAwaiter (
              ResourceRequest asyncOperation )
```

Initializes a new instance of a ResourceRequestAwaiter

Parameters

asyncOperation	the resource request to wait for

7.7.3 Member Function Documentation

7.7.3.1 GetResult()

```
Object HamerSoft.BetterResources.Awaiters.ResourceRequestAwaiter.GetResult
( )
```

Get the result Object of the ResourceRequest

Returns

7.7.4 Property Documentation

7.7.4.1 IsCompleted `bool HamerSoft.BetterResources.Awaiters.ResourceRequestAwaiter.IsCompleted [get]`

a flag indicating the operation is complete

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

- Runtime/Awaiters/ResourceRequestAwaiter.cs

7.8 HamerSoft.BetterResources.Extensions.TaskExtensions Class Reference

Extensions to use tasks as coroutines

Static Public Member Functions

- static IEnumerator ToCoroutine (this Task task, CancellationToken token=default)

  Convert a task to coroutine
- static IEnumerator ToCoroutine< T > (this Task< T > task, Action< T > callback=null, CancellationToken token=default)

  Convert a task of Type T to a coroutine

7.8.1 Detailed Description

Extensions to use tasks as coroutines

7.8.2 Member Function Documentation

7.8.2.1 ToCoroutine()

```
static IEnumerator HamerSoft.BetterResources.Extensions.TaskExtensions.↵
ToCoroutine (
            this Task task,
            CancellationToken token = default )  [inline], [static]
```

Convert a task to coroutine

Parameters

task	the task
token	optional cancellation token

Returns

IEnumerator to yield in coroutines

7.8.2.2 ToCoroutine< T >() `static IEnumerator HamerSoft.BetterResources.Extensions.TaskExtensions.ToCoroutine< T > (`

```
             this Task< T > task,
             Action< T > callback = null,
             CancellationToken token = default )  [inline], [static]
```

Convert a task of Type T to a coroutine

Parameters

task	to task
callback	callback to catch the return value of task T
token	optional cancellation token

Template Parameters

T	Task Type param

Returns

IEnumerator to yield in coroutines

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

- Runtime/Extensions/TaskExtensions.cs

Index