

# Namespace LlamaLogic.Packages

## Classes

### [DataBasePackedFile](#)

Represents a Maxis DBPF format package file (  )

### [ResourceFileTypeAttribute](#)

Specifies the file type of a package resource type independent of storage within a  [DataBasePackedFile](#)

### [ResourceFileTypeFormatAttribute](#)

Specifies the extension and mime type of a file type of a package resource independent of storage within a  [DataBasePackedFile](#)

### [ResourceToolingMetadataAttribute](#)

Specifies that the  [ResourceType](#)  is tooling metadata and not intended for use by the game

### [SmartSimUtilities](#)

Provides utility methods specific to The Sims 4

## Structs

### [ResourceKey](#)

Represents the unique key of a resource within a  [DataBasePackedFile](#)

## Enums

### [CompressionMode](#)

Represents the compression mode to use when setting the content of a resource in a  [DataBasePackedFile](#)

### [ResourceFileType](#)

The file type of a package resource independent of storage within a  [DataBasePackedFile](#)

### [ResourceKeyOrder](#)

Specifies the order in which a  [ResourceKey](#)  sequence should be processed

### [ResourceType](#)

A type of resource within a  [DataBasePackedFile](#)

# Enum CompressionMode

Namespace: [LlamaLogic.Packages](#)

Assembly: LlamaLogic.Packages.dll

Represents the compression mode to use when setting the content of a resource in a [DataBasePacked File](#)

```
public enum CompressionMode
```

## Fields

**Auto = 0**

The library will choose whether to compress the data or not

**CallerSuppliedInternal = 4**

⚠ The data will not be processed for compression by the library and the resource will be marked as having been compressed using Maxis' internal method (use with caution)

**CallerSuppliedStreamable = 5**

⚠ The data will not be processed for compression by the library and the resource will be marked as having been compressed using Maxis' streamable method (use with caution)

**ForceOff = 1**

The data will not be compressed

**ForceOn = 2**

The data will be compressed

**SetDeletedFlag = 3**

⚠ The data will not be processed for compression by the library—thus callers will need to invoke [ZLibCompress\(ReadOnlyMemory<byte>\)](#) or [ZLibCompressAsync\(ReadOnlyMemory<byte>, CancellationToken\)](#) themselves and pass the result to [Set\(ResourceKey, ReadOnlyMemory<byte>, CompressionMode\)](#) or [SetAsync\(ResourceKey, ReadOnlyMemory<byte>, CompressionMode, CancellationToken\)](#), respectively, if they intend for the content to be compressed with ZLib despite being flagged—and the deleted flag will be set (use with caution)

## Remarks

*Important:* While this library does support the older compression methods, it does so for decompression only. When it does compress resources, it always uses the newer ZLib compression method.

**Warning:** The Sims 4 does not always expect or tolerate ZLib compression for certain types of resources, such as for world package files which ship with the game (see [thepancake1's comment about this on the Creator Musings Discord](#) for more information).

# Class DataBasePackedFile

Namespace: [LlamaLogic.Packages](#)

Assembly: LlamaLogic.Packages.dll

Represents a Maxis DBPF format package file (  )

```
public sealed class DataBasePackedFile : IAsyncDisposable, IDisposable
```

## Inheritance

[object](#) ← DataBasePackedFile

## Implements

[IAsyncDisposable](#), [IDisposable](#)

## Inherited Members

[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.GetHashCode\(\)](#), [object.GetType\(\)](#),  
[object.ReferenceEquals\(object, object\)](#), [object.ToString\(\)](#)

## Remarks

### Streams and Disposal

When a package is opened by this class from a [Stream](#), the class assumes ownership of the [Stream](#). Manipulating the [Stream](#) afterward may result in undefined behavior. The class will dispose of the [Stream](#) when it is, itself, disposed.

To use the [Save\(bool, ResourceKeyOrder\)](#) or [SaveAsync\(bool, ResourceKeyOrder, CancellationToken\)](#) methods, the package must have been opened from a [Stream](#) which is writeable. This can also be determined by the caller in advance of attempting to use them by checking the value of the [CanSaveInPlace](#) property. However, for ease of use in web frameworks like ASP.NET Core, the [CopyTo\(Stream, ResourceKeyOrder\)](#) and [CopyToAsync\(Stream, ResourceKeyOrder, CancellationToken\)](#) methods do *not* require that the [Stream](#) which they are passed is seekable.

### Lazy Loading

When a package is first opened using this class, the package index is read but the content of the package resources is not. All of the retrieval methods of this class have async counterparts for callers who prefer not to be blocked by I/O operations which may be required when retrieving resource content.

Because resource names (when applicable) are determined by their content, this class will not index resource names until they are first referenced, either explicitly by calling [GetNames\(\)](#) or [GetNamesAsync\(CancellationToken\)](#), or implicitly by calling any of the other methods which deal with resource names. Afterward, the names are cached for the lifetime of the package instance and updated as the caller sets the content of resources or deletes them.

## Thread Safety

All properties and methods of this class, including async methods, are kept thread-safe with internally managed locks. Async methods which accept cancellation tokens will honor signals so long as they have not progressed to mutating the state of the package.

## Compression

Decompression of resource content is handled automatically by this class.

When retrieving resource content, this class will automatically decompress it using either proprietary Maxis internal or streaming compression, or ZLib compression. When saving resources, this class will always use ZLib compression. For more details, see the remarks for the [CompressionMode](#) enumeration.

Resource content is compressed in memory when it is set to minimize the amount of memory used while operating on a package.

## Deleted Entries

This class honors the deleted entry flag in the package index under normal circumstances. If a caller attempts to retrieve the content of a resource flagged as deleted, the class will throw a [FileNotFoundException](#) by default. All retrieval methods have an optional `force bool` parameter which, when `true`, will cause the class to attempt to retrieve the content of a resource flagged as deleted (this is not guaranteed to work).

## Constructors

### DataBasePackedFile()

Initializes a new, blank [DataBasePackedFile](#)

```
public DataBasePackedFile()
```

### DataBasePackedFile(Stream)

Initializes a  [DataBasePackedFile](#)  from the specified `stream` (   )

```
public DataBasePackedFile(Stream stream)
```

## Parameters

`stream`  [Stream](#)  

## DataBasePackedFile(string, bool)

Initializes a  [DataBasePackedFile](#)  from the specified `path` (   )

```
public DataBasePackedFile(string path, bool forReadOnly = true)
```

## Parameters

`path`  [string](#)  

`forReadOnly`  [bool](#)  

# Properties

## CanSaveInPlace

Gets whether the package can be saved in place

```
public bool CanSaveInPlace { get; }
```

## Property Value

[bool](#)  

## Count

Gets the number of resources

```
public int Count { get; }
```

## Property Value

[int ↗](#)

## CreationTime

Gets/sets when the package was created

```
public DateTimeOffset CreationTime { get; set; }
```

## Property Value

[DateTimeOffset ↗](#)

## FileVersion

Gets/sets the version of the format of the file

```
public Version FileVersion { get; set; }
```

## Property Value

[Version ↗](#)

## Remarks

This class is intended to work with versions of the Maxis DBPF format up to version [2.1](#) and going back as far as possible. While it will permit the caller to specify any version they like, it does not guard against potential negative consequences which may result from such packages actually being used. The default version for newly created packages is currently [2.1](#).

## this[ResourceKey]

Gets/sets the content of a resource with the specified [key](#)

```
public object? this[ResourceKey key] { get; set; }
```

## Parameters

key [ResourceKey](#)

## Property Value

[object](#)

## Keys

Gets a list of keys for all the resources in the package

```
public IReadOnlyList<ResourceKey> Keys { get; }
```

## Property Value

[IReadOnlyList](#) <[ResourceKey](#)>

## UpdatedTime

Gets/sets when the package was last updated

```
public DateTimeOffset UpdatedTime { get; set; }
```

## Property Value

[DateTimeOffset](#)

## UserVersion

Gets/sets the version of the user (presumably the user agent, actually? -- is Maxis versioning human beings?)

```
public Version UserVersion { get; set; }
```

## Property Value

[Version](#) ↗

## Methods

### ContainsKey(ResourceKey)

Gets whether the package contains a resource with the specified **key**

```
public bool ContainsKey(ResourceKey key)
```

#### Parameters

**key** [ResourceKey](#)

#### Returns

[bool](#) ↗

### ContainsKeyAsync(ResourceKey, CancellationToken)

Gets whether the package contains a resource with the specified **key** asynchronously

```
public Task<bool> ContainsKeyAsync(ResourceKey key, CancellationToken cancellationToken  
= default)
```

#### Parameters

**key** [ResourceKey](#)

**cancellationToken** [CancellationToken](#) ↗

#### Returns

[Task](#) <bool>

## CopyTo(Stream, ResourceKeyOrder)

Copies the package in binary format to the specified [destination](#) (↗)

```
public void CopyTo(Stream destination, ResourceKeyOrder resourceKeyOrder  
= ResourceKeyOrder.Preserve)
```

### Parameters

[destination](#) [Stream](#)

[resourceKeyOrder](#) [ResourceKeyOrder](#)

### Exceptions

[ArgumentNullException](#)

[destination](#) is [null](#)

[ArgumentException](#)

[destination](#) is not writeable

## CopyToAsync(Stream, ResourceKeyOrder, CancellationToken)

Copies the package in binary format to the specified [destination](#) asynchronously (↗)

```
public Task CopyToAsync(Stream destination, ResourceKeyOrder resourceKeyOrder =  
ResourceKeyOrder.Preserve, CancellationToken cancellationToken = default)
```

### Parameters

[destination](#) [Stream](#)

[resourceKeyOrder](#) [ResourceKeyOrder](#)

[cancellationToken](#) [CancellationToken](#)

Returns

[Task](#)

Exceptions

[ArgumentNullException](#)

`destination` is [null](#)

[ArgumentException](#)

`destination` is not writeable

## Delete(ResourceKey)

Deletes the resource with the specified `key` and returns [true](#) if it was found; otherwise, [false](#)

```
public bool Delete(ResourceKey key)
```

Parameters

`key` [ResourceKey](#)

Returns

[bool](#)

## Dispose()

Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.

```
public void Dispose()
```

## DisposeAsync()

Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources asynchronously.

```
public ValueTask DisposeAsync()
```

Returns

[ValueTask](#)

A task that represents the asynchronous dispose operation.

## ~DataBasePackedFile()

Called by the finalizer

```
protected ~DataBasePackedFile()
```

## ForEach(ResourceKeyOrder, Predicate<ResourceKey>, Action<ResourceKey, ReadOnlyMemory<byte>>, CancellationToken)

Processes undeleted resources in the specified `keyOrder` if they satisfy the specified `keyPredicate` using the specified `processResourceAction`

```
public void ForEach(ResourceKeyOrder keyOrder, Predicate<ResourceKey> keyPredicate,
Action<ResourceKey, ReadOnlyMemory<byte>> processResourceAction, CancellationToken
cancellationToken = default)
```

Parameters

`keyOrder` [ResourceKeyOrder](#)

`keyPredicate` [Predicate](#)<[ResourceKey](#)>

`processResourceAction` [Action](#)<[ResourceKey](#), [ReadOnlyMemory](#)<[byte](#)>>>

`cancellationToken` [CancellationToken](#)

## ForEachAsync(ResourceKeyOrder, Predicate<ResourceKey>, Func<ResourceKey, ReadOnlyMemory<byte>, Task>, CancellationToken)

Processes undeleted resources in the specified `keyOrder` if they satisfy the specified `keyPredicate` using the specified `processResourceAsyncAction`

```
public Task ForEachAsync(ResourceKeyOrder keyOrder, Predicate<ResourceKey> keyPredicate,
Func<ResourceKey, ReadOnlyMemory<byte>, Task> processResourceAsyncAction, CancellationToken
cancellationToken = default)
```

### Parameters

`keyOrder` [ResourceKeyOrder](#)

`keyPredicate` [Predicate](#)<[ResourceKey](#)>

`processResourceAsyncAction` [Func](#)<[ResourceKey](#), [ReadOnlyMemory](#)<[byte](#)>, [Task](#)>

`cancellationToken` [CancellationToken](#)

### Returns

[Task](#)

## ForEachRaw(ResourceKeyOrder, Predicate<ResourceKey>, Action<ResourceKey, ReadOnlyMemory<byte>>, CancellationToken)

Processes undeleted raw resources (meaning that the library will not decompress a resource for you if it is compressed) in the specified `keyOrder` if they satisfy the specified `keyPredicate` using the specified `processResourceAction`

```
public void ForEachRaw(ResourceKeyOrder keyOrder, Predicate<ResourceKey> keyPredicate,
Action<ResourceKey, ReadOnlyMemory<byte>> processResourceAction, CancellationToken
cancellationToken = default)
```

### Parameters

keyOrder [ResourceKeyOrder](#)

keyPredicate [Predicate](#)<ResourceKey>

processResourceAction [Action](#)<ResourceKey, [ReadOnlyMemory](#)<byte>>>

cancellationToken [CancellationToken](#)

ForEachRawAsync(ResourceKeyOrder, Predicate<ResourceKey>, Func<ResourceKey, [ReadOnlyMemory](#)<byte>, Task>, CancellationToken)

Processes undeleted raw resources (meaning that the library will not decompress a resource for you if it is compressed) in the specified `keyOrder` if they satisfy the specified `keyPredicate` using the specified `processResourceAsyncAction`

```
public Task ForEachRawAsync(ResourceKeyOrder keyOrder, Predicate<ResourceKey> keyPredicate, Func<ResourceKey, ReadOnlyMemory<byte>, Task> processResourceAsyncAction, CancellationToken cancellationToken = default)
```

## Parameters

keyOrder [ResourceKeyOrder](#)

keyPredicate [Predicate](#)<ResourceKey>

processResourceAsyncAction [Func](#)<ResourceKey, [ReadOnlyMemory](#)<byte>, Task>

cancellationToken [CancellationToken](#)

## Returns

[Task](#)

FromPathAsync(string, bool)

Initializes a [DataBasePackedFile](#) asynchronously from the specified `path` (  )

```
public static Task<DataBasePackedFile> FromPathAsync(string path, bool forReadOnly = true)
```

## Parameters

path [string](#)

forReadOnly [bool](#)

## Returns

[Task](#) <[DataBasePackedFile](#)>

## FromStreamAsync(Stream, CancellationToken)

Initializes a [DataBasePackedFile](#) asynchronously from the specified `stream` ( [zzz](#))

```
public static Task<DataBasePackedFile> FromStreamAsync(Stream stream, CancellationToken  
cancellationToken = default)
```

## Parameters

stream [Stream](#)

cancellationToken [CancellationToken](#)

## Returns

[Task](#) <[DataBasePackedFile](#)>

## Get(ResourceKey, bool)

Gets the content of a resource with the specified `key`

```
public ReadOnlyMemory<byte> Get(ResourceKey key, bool force = false)
```

## Parameters

key [ResourceKey](#)

The key of the resource

**force** `bool`

`true` to get the content of the resource even if it has been marked as deleted; otherwise, `false` (default)

Returns

`ReadOnlyMemory<byte>`

## GetAllSizes(`Predicate<ResourceKey>?`)

Gets the size of the content of each resource in the package indexed by key

```
public IReadOnlyDictionary<ResourceKey, int> GetAllSizes(Predicate<ResourceKey>? predicate = null)
```

Parameters

`predicate` `Predicate<ResourceKey>`

An optional predicate to filter the keys

Returns

`IReadOnlyDictionary<ResourceKey, int>`

## GetAllSizesAsync(`Predicate<ResourceKey>?`, `CancellationToken`)

Gets the size of the content of each resource in the package indexed by key, asynchronously

```
public Task<IReadOnlyDictionary<ResourceKey, int>> GetAllSizesAsync(Predicate<ResourceKey>? predicate = null, CancellationToken cancellationToken = default)
```

Parameters

`predicate` `Predicate<ResourceKey>`

An optional predicate to filter the keys

cancellationToken [CancellationToken](#)

The token to monitor for cancellation requests

Returns

[Task](#) <[IReadOnlyDictionary](#) <[ResourceKey](#), [int](#)>>

## GetAsync(ResourceKey, bool, CancellationToken)

Gets the content of a resource with the specified [key](#) asynchronously

```
public Task<ReadOnlyMemory<byte>> GetAsync(ResourceKey key, bool force = false,  
CancellationToken cancellationToken = default)
```

Parameters

[key](#) [ResourceKey](#)

The key of the resource

[force](#) [bool](#)

[true](#) to get the content of the resource even if it has been marked as deleted; otherwise, [false](#) (default)

[cancellationToken](#) [CancellationToken](#)

The token to monitor for cancellation requests

Returns

[Task](#) <[ReadOnlyMemory](#) <[byte](#)>>

## GetCount()

Gets the number of resources

```
public int GetCount()
```

Returns

[int](#)

## GetCountAsync(CancellationToken)

Gets the number of resources asynchronously

```
public Task<int> GetCountAsync(CancellationToken cancellationToken = default)
```

Parameters

cancellationToken  [CancellationToken](#)

Returns

[Task](#)<[int](#)>

## GetData(ResourceKey, bool)

Gets the content of a resource with the specified [key](#) as a [DataModel](#)

```
public DataModel GetData(ResourceKey key, bool force = false)
```

Parameters

key [ResourceKey](#)

force [bool](#)

Returns

[DataModel](#)

## GetDataAsync(ResourceKey, bool, CancellationToken)

Gets the content of a resource with the specified [key](#) as a [DataModel](#) asynchronously

```
public Task<DataModel> GetDataAsync(ResourceKey key, bool force = false, CancellationToken cancellationToken = default)
```

## Parameters

key [ResourceKey](#)

force [bool](#)

cancellationToken [CancellationToken](#)

## Returns

[Task](#) <[DataModel](#)>

## GetKeys(ResourceKeyOrder)

Gets a list of keys for all the resources in the package

```
public IReadOnlyList<ResourceKey> GetKeys(ResourceKeyOrder resourceKeyOrder = ResourceKeyOrder.Preserve)
```

## Parameters

resourceKeyOrder [ResourceKeyOrder](#)

## Returns

[IReadOnlyList](#) <[ResourceKey](#)>

## GetKeysAsync(ResourceKeyOrder, CancellationToken)

Gets a list of keys for all the resources in the package asynchronously

```
public Task<IReadOnlyList<ResourceKey>> GetKeysAsync(ResourceKeyOrder resourceKeyOrder = ResourceKeyOrder.Preserve, CancellationToken cancellationToken = default)
```

Parameters

`resourceKeyOrder` [ResourceKeyOrder](#)

`cancellationToken` [CancellationToken](#)

Returns

[Task](#) <[IReadOnlyList](#) <[ResourceKey](#)>>

## GetKeysByName(string)

Gets a list of keys for resources with the specified `name`

```
public IReadOnlyList<ResourceKey> GetKeysByName(string name)
```

Parameters

`name` [string](#)

Returns

[IReadOnlyList](#) <[ResourceKey](#)>

## GetKeysByNameAsync(string, CancellationToken)

Gets a list of keys for resources with the specified `name` asynchronously

```
public Task<IReadOnlyList<ResourceKey>> GetKeysByNameAsync(string name, CancellationToken cancellationToken = default)
```

Parameters

`name` [string](#)

`cancellationToken` [CancellationToken](#)

Returns

[Task](#) <IReadOnlyList<[ResourceKey](#)>>

## GetModFileManifest(ResourceKey, bool)

Gets the content of a resource with the specified `key` as a [ModFileManifestModel](#)

```
public ModFileManifestModel GetModFileManifest(ResourceKey key, bool force = false)
```

Parameters

`key` [ResourceKey](#)

`force` [bool](#)

Returns

[ModFileManifestModel](#)

## GetModFileManifestAsync(ResourceKey, bool, CancellationToken)

Gets the content of a resource with the specified `key` as a [ModFileManifestModel](#) asynchronously

```
public Task<ModFileManifestModel> GetModFileManifestAsync(ResourceKey key, bool force = false, CancellationToken cancellationToken = default)
```

Parameters

`key` [ResourceKey](#)

`force` [bool](#)

`cancellationToken` [CancellationToken](#)

Returns

[Task](#) <[ModFileManifestModel](#)>

## GetModelAsync<TModel>(ResourceKey, bool, CancellationToken)

Gets the content of a resource with the specified `key` as a `TModel` asynchronously

```
public Task<TModel> GetModelAsync<TModel>(ResourceKey key, bool force = false,  
CancellationToken cancellationToken = default) where TModel : IModel<TModel>
```

Parameters

`key` [ResourceKey](#)

`force` [bool](#)

`cancellationToken` [CancellationToken](#)

Returns

[Task](#) <TModel>

Type Parameters

`TModel`

A model used to browse and modify certain types of resources

Exceptions

[ArgumentException](#)

`TModel` cannot deal with resources of the type specified by `key`

## GetModel<TModel>(ResourceKey, bool)

Gets the content of a resource with the specified `key` as a `TModel`

```
public TModel GetModel<TModel>(ResourceKey key, bool force = false) where TModel  
: IModel<TModel>
```

Parameters

**key** [ResourceKey](#)

**force** [bool](#)

Returns

TModel

Type Parameters

[TModel](#)

A model used to browse and modify certain types of resources

Exceptions

[ArgumentException](#)

[TModel](#) cannot deal with resources of the type specified by **key**

## GetNameByKey(ResourceKey)

Gets the name for the resource with the specified **key**

`public string? GetNameByKey(ResourceKey key)`

Parameters

**key** [ResourceKey](#)

Returns

[string](#)

## GetNameByKeyAsync(ResourceKey, CancellationToken)

Gets the name for the resource with the specified **key** asynchronously

`public Task<string?> GetNameByKeyAsync(ResourceKey key, CancellationToken cancellationToken)`

```
= default)
```

## Parameters

key [ResourceKey](#)

cancellationToken [CancellationToken](#)

## Returns

[Task](#) <[string](#)>

## GetNames()

Gets the names for all the resources in the package

```
public IReadOnlyList<string> GetNames()
```

## Returns

[IReadOnlyList](#) <[string](#)>

## GetNamesAsync(CancellationToken)

Gets the names for all the resources in the package asynchronously

```
public Task<IReadOnlyList<string>> GetNamesAsync(CancellationToken cancellationToken  
= default)
```

## Parameters

cancellationToken [CancellationToken](#)

## Returns

[Task](#) <[IReadOnlyList](#) <[string](#)>>

## GetRaw(ResourceKey, bool)

Gets the raw content of a resource with the specified `key`, meaning that the library will not decompress it for you if it is compressed

```
public ReadOnlyMemory<byte> GetRaw(ResourceKey key, bool force = false)
```

### Parameters

`key` [ResourceKey](#)

The key of the resource

`force` [bool](#)

[true](#) to get the content of the resource even if it has been marked as deleted; otherwise, [false](#) (default)

### Returns

[ReadOnlyMemory](#) <[byte](#)>

## GetRawAsync(ResourceKey, bool, CancellationToken)

Gets the raw content of a resource with the specified `key` asynchronously, meaning that the library will not decompress it for you if it is compressed

```
public Task<ReadOnlyMemory<byte>> GetRawAsync(ResourceKey key, bool force = false, CancellationToken cancellationToken = default)
```

### Parameters

`key` [ResourceKey](#)

The key of the resource

`force` [bool](#)

[true](#) to get the content of the resource even if it has been marked as deleted; otherwise, [false](#) (default)

cancellationToken [CancellationToken](#)

The token to monitor for cancellation requests

Returns

[Task](#) <[ReadOnlyMemory](#) <[byte](#)>>

## GetSize(ResourceKey)

Gets the size of the content of the resource with the specified **key**

```
public int GetSize(ResourceKey key)
```

Parameters

**key** [ResourceKey](#)

Returns

[int](#)

## GetSizeAsync(ResourceKey, CancellationToken)

Gets the size of the content of the resource with the specified **key** asynchronously

```
public Task<int> GetSizeAsync(ResourceKey key, CancellationToken cancellationToken  
= default)
```

Parameters

**key** [ResourceKey](#)

**cancellationToken** [CancellationToken](#)

Returns

[Task](#) <[int](#)>

## GetStringTable(ResourceKey, bool)

Gets the content of a resource with the specified `key` as a [StringTableModel](#)

```
public StringTableModel GetStringTable(ResourceKey key, bool force = false)
```

Parameters

`key` [ResourceKey](#)

`force` [bool](#)

Returns

[StringTableModel](#)

## GetStringTableAsync(ResourceKey, bool, CancellationToken)

Gets the content of a resource with the specified `key` as a [StringTableModel](#) asynchronously

```
public Task<StringTableModel> GetStringTableAsync(ResourceKey key, bool force = false,  
CancellationToken cancellationToken = default)
```

Parameters

`key` [ResourceKey](#)

`force` [bool](#)

`cancellationToken` [CancellationToken](#)

Returns

[Task](#) <[StringTableModel](#)>

## GetText(ResourceKey, bool)

Gets the content of a resource with the specified `key` as a [string](#)

```
public string GetText(ResourceKey key, bool force = false)
```

## Parameters

**key** [ResourceKey](#)

The key of the resource

**force** [bool](#)

[true](#) to get the content of the resource even if it has been marked as deleted; otherwise, [false](#) (default)

## Returns

[string](#)

## GetTextAsync(ResourceKey, bool, CancellationToken)

Gets the content of a resource with the specified **key** as a [string](#) asynchronously

```
public Task<string> GetTextAsync(ResourceKey key, bool force = false, CancellationToken cancellationToken = default)
```

## Parameters

**key** [ResourceKey](#)

The key of the resource

**force** [bool](#)

[true](#) to get the content of the resource even if it has been marked as deleted; otherwise, [false](#) (default)

**cancellationToken** [CancellationToken](#)

The token to monitor for cancellation requests

## Returns

[Task](#) <string>

## GetXml(ResourceKey, bool)

Gets the content of a resource with the specified `key` as an [XDocument](#)

```
public XDocument GetXml(ResourceKey key, bool force = false)
```

### Parameters

`key` [ResourceKey](#)

The key of the resource

`force` [bool](#)

`true` to get the content of the resource even if it has been marked as deleted; otherwise, `false` (default)

### Returns

[XDocument](#)

## GetXmlAsync(ResourceKey, bool, CancellationToken)

Gets the content of a resource with the specified `key` as an [XDocument](#) asynchronously

```
public Task<XDocument> GetXmlAsync(ResourceKey key, bool force = false, CancellationToken cancellationToken = default)
```

### Parameters

`key` [ResourceKey](#)

The key of the resource

`force` [bool](#)

`true` to get the content of the resource even if it has been marked as deleted; otherwise, `false` (default)

`cancellationToken` [CancellationToken](#)

The token to monitor for cancellation requests

Returns

[Task](#) <[XDocument](#)>

## LoadAll(bool, CompressionMode)

Loads all resources in the package into memory

```
public void LoadAll(bool force = false, CompressionMode compressionMode  
= CompressionMode.Auto)
```

Parameters

`force` [bool](#)

[true](#) to get the content of the resources even if they has been marked as deleted; otheriwse, [false](#)  
(default)

`compressionMode` [CompressionMode](#)

The compression mode to use for resources in memory

## LoadAllAsync(bool, CompressionMode, CancellationToken)

Loads all resources in the package into memory

```
public Task LoadAllAsync(bool force = false, CompressionMode compressionMode =  
CompressionMode.Auto, CancellationToken cancellationToken = default)
```

Parameters

`force` [bool](#)

[true](#) to get the content of the resources even if they has been marked as deleted; otheriwse, [false](#)  
(default)

`compressionMode` [CompressionMode](#)

The compression mode to use for resources in memory

`cancellationToken` [CancellationToken](#)

The token to monitor for cancellation requests

Returns

[Task](#)

## LoadNames()

Loads the names of all the resources in the package (a no-op if the names are already loaded)

```
public void LoadNames()
```

## LoadNamesAsync(CancellationToken)

Loads the names of all the resources in the package asynchronously (a no-op if the names are already loaded)

```
public Task LoadNamesAsync(CancellationToken cancellationToken = default)
```

Parameters

`cancellationToken` [CancellationToken](#)

Returns

[Task](#)

## Save(bool, ResourceKeyOrder)

Saves the package to the stream from which it was loaded

```
public void Save(bool unloadFromMemory = false, ResourceKeyOrder resourceKeyOrder = ResourceKeyOrder.Preserve)
```

## Parameters

unloadFromMemory [bool](#)

resourceKeyOrder [ResourceKeyOrder](#)

## Exceptions

[InvalidOperationException](#)

The package was not loaded from stream or the stream is not writeable

## SaveAs(string, ResourceKeyOrder)

Saves the package to the specified [path](#) (   )

```
public void SaveAs(string path, ResourceKeyOrder resourceKeyOrder = ResourceKeyOrder.Preserve)
```

## Parameters

path [string](#)

resourceKeyOrder [ResourceKeyOrder](#)

## SaveAsAsync(string, ResourceKeyOrder, CancellationToken)

Saves the package to the specified [path](#) asynchronously (   )

```
public Task SaveAsAsync(string path, ResourceKeyOrder resourceKeyOrder = ResourceKeyOrder.Preserve, CancellationToken cancellationToken = default)
```

## Parameters

path [string](#)

`resourceKeyOrder` [ResourceKeyOrder](#)

`cancellationToken` [CancellationToken](#)

Returns

[Task](#)

## SaveAsync(bool, ResourceKeyOrder, CancellationToken)

Saves the package to the stream from which it was loaded asynchronously

```
public Task SaveAsync(bool unloadFromMemory = false, ResourceKeyOrder resourceKeyOrder = ResourceKeyOrder.Preserve, CancellationToken cancellationToken = default)
```

Parameters

`unloadFromMemory` [bool](#)

`resourceKeyOrder` [ResourceKeyOrder](#)

`cancellationToken` [CancellationToken](#)

Returns

[Task](#)

Exceptions

[InvalidOperationException](#)

The package was not loaded from stream or the stream is not writeable

## Set(ResourceKey, IModel, CompressionMode)

Sets the `content` of a resource with the specified `key`, returning [true](#) if the resource was compressed; otherwise, [false](#)

```
public bool Set(ResourceKey key, IModel content, CompressionMode compressionMode)
```

```
= CompressionMode.Auto)
```

## Parameters

**key** [ResourceKey](#)

The key of the resource

**content** [IModel](#)

The content of the resource

**compressionMode** [CompressionMode](#)

The compression mode to use for the resource

## Returns

[bool](#)

## Set(ResourceKey, ReadOnlyMemory<byte>, CompressionMode)

Sets the **content** of a resource with the specified **key**, returning [true](#) if the resource was compressed; otherwise, [false](#)

```
public bool Set(ResourceKey key, ReadOnlyMemory<byte> content, CompressionMode  
compressionMode = CompressionMode.Auto)
```

## Parameters

**key** [ResourceKey](#)

The key of the resource

**content** [ReadOnlyMemory<byte>](#)

The content of the resource

**compressionMode** [CompressionMode](#)

The compression mode to use for the resource

Returns

[bool](#)

## Set(ResourceKey, string, CompressionMode)

Sets the [content](#) of a resource with the specified [key](#), returning [true](#) if the resource was compressed; otherwise, [false](#)

```
public bool Set(ResourceKey key, string content, CompressionMode compressionMode  
= CompressionMode.Auto)
```

Parameters

[key](#) [ResourceKey](#)

The key of the resource

[content](#) [string](#)

The content of the resource

[compressionMode](#) [CompressionMode](#)

The compression mode to use for the resource

Returns

[bool](#)

## Set(ResourceKey, XDocument, CompressionMode)

Sets the [content](#) of a resource with the specified [key](#), returning [true](#) if the resource was compressed; otherwise, [false](#)

```
public bool Set(ResourceKey key, XDocument content, CompressionMode compressionMode  
= CompressionMode.Auto)
```

Parameters

**key** [ResourceKey](#)

The key of the resource

**content** [XDocument](#)

The content of the resource

**compressionMode** [CompressionMode](#)

The compression mode to use for the resource

Returns

[bool](#)

## SetAsync(ResourceKey, IMModel, CompressionMode, CancellationToken)

Sets the **content** of a resource with the specified **key**, returning [true](#) if the resource was compressed; otherwise, [false](#)

```
public Task<bool> SetAsync(ResourceKey key, IMModel content, CompressionMode compressionMode  
= CompressionMode.Auto, CancellationToken cancellationToken = default)
```

Parameters

**key** [ResourceKey](#)

The key of the resource

**content** [IMModel](#)

The content of the resource

**compressionMode** [CompressionMode](#)

The compression mode to use for the resource

**cancellationToken** [CancellationToken](#)

The token to monitor for cancellation requests

Returns

[Task](#) <bool>

## SetAsync(ResourceKey, ReadOnlyMemory<byte>, CompressionMode, CancellationToken)

Sets the `content` of a resource with the specified `key`, returning [true](#) if the resource was compressed; otherwise, [false](#)

```
public Task<bool> SetAsync(ResourceKey key, ReadOnlyMemory<byte> content, CompressionMode  
compressionMode = CompressionMode.Auto, CancellationToken cancellationToken = default)
```

Parameters

`key` [ResourceKey](#)

The key of the resource

`content` [ReadOnlyMemory](#) <byte>

The content of the resource

`compressionMode` [CompressionMode](#)

The compression mode to use for the resource

`cancellationToken` [CancellationToken](#)

The token to monitor for cancellation requests

Returns

[Task](#) <bool>

## SetAsync(ResourceKey, string, CompressionMode, CancellationToken)

Sets the `content` of a resource with the specified `key`, returning [true](#) if the resource was compressed; otherwise, [false](#)

```
public Task<bool> SetAsync(ResourceKey key, string content, CompressionMode compressionMode = CompressionMode.Auto, CancellationToken cancellationToken = default)
```

## Parameters

### key [ResourceKey](#)

The key of the resource

### content [string](#)

The content of the resource

### compressionMode [CompressionMode](#)

The compression mode to use for the resource

### cancellationToken [CancellationToken](#)

The token to monitor for cancellation requests

## Returns

### [Task](#)<[bool](#)>

## SetAsync(ResourceKey, XDocument, CompressionMode, CancellationToken)

Sets the [content](#) of a resource with the specified [key](#), returning [true](#) if the resource was compressed; otherwise, [false](#)

```
public Task<bool> SetAsync(ResourceKey key, XDocument content, CompressionMode compressionMode = CompressionMode.Auto, CancellationToken cancellationToken = default)
```

## Parameters

### key [ResourceKey](#)

The key of the resource

### content [XDocument](#)

The content of the resource

`compressionMode` [CompressionMode](#)

The compression mode to use for the resource

`cancellationToken` [CancellationToken](#)

The token to monitor for cancellation requests

Returns

[Task](#) <[bool](#)>

## SetXml(ResourceKey, string, CompressionMode)

Sets the `xmlContent` of a resource with the specified `key`, returning [true](#) if the resource was compressed; otherwise, [false](#)

```
public bool SetXml(ResourceKey key, string xmlContent, CompressionMode compressionMode  
= CompressionMode.Auto)
```

Parameters

`key` [ResourceKey](#)

The key of the resource

`xmlContent` [string](#)

The XML content of the resource

`compressionMode` [CompressionMode](#)

The compression mode to use for the resource

Returns

[bool](#)

## SetXmlAsync(ResourceKey, string, CompressionMode, CancellationToken)

Sets the `xmlContent` of a resource with the specified `key`, returning `true` if the resource was compressed; otherwise, `false`

```
public Task<bool> SetXmlAsync(ResourceKey key, string xmlContent, CompressionMode  
compressionMode = CompressionMode.Auto, CancellationToken cancellationToken = default)
```

### Parameters

`key` [ResourceKey](#)

The key of the resource

`xmlContent` [string](#)

The XML content of the resource

`compressionMode` [CompressionMode](#)

The compression mode to use for the resource

`cancellationToken` [CancellationToken](#)

The token to monitor for cancellation requests

### Returns

[Task](#) <[bool](#)>

## ZLibCompress(ReadOnlyMemory<byte>)

Compresses the specified `memory` using ZLib, returning the result

```
public static ReadOnlyMemory<byte> ZLibCompress(ReadOnlyMemory<byte> memory)
```

### Parameters

`memory` [ReadOnlyMemory](#) <[byte](#)>

Returns

[ReadOnlyMemory](#) <[byte](#)>

## ZLibCompressAsync(ReadOnlyMemory<byte>, CancellationToken)

Compresses the specified `memory` using ZLib asynchronously, returning the result

```
public static Task<ReadOnlyMemory<byte>> ZLibCompressAsync(ReadOnlyMemory<byte> memory, CancellationToken cancellationToken = default)
```

Parameters

`memory` [ReadOnlyMemory](#) <[byte](#)>

`cancellationToken` [CancellationToken](#)

Returns

[Task](#) <[ReadOnlyMemory](#) <[byte](#)>>

# Enum ResourceType

Namespace: [LlamaLogic.Packages](#)

Assembly: LlamaLogic.Packages.dll

The file type of a package resource independent of storage within a [DataBasePackedFile](#)

```
public enum ResourceType
```

## Fields

[ResourceFileTypeFormat("bin", "application/octet-stream")] Binary = 0

Binary

[ResourceFileTypeFormat("dds", "image/vnd.ms-dds")] DirectDrawSurface = 1

DDS Image

[ResourceFileTypeFormat("json", "application/json")] JavaScriptObjectNotation = 2

JavaScript Object Notation

[ResourceFileTypeFormat("png", "image/png")] PortableNetworkGraphic = 3

PNG Image

[ResourceFileTypeFormat("xml", "application/xml")] TuningMarkup = 4

Tuning XML

[ResourceFileTypeFormat("yml", "application/x-yaml")] YetAnotherMarkupLanguage = 5

Yet Another Markup Language

# Class ResourceTypeAttribute

Namespace: [LlamaLogic.Packages](#)

Assembly: LlamaLogic.Packages.dll

Specifies the file type of a package resource type independent of storage within a [DataBasePackedFile](#)

```
[AttributeUsage(AttributeTargets.Field, AllowMultiple = false)]
public sealed class ResourceTypeAttribute : Attribute
```

## Inheritance

[object](#) ← [Attribute](#) ← ResourceTypeAttribute

## Inherited Members

[Attribute.Equals\(object\)](#) , [Attribute.GetCustomAttribute\(Assembly, Type\)](#) ,  
[Attribute.GetCustomAttribute\(Assembly, Type, bool\)](#) ,  
[Attribute.GetCustomAttribute\(MemberInfo, Type\)](#) ,  
[Attribute.GetCustomAttribute\(MemberInfo, Type, bool\)](#) ,  
[Attribute.GetCustomAttribute\(Module, Type\)](#) , [Attribute.GetCustomAttribute\(Module, Type, bool\)](#) ,  
[Attribute.GetCustomAttribute\(ParameterInfo, Type\)](#) ,  
[Attribute.GetCustomAttribute\(ParameterInfo, Type, bool\)](#) , [Attribute.GetCustomAttributes\(Assembly\)](#) ,  
[Attribute.GetCustomAttributes\(Assembly, bool\)](#) , [Attribute.GetCustomAttributes\(Assembly, Type\)](#) ,  
[Attribute.GetCustomAttributes\(Assembly, Type, bool\)](#) , [Attribute.GetCustomAttributes\(MemberInfo\)](#) ,  
[Attribute.GetCustomAttributes\(MemberInfo, bool\)](#) ,  
[Attribute.GetCustomAttributes\(MemberInfo, Type\)](#) ,  
[Attribute.GetCustomAttributes\(MemberInfo, Type, bool\)](#) , [Attribute.GetCustomAttributes\(Module\)](#) ,  
[Attribute.GetCustomAttributes\(Module, bool\)](#) , [Attribute.GetCustomAttributes\(Module, Type\)](#) ,  
[Attribute.GetCustomAttributes\(Module, Type, bool\)](#) , [Attribute.GetCustomAttributes\(ParameterInfo\)](#) ,  
[Attribute.GetCustomAttributes\(ParameterInfo, bool\)](#) ,  
[Attribute.GetCustomAttributes\(ParameterInfo, Type\)](#) ,  
[Attribute.GetCustomAttributes\(ParameterInfo, Type, bool\)](#) , [Attribute.GetHashCode\(\)](#) ,  
[Attribute.IsDefaultAttribute\(\)](#) , [Attribute.IsDefined\(Assembly, Type\)](#) ,  
[Attribute.IsDefined\(Assembly, Type, bool\)](#) , [Attribute.IsDefined\(MemberInfo, Type\)](#) ,  
[Attribute.IsDefined\(MemberInfo, Type, bool\)](#) , [Attribute.IsDefined\(Module, Type\)](#) ,  
[Attribute.IsDefined\(Module, Type, bool\)](#) , [Attribute.IsDefined\(ParameterInfo, Type\)](#) ,  
[Attribute.IsDefined\(ParameterInfo, Type, bool\)](#) , [Attribute.Match\(object\)](#) , [Attribute.TypeId](#) ,  
[object.Equals\(object, object\)](#) , [object.GetType\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Constructors

### ResourceFileTypeAttribute(ResourceFileType)

Specifies the file type of a package resource type independent of storage within a  [DataBasePackedFile](#)

```
public ResourceFileTypeAttribute(ResourceFileType packageResourceFileType)
```

#### Parameters

packageResourceFileType  [ResourceFileType](#)

## Properties

### ResourceFileType

Gets the file type of the decorated package resource type independent of storage within a  [DataBasePackedFile](#)

```
public ResourceFileType ResourceFileType { get; }
```

#### Property Value

[ResourceFileType](#)

# Class ResourceTypeFormatAttribute

Namespace: [LlamaLogic.Packages](#)

Assembly: LlamaLogic.Packages.dll

Specifies the extension and mime type of a file type of a package resource independent of storage within a  [DataBasePackedFile](#)

```
[AttributeUsage(AttributeTargets.Field, AllowMultiple = false)]
public sealed class ResourceTypeFormatAttribute : Attribute
```

## Inheritance

[object](#) ← [Attribute](#) ← ResourceTypeFormatAttribute

## Inherited Members

[Attribute.Equals\(object\)](#) , [Attribute.GetCustomAttribute\(Assembly, Type\)](#) ,  
[Attribute.GetCustomAttribute\(Assembly, Type, bool\)](#) ,  
[Attribute.GetCustomAttribute\(MemberInfo, Type\)](#) ,  
[Attribute.GetCustomAttribute\(MemberInfo, Type, bool\)](#) ,  
[Attribute.GetCustomAttribute\(Module, Type\)](#) , [Attribute.GetCustomAttribute\(Module, Type, bool\)](#) ,  
[Attribute.GetCustomAttribute\(ParameterInfo, Type\)](#) ,  
[Attribute.GetCustomAttribute\(ParameterInfo, Type, bool\)](#) , [Attribute.GetCustomAttributes\(Assembly\)](#) ,  
[Attribute.GetCustomAttributes\(Assembly, bool\)](#) , [Attribute.GetCustomAttributes\(Assembly, Type\)](#) ,  
[Attribute.GetCustomAttributes\(Assembly, Type, bool\)](#) , [Attribute.GetCustomAttributes\(MemberInfo\)](#) ,  
[Attribute.GetCustomAttributes\(MemberInfo, bool\)](#) ,  
[Attribute.GetCustomAttributes\(MemberInfo, Type\)](#) ,  
[Attribute.GetCustomAttributes\(MemberInfo, Type, bool\)](#) , [Attribute.GetCustomAttributes\(Module\)](#) ,  
[Attribute.GetCustomAttributes\(Module, bool\)](#) , [Attribute.GetCustomAttributes\(Module, Type\)](#) ,  
[Attribute.GetCustomAttributes\(Module, Type, bool\)](#) , [Attribute.GetCustomAttributes\(ParameterInfo\)](#) ,  
[Attribute.GetCustomAttributes\(ParameterInfo, bool\)](#) ,  
[Attribute.GetCustomAttributes\(ParameterInfo, Type\)](#) ,  
[Attribute.GetCustomAttributes\(ParameterInfo, Type, bool\)](#) , [Attribute.GetHashCode\(\)](#) ,  
[Attribute.IsDefaultAttribute\(\)](#) , [Attribute.IsDefined\(Assembly, Type\)](#) ,  
[Attribute.IsDefined\(Assembly, Type, bool\)](#) , [Attribute.IsDefined\(MemberInfo, Type\)](#) ,  
[Attribute.IsDefined\(MemberInfo, Type, bool\)](#) , [Attribute.IsDefined\(Module, Type\)](#) ,  
[Attribute.IsDefined\(Module, Type, bool\)](#) , [Attribute.IsDefined\(ParameterInfo, Type\)](#) ,  
[Attribute.IsDefined\(ParameterInfo, Type, bool\)](#) , [Attribute.Match\(object\)](#) , [Attribute.TypeId](#) ,  
[object.Equals\(object, object\)](#) , [object.GetType\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

# Constructors

## ResourceFileTypeFormatAttribute(string, string?)

Specifies the extension and mime type of a file type of a package resource independent of storage within a  [DataBasePackedFile](#)

```
public ResourceFileTypeFormatAttribute(string extension, string? mimeType = null)
```

### Parameters

extension [string](#)

mimeType [string](#)

# Properties

## Extension

Gets the extention used for files in the format

```
public string Extension { get; }
```

### Property Value

[string](#)

## MimeType

Gets the MIME type of the format

```
public string?MimeType { get; }
```

### Property Value

[string](#)



# Struct ResourceKey

Namespace: [LlamaLogic.Packages](#)

Assembly: LlamaLogic.Packages.dll

Represents the unique key of a resource within a [DataBasePackedFile](#)

```
public struct ResourceKey : IEquatable<ResourceKey>
```

Implements

[IEquatable](#) <ResourceKey>

Inherited Members

[object.Equals\(object, object\)](#) , [object.GetType\(\)](#) , [object.ReferenceEquals\(object, object\)](#)

Extension Methods

[SmartSimUtilities.GetResourceKeyForLocale\(ResourceKey, CultureInfo\)](#) ,

[SmartSimUtilities.GetStringTableLocale\(ResourceKey\)](#).

## Constructors

### ResourceKey(ResourceType, uint, ulong)

Represents the unique key of a resource within a [DataBasePackedFile](#)

```
public ResourceKey(ResourceType type, uint group, ulong fullInstance)
```

Parameters

type  [ResourceType](#)

group  [uint](#)

fullInstance  [ulong](#)

## Fields

## Empty

Represents the empty [ResourceKey](#)

```
public static readonly ResourceKey Empty
```

### Field Value

[ResourceKey](#)

## FullInstance

A value that uniquely identifies an individual resource within a given type and group

```
public readonly ulong FullInstance
```

### Field Value

[ulong](#)

## Group

A value used to group related resources together

```
public readonly uint Group
```

### Field Value

[uint](#)

## Remarks

In many cases, this value can be 0, especially for custom content. However, Maxis content and some specific modding scenarios might use different group values to organize resources. Group values help in categorizing resources that belong to a specific set or context.

## Type

The type of resource indicated by the key

```
public readonly ResourceType Type
```

### Field Value

[ResourceType](#)

## Properties

### FullInstanceHex

Gets the hex representation of [FullInstance](#)

```
public readonly string FullInstanceHex { get; }
```

### Property Value

[string](#)

### FullTgi

Gets the full TGI or TGI ID (type, group, and instance in hex format) of the resource key

```
public readonly string FullTgi { get; }
```

### Property Value

[string](#)

### GroupHex

Gets the hex representation of [Group](#)

```
public readonly string GroupHex { get; }
```

Property Value

[string](#)

## Instance

Gets the low order bits of [FullInstance](#)

```
public readonly uint Instance { get; }
```

Property Value

[uint](#)

## InstanceEx

Gets the high order bits of [FullInstance](#)

```
public readonly uint InstanceEx { get; }
```

Property Value

[uint](#)

## InstanceExHex

Gets the hex representation of [InstanceEx](#)

```
public readonly string InstanceExHex { get; }
```

Property Value

[string](#)

## InstanceHex

Gets the hex representation of [Instance](#)

```
public readonly string InstanceHex { get; }
```

Property Value

[string](#) ↗

## TypeHex

Gets the hex representation of [Type](#)

```
public readonly string TypeHex { get; }
```

Property Value

[string](#) ↗

## Methods

### Deconstruct(out ResourceType, out uint, out ulong)

Deconstructs the key into its components

```
public readonly void Deconstruct(out ResourceType type, out uint group, out  
ulong fullInstance)
```

Parameters

**type** [ResourceType](#)

The [Type](#)

**group** [uint](#) ↗

The [Group](#)

`fullInstance` [ulong](#)

The [FullInstance](#)

## Equals(ResourceKey)

Indicates whether the current object is equal to another object of the same type.

```
public readonly bool Equals(ResourceKey other)
```

### Parameters

`other` [ResourceKey](#)

An object to compare with this object.

### Returns

[bool](#)

[true](#) if the current object is equal to the `other` parameter; otherwise, [false](#).

## Equals(object?)

Indicates whether this instance and a specified object are equal.

```
public override readonly bool Equals(object? obj)
```

### Parameters

`obj` [object](#)

The object to compare with the current instance.

### Returns

[bool](#)

[true](#) if `obj` and this instance are the same type and represent the same value; otherwise, [false](#).

## GetHashCode()

Returns the hash code for this instance.

```
public override readonly int GetHashCode()
```

Returns

[int](#)

A 32-bit signed integer that is the hash code for this instance.

## Parse(string)

Parses a string into a [ResourceKey](#)

```
public static ResourceKey Parse(string s)
```

Parameters

[s](#) [string](#)

The string to parse

Returns

[ResourceKey](#)

Exceptions

[FormatException](#)

[s](#) is not in the correct format

## Parse(string, IFormatProvider?)

Parses a string into a [ResourceKey](#)

```
public static ResourceKey Parse(string s, IFormatProvider? provider)
```

## Parameters

s [string](#)

The string to parse

provider [IFormatProvider](#)

An object that provides culture-specific formatting information about s

## Returns

[ResourceKey](#)

## Exceptions

[FormatException](#)

s is not in the correct format

## ToString()

Returns the fully qualified type name of this instance.

```
public override readonly string ToString()
```

## Returns

[string](#)

The fully qualified type name.

## TryParse(string?, out ResourceKey)

Tries to parse a string into a [ResourceKey](#).

```
public static bool TryParse(string? s, out ResourceKey result)
```

## Parameters

s [string](#)

The string to parse

result [ResourceKey](#)

When this method returns, contains the result of successfully parsing s or an undefined value on failure

## Returns

[bool](#)

## TryParse(string?, IFormatProvider?, out ResourceKey)

Tries to parse a string into a [ResourceKey](#).

```
public static bool TryParse(string? s, IFormatProvider? provider, out ResourceKey result)
```

## Parameters

s [string](#)

The string to parse

provider [IFormatProvider](#)

An object that provides culture-specific formatting information about s

result [ResourceKey](#)

When this method returns, contains the result of successfully parsing s or an undefined value on failure

## Returns

[bool](#)

# Operators

## operator ==(ResourceKey, ResourceKey)

```
public static bool operator ==(ResourceKey left, ResourceKey right)
```

### Parameters

left [ResourceKey](#)

right [ResourceKey](#)

### Returns

[bool](#)

## implicit operator ResourceKey(string)

Converts a string into a [ResourceKey](#).

```
public static implicit operator ResourceKey(string s)
```

### Parameters

s [string](#)

The string to convert

### Returns

[ResourceKey](#)

## operator !=(ResourceKey, ResourceKey)

```
public static bool operator !=(ResourceKey left, ResourceKey right)
```

Parameters

**left** [ResourceKey](#)

**right** [ResourceKey](#)

Returns

[bool](#) ↗

# Enum ResourceKeyOrder

Namespace: [LlamaLogic.Packages](#)

Assembly: LlamaLogic.Packages.dll

Specifies the order in which a [ResourceKey](#) sequence should be processed

```
public enum ResourceKeyOrder
```

## Fields

**InstanceTypeGroup = 1**

The operation will process the [ResourceKey](#) sequence in the order [FullInstance](#), then [Type](#), then [Group](#)

**Preserve = 0**

The operation will process the [ResourceKey](#) sequence in the order in which it was originally encountered

**TypeGroupInstance = 2**

The operation will process the [ResourceKey](#) sequence in the order [Type](#), then [Group](#), then [FullInstance](#)

# Class ResourceToolingMetadataAttribute

Namespace: [LlamaLogic.Packages](#)

Assembly: LlamaLogic.Packages.dll

Specifies that the [ResourceType](#) is tooling metadata and not intended for use by the game

```
[AttributeUsage(AttributeTargets.Field, AllowMultiple = false)]
public sealed class ResourceToolingMetadataAttribute : Attribute
```

## Inheritance

[object](#) ← [Attribute](#) ← ResourceToolingMetadataAttribute

## Inherited Members

[Attribute.Equals\(object\)](#) , [Attribute.GetCustomAttribute\(Assembly, Type\)](#) ,  
[Attribute.GetCustomAttribute\(Assembly, Type, bool\)](#) ,  
[Attribute.GetCustomAttribute\(MemberInfo, Type\)](#) ,  
[Attribute.GetCustomAttribute\(MemberInfo, Type, bool\)](#) ,  
[Attribute.GetCustomAttribute\(Module, Type\)](#) , [Attribute.GetCustomAttribute\(Module, Type, bool\)](#) ,  
[Attribute.GetCustomAttribute\(ParameterInfo, Type\)](#) ,  
[Attribute.GetCustomAttribute\(ParameterInfo, Type, bool\)](#) , [Attribute.GetCustomAttributes\(Assembly\)](#) ,  
[Attribute.GetCustomAttributes\(Assembly, bool\)](#) , [Attribute.GetCustomAttributes\(Assembly, Type\)](#) ,  
[Attribute.GetCustomAttributes\(Assembly, Type, bool\)](#) , [Attribute.GetCustomAttributes\(MemberInfo\)](#) ,  
[Attribute.GetCustomAttributes\(MemberInfo, bool\)](#) ,  
[Attribute.GetCustomAttributes\(MemberInfo, Type\)](#) ,  
[Attribute.GetCustomAttributes\(MemberInfo, Type, bool\)](#) , [Attribute.GetCustomAttributes\(Module\)](#) ,  
[Attribute.GetCustomAttributes\(Module, bool\)](#) , [Attribute.GetCustomAttributes\(Module, Type\)](#) ,  
[Attribute.GetCustomAttributes\(Module, Type, bool\)](#) , [Attribute.GetCustomAttributes\(ParameterInfo\)](#) ,  
[Attribute.GetCustomAttributes\(ParameterInfo, bool\)](#) ,  
[Attribute.GetCustomAttributes\(ParameterInfo, Type\)](#) ,  
[Attribute.GetCustomAttributes\(ParameterInfo, Type, bool\)](#) , [Attribute.GetHashCode\(\)](#) ,  
[Attribute.IsDefaultAttribute\(\)](#) , [Attribute.IsDefined\(Assembly, Type\)](#) ,  
[Attribute.IsDefined\(Assembly, Type, bool\)](#) , [Attribute.IsDefined\(MemberInfo, Type\)](#) ,  
[Attribute.IsDefined\(MemberInfo, Type, bool\)](#) , [Attribute.IsDefined\(Module, Type\)](#) ,  
[Attribute.IsDefined\(Module, Type, bool\)](#) , [Attribute.IsDefined\(ParameterInfo, Type\)](#) ,  
[Attribute.IsDefined\(ParameterInfo, Type, bool\)](#) , [Attribute.Match\(object\)](#) , [Attribute.TypeId](#) ,  
[object.Equals\(object, object\)](#) , [object.GetType\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

# Enum ResourceType

Namespace: [LlamaLogic.Packages](#)

Assembly: LlamaLogic.Packages.dll

A type of resource within a [DataBasePackedFile](#)

```
public enum ResourceType : uint
```

## Fields

AVI = 929579223

AVI

[ResourceFileType(ResourceFileType.TuningMarkup)] AccountRewardTuning = 3138329048

Account Reward Tuning

[ResourceFileType(ResourceFileType.TuningMarkup)] AchievementCategoryTuning = 609337601

Achievement Category Tuning

[ResourceFileType(ResourceFileType.TuningMarkup)] AchievementCollectionTuning = 80917605

Achievement Collection Tuning

[ResourceFileType(ResourceFileType.TuningMarkup)] AchievementTuning = 2018877086

Achievement Tuning

[ResourceFileType(ResourceFileType.TuningMarkup)] ActionTuning = 209137191

Action Tuning

AgeGenderMap = 17805169

Age Gender Map

[ResourceFileType(ResourceFileType.TuningMarkup)] Ambience = 3537656749

Ambience

`AnimationBoundaryConditionCache = 479834948`

Animation Boundary Condition Cache

`AnimationConstraintCache = 3794048034`

Animation Constraint Cache

`[ResourceFileType(ResourceFileType.TuningMarkup)] AnimationStateMachine = 47570707`

Animation State Machine

`[ResourceFileType(ResourceFileType.TuningMarkup)] AnimationTuning = 3994535597`

Animation Tuning

`[ResourceFileType(ResourceFileType.PortableNetworkGraphic)] ApartmentThumbnail1 = 2870590650`

Apartment Thumbnail 1

`[ResourceFileType(ResourceFileType.PortableNetworkGraphic)] ApartmentThumbnail2 = 3175683878`

Apartment Thumbnail 2

`[ResourceFileType(ResourceFileType.TuningMarkup)] AspirationCategoryTuning = 3813727192`

Aspiration Category Tuning

`[ResourceFileType(ResourceFileType.TuningMarkup)] AspirationTrackTuning = 3223387309`

Aspiration Track Tuning

`[ResourceFileType(ResourceFileType.TuningMarkup)] AspirationTuning = 683034229`

Aspiration Tuning

`[ResourceFileType(ResourceFileType.Binary)] AudioConfiguration = 4244956094`

Audio Configuration

`AudioConfigurationReference = 968010314`

## Audio Configuration Reference

**AudioEffects** = 32437818

Audio Effects

**AudioVocals** = 27600859

Audio Vocals

**AutomationTuning** = 3671155384

Automation Tuning

[ResourceFileType(ResourceFileType.TuningMarkup)] **AwayActionTuning** = 2947394632

Away Action Tuning

[ResourceFileType(ResourceFileType.TuningMarkup)] **BalloonTuning** = 3966406598

Balloon Tuning

**BatchFixHistory** = 1810979774

Batch Fix History

[ResourceFileType(ResourceFileType.Binary)] **BlendGeometry** = 108833297

Blend Geometry

**Block** = 127102176

Block

**Blueprint** = 958717478

Blueprint

[ResourceFileType(ResourceFileType.PortableNetworkGraphic)] **BlueprintImage** = 3543935006

Blueprint Image

[ResourceFileType(ResourceFileType.PortableNetworkGraphic)] **BodyPartThumbnail** = 1529359685

Body Part Thumbnail

BonePose = 55959718

Bone Pose

[ResourceFileType(ResourceFileType.TuningMarkup)] BreedTuning = 874331941

Breed Tuning

[ResourceFileType(ResourceFileType.TuningMarkup)] BroadcasterTuning = 3736796019

Broadcaster Tuning

[ResourceFileType(ResourceFileType.TuningMarkup)] BucksPerkTuning = 3963461902

Bucks Perk Tuning

[ResourceFileType(ResourceFileType.TuningMarkup)] BuffTuning = 1612179606

Buff Tuning

[ResourceFileType(ResourceFileType.TuningMarkup)] BuildBuyTuning = 2919482169

Build Buy Tuning

[ResourceFileType(ResourceFileType.TuningMarkup)] BusinessRuleTuning = 3102051436

Business Rule Tuning

[ResourceFileType(ResourceFileType.TuningMarkup)] BusinessTuning = 1977092083

Business Tuning

[ResourceFileType(ResourceFileType.PortableNetworkGraphic)] BuyBuildThumbnail = 1009419847

Buy Build Thumbnail

[ResourceFileType(ResourceFileType.TuningMarkup)] CASCameraTuning = 626712290

CAS Camera Tuning

[ResourceFileType(ResourceFileType.TuningMarkup)] CASLightingTuning = 2148152976

CAS Lighting Tuning

[ResourceFileType(ResourceFileType.TuningMarkup)] CASMenuItemTuning = 213537012

CAS Menu Item Tuning

[ResourceFileType(ResourceFileType.TuningMarkup)] CASMenuTuning = 2472182722

CAS Menu Tuning

[ResourceFileType(ResourceFileType.TuningMarkup)] CASModifierTuning = 4088135484

CAS Modifier Tuning

[ResourceFileType(ResourceFileType.TuningMarkup)] CASOccultSkintoneTuning = 2076404814

CAS Occult Skintone Tuning

[ResourceFileType(ResourceFileType.TuningMarkup)] CASOccultTuning = 563571177

CAS Occult Tuning

[ResourceFileType(ResourceFileType.Binary)] CASPart = 55242443

CAS Part

[ResourceFileType(ResourceFileType.PortableNetworkGraphic)] CASPartThumbnail = 1008398834

CAS Part Thumbnail

[ResourceFileType(ResourceFileType.TuningMarkup)] CASPreferenceCategoryTuning = 3456433227

CAS Preference Category Tuning

[ResourceFileType(ResourceFileType.TuningMarkup)] CASPreferenceItemTuning = 3966303522

CAS Preference Item Tuning

[ResourceFileType(ResourceFileType.Binary)] CASPreset = 3936561885

CAS Preset

[ResourceFileType(ResourceFileType.TuningMarkup)] CASStroriesAnswerTuning = 2163289367

CAS Stories Answer Tuning

```
[ResourceFileType(ResourceFileType.TuningMarkup)] CASStroriesQuestionTuning =  
52718493
```

CAS Stories Question Tuning

```
[ResourceFileType(ResourceFileType.TuningMarkup)] CASStroriesTraitChooserTuning =  
2376930633
```

CAS Stories Trait Chooser Tuning

```
[ResourceFileType(ResourceFileType.TuningMarkup)] CASThumbnailPartTuning =  
2932986028
```

CAS Thumbnail Part Tuning

```
[ResourceFileType(ResourceFileType.TuningMarkup)] CASTuning = 3796587143
```

CAS Tuning

```
CMRF = 1700995767
```

CMRF

```
[ResourceFileType(ResourceFileType.TuningMarkup)] CallToActionTuning = 4114068192
```

Call To Action Tuning

```
[ResourceFileType(ResourceFileType.TuningMarkup)] CameraTuning = 306800208
```

Camera Tuning

```
[ResourceFileType(ResourceFileType.TuningMarkup)] CareerEventTuning = 2487354146
```

Career Event Tuning

```
[ResourceFileType(ResourceFileType.TuningMarkup)] CareerGigTuning = 3436908253
```

Career Gig Tuning

```
[ResourceFileType(ResourceFileType.TuningMarkup)] CareerLevelTuning = 745582072
```

Career Level Tuning

```
[ResourceFileType(ResourceFileType.TuningMarkup)] CareerTrackTuning = 1221024995
```

Career Track Tuning

[ResourceFileType(ResourceFileType.TuningMarkup)] CareerTuning = 1939434475

Career Tuning

[ResourceFileType(ResourceFileType.TuningMarkup)] ClanTuning = 3737052837

Clan Tuning

[ResourceFileType(ResourceFileType.TuningMarkup)] ClanValueTuning = 2576273579

Clan Value Tuning

[ResourceFileType(ResourceFileType.TuningMarkup)] ClientTutorialTuning = 645072749

Client Tutorial Tuning

[ResourceFileType(ResourceFileType.Binary)] Clip = 1797309683

Clip

ClipHeader = 3158986820

Clip Header

[ResourceFileType(ResourceFileType.TuningMarkup)] ClubInteractionGroupTuning = 4195351092

Club Interaction Group Tuning

[ResourceFileType(ResourceFileType.TuningMarkup)] ClubSeedTuning = 794407991

Club Seed Tuning

ColorBlendedTerrain = 2934464740

Color Blended Terrain

ColorList = 2810271350

Color List

ColorTimelineData = 4250380503

Color Timeline Data

Column = 493744591

Column

[ResourceFileType(ResourceFileType.Binary)] CombinedTuning = 1659456824

Combined Binary Tuning

[ResourceFileType(ResourceFileType.TuningMarkup)] ConditionalLayerTuning = 2441338001

Conditional Layer Tuning

[ResourceFileType(ResourceFileType.TuningMarkup)] Credits = 1721014748

Credits

Cursor = 647463969

Cursor

[ResourceFileType(ResourceFileType.Binary)] CutoutInfoTable = 2177505808

Cutout Info Table

[ResourceFileType(ResourceFileType.DirectDrawSurface)] DSTImage = 11720834

DST Image

DecoTrim = 332336850

Deco Trim

[ResourceFileType(ResourceFileType.Binary)] DeformerMap = 3678658665

Deformer Map

[ResourceFileType(ResourceFileType.TuningMarkup)] DetectiveClueTuning = 1400130038

Detective Clue Tuning

[ResourceFileType(ResourceFileType.TuningMarkup)] DevelopmentalMilestoneTuning = 3307360148

Developmental Milestone Tuning

[ResourceFileType(ResourceFileType.TuningMarkup)] DramaNodeTuning = 626258997

Drama Node Tuning

[ResourceFileType(ResourceFileType.TuningMarkup)] EnsembleTuning = 3112702240

Ensemble Tuning

Fence = 68746794

Fence

Floor = 3036111561

Floor

FloorTrim = 2227319321

Floor Trim

[ResourceFileType(ResourceFileType.TuningMarkup)] FontConfiguration = 53690476

Font Configuration

[ResourceFileType(ResourceFileType.Binary)] Footprint = 3548561239

Footprint

Footprint3 = 3073696847

Footprint 3

Foundation = 799971390

Foundation

FountainTrim = 3886917533

Fountain Trim

Frieze = 2690089244

Frieze

[ResourceFileType(ResourceFileType.TuningMarkup)] GameRulesetTuning = 3779558936

Game Ruleset Tuning

GenericMTX = 2885921078

Generic MTX

[ResourceFileType(ResourceFileType.Binary)] Geometry = 22681673

Geometry

[ResourceFileType(ResourceFileType.TuningMarkup)] GuidanceTipTuning = 3567295165

Guidance Tip Tuning

HalfWall = 2438063804

HalfWall

[ResourceFileType(ResourceFileType.TuningMarkup)] HeadlineTuning = 4093714525

Headline Tuning

[ResourceFileType(ResourceFileType.TuningMarkup)] HolidayDefinitionTuning = 238120813

Holiday Definition Tuning

[ResourceFileType(ResourceFileType.TuningMarkup)] HolidayTraditionTuning = 1070408838

Holiday Tradition Tuning

HouseholdDescription = 1923050575

Household Description

[ResourceFileType(ResourceFileType.TuningMarkup)] HouseholdMilestoneTuning = 963831539

Household Milestone Tuning

HouseholdTemplate = 3015981296

Household Template

[ResourceFileType(ResourceFileType.TuningMarkup)] HsvTweakerSettingsTuning = 603754750

Hsv Tweaker Settings Tuning

[ResourceFileType(ResourceFileType.TuningMarkup)] InteractionTuning = 3900887599

Interaction Tuning

LRLEImage = 734023391

LRLE Image

Ladder = 3423141965

Ladder

Light = 62178845

Light

[ResourceFileType(ResourceFileType.TuningMarkup)] LocomotionBuilder = 87703163

Locomotion Builder

[ResourceFileType(ResourceFileType.TuningMarkup)] LocomotionConfig = 2600355829

Locomotion Config

[ResourceFileType(ResourceFileType.TuningMarkup)] LotDecorationPresetTuning = 3726571771

Lot Decoration Preset Tuning

[ResourceFileType(ResourceFileType.TuningMarkup)] LotDecorationTuning = 4264407467

Lot Decoration Tuning

[ResourceFileType(ResourceFileType.Binary)] LotDescription = 26488364

Lot Description

LotFootprintReference = 3221768598

Lot Footprint Reference

[ResourceFileType(ResourceFileType.PortableNetworkGraphic)] LotPreviewThumbnail = 221481530

Lot Preview Thumbnail

[ResourceFileType(ResourceFileType.TuningMarkup)] LotTuning = 3632270694

Lot Tuning

[ResourceFileType(ResourceFileType.TuningMarkup)] LotTypeEventMap = 305120874

## Lot Type Event Map

[ResourceFileType(ResourceFileType.TuningMarkup)] LunarCycleTuning = 1430862616

Lunar Cycle Tuning

MTXCatalog = 1845434982

MTX Catalog

Magalog = 3105757403

Magalog

MagazineCollection = 1946487583

Magazine Collection

MaterialDefinition = 30467933

Material Definition

MaterialSet = 33659250

Material Set

MaxisWorldPipeline1 = 3773657385

Maxis World Pipeline 1

MaxisWorldPipeline2 = 4196775902

Maxis World Pipeline 2

[ResourceFileType(ResourceFileType.TuningMarkup)] ModalMusicMapping = 444925637

Modal Music Mapping

Model = 23466547

Model

ModelCutout = 123169303

Model Cutout

ModelLOD = 30478132

Model LOD

ModularPiece = 2568481485

Modular Piece

ModularPieceCatalog = 2688883901

Modular Piece Catalog

[ResourceFileType(ResourceFileType.TuningMarkup)] MoodTuning = 3128647864

Mood Tuning

MusicData = 3254962032

Music Data

NameMap = 23462796

Name Map

[ResourceFileType(ResourceFileType.TuningMarkup)] NarrativeTuning = 1047870521

Narrative Tuning

[ResourceFileType(ResourceFileType.TuningMarkup)] NativeBuildBuyTuning = 445966772

Native Build Buy Tuning

[ResourceFileType(ResourceFileType.TuningMarkup)] NativeSeasonsWeatherTuning = 3023723217

Native Seasons Weather Tuning

[ResourceFileType(ResourceFileType.TuningMarkup)] NotebookEntryTuning = 2567109238

Notebook Entry Tuning

[ResourceFileType(ResourceFileType.Binary)] ObjectCatalog = 832458525

Object Catalog

ObjectCatalogSet = 4283826444

Object Catalog Set

[ResourceFileType(ResourceFileType.Binary)] ObjectDefinition = 3235601127

Object Definition

[ResourceFileType(ResourceFileType.TuningMarkup)] ObjectModifiers = 3794908120

Object Modifiers

[ResourceFileType(ResourceFileType.TuningMarkup)] ObjectPartTuning = 1900520272

Object Part Tuning

[ResourceFileType(ResourceFileType.TuningMarkup)] ObjectStateTuning = 1526890910

Object State Tuning

[ResourceFileType(ResourceFileType.TuningMarkup)] ObjectTuning = 3055412916

Object Tuning

[ResourceFileType(ResourceFileType.TuningMarkup)] ObjectiveTuning = 6899006

Objective Tuning

[ResourceFileType(ResourceFileType.TuningMarkup)] OpenStreetDirectorTuning = 1265622724

Open Street Director Tuning

OpenTypeFont = 628714954

OpenType Font

[ResourceFileType(ResourceFileType.PortableNetworkGraphic)] PNG = 18461549

PNG

[ResourceFileType(ResourceFileType.PortableNetworkGraphic)] PNGImage = 796721156

PNG Image

Path = 975082574

Path

[ResourceFileType(ResourceFileType.PortableNetworkGraphic)]

PetBreedCoatPatternThumbnail = 2389771869

Pet Breed Coat Pattern Thumbnail

PetCoatBrush = 2642310488

Pet Coat Brush

PetCoatPattern = 3302993517

Pet Coat Pattern

[ResourceFileType(ResourceFileType.PortableNetworkGraphic)] PetFacePresetThumbnail = 3061216162

Pet Face Preset Thumbnail

PetPeltLayer = 649036600

Pet Pelt Layer

[ResourceFileType(ResourceFileType.TuningMarkup)] PieMenuCategoryTuning = 65657188

Pie Menu Category Tuning

Platform = 2579526276

Platform

Pond = 1342387004

Pond

PoolTrim = 2782919923

Pool Trim

[ResourceFileType(ResourceFileType.TuningMarkup)] PostureTuning = 2909789983

Posture Tuning

[ResourceFileType(ResourceFileType.TuningMarkup)] PubertyTuning = 1209600077

Puberty Tuning

QueryableWorldMaskManifest = 1316074726

Queryable World Mask Manifest

[ResourceFileType(ResourceFileType.DirectDrawSurface)] RLE2Image = 877907861

RLE 2 Image

[ResourceFileType(ResourceFileType.DirectDrawSurface)] RLESImage = 3129306232

RLES Image

[ResourceFileType(ResourceFileType.TuningMarkup)] RabbitHoleTuning = 2976568058

Rabbit Hole Tuning

Railing = 471658999

Railing

[ResourceFileType(ResourceFileType.TuningMarkup)] RecipeTuning = 3952605219

Recipe Tuning

RecordBundle = 1051371644

Record Bundle

[ResourceFileType(ResourceFileType.Binary)] RegionDescription = 3596464121

Region Description

[ResourceFileType(ResourceFileType.Binary)] RegionMap = 2887187436

Region Map

[ResourceFileType(ResourceFileType.TuningMarkup)] RegionSortTuning = 1062717573

Region Sort Tuning

[ResourceFileType(ResourceFileType.TuningMarkup)] RegionTuning = 1374134669

Region Tuning

[ResourceFileType(ResourceFileType.TuningMarkup)] RelationshipBitTuning = 151314192

Relationship Bit Tuning

[ResourceFileType(ResourceFileType.TuningMarkup)] RelationshipLockTuning = 2922702451

Relationship Lock Tuning

[ResourceFileType(ResourceFileType.TuningMarkup)] RendererCensorTuning = 3788962514

Renderer Censor Tuning

[ResourceFileType(ResourceFileType.TuningMarkup)] RendererFadeTuning = 69178842

Renderer Fade Tuning

[ResourceFileType(ResourceFileType.TuningMarkup)] RendererGlobalDOFSettingsTuning = 2368970974

Renderer Global DOF Settings Tuning

[ResourceFileType(ResourceFileType.TuningMarkup)]  
RendererGlobalHighlightSettingsTuning = 3162386069

Renderer Global Highlight Settings Tuning

[ResourceFileType(ResourceFileType.TuningMarkup)] RendererGlobalLightSettingsTuning = 1790074981

Renderer Global Light Settings Tuning

[ResourceFileType(ResourceFileType.TuningMarkup)] RendererGlobalSSAOSettingsTuning = 795796239

Renderer Global SSAO Settings Tuning

[ResourceFileType(ResourceFileType.TuningMarkup)] RendererGlobalShadowSettingsTuning = 1160337457

Renderer Global Shadow Settings Tuning

[ResourceFileType(ResourceFileType.TuningMarkup)]  
RendererGlobalVolLightScatteringSettingsTuning = 3319940574

Renderer Global Vol Light Scattering Settings Tuning

[ResourceFileType(ResourceFileType.TuningMarkup)] RewardTuning = 1873057832

Reward Tuning

Rig = 2393838558

Rig

RoadDefinition = 2422433294

Road Definition

[ResourceFileType(ResourceFileType.TuningMarkup)] RoleStateTuning = 239932923

Role State Tuning

Roof = 2448276798

Roof

RoofPattern = 4058889606

Roof Pattern

RoofTrim = 2956008719

Roof Trim

[ResourceFileType(ResourceFileType.TuningMarkup)] RoyaltyTuning = 938421991

Royalty Tuning

S4sMergedPackageManifest = 2142678410

Sims 4 Studio Merged Package Manifest

ScaleFormGFX = 1659684250

ScaleForm GFX

Sculpt = 2635774068

Sculpt

[ResourceFileType(ResourceFileType.TuningMarkup)] SeasonTuning = 3381515358

Season Tuning

[ResourceFileType(ResourceFileType.TuningMarkup)] ServiceNPCTuning = 2629964386

Service NPC Tuning

[ResourceFileType(ResourceFileType.Binary)] ShaderPrecomp = 670978917

## Game Shaders for Windows

[ResourceFileType(ResourceFileType.TuningMarkup)] SicknessTuning = 3288062174

Sickness Tuning

[ResourceFileType(ResourceFileType.Binary)] SimData = 1415235194

Sim Data

[ResourceFileType(ResourceFileType.PortableNetworkGraphic)]

SimFeaturedOutfitThumbnail = 3449676359

Sim Featured Outfit Thumbnail

[ResourceFileType(ResourceFileType.TuningMarkup)] SimFilterTuning = 1846401695

Sim Filter Tuning

SimHotspotControl = 2333671278

Sim Hotspot Control

[ResourceFileType(ResourceFileType.PortableNetworkGraphic)] SimHouseholdThumbnail =

1003770887

Sim Household Thumbnail

SimInfo = 39769844

Sim Info

[ResourceFileType(ResourceFileType.TuningMarkup)] SimInfoFixupTuning = 3797424274

Sim Info Fixup Tuning

[ResourceFileType(ResourceFileType.TuningMarkup)] SimLightingTuning = 2648399834

Sim Lighting Tuning

SimModifier = 3321263678

Sim Modifier

SimPreset = 273810874

Sim Preset

```
[ResourceFileType(ResourceFileType.PortableNetworkGraphic)] SimPresetThumbnail = 2626836499
```

Sim Preset Thumbnail

```
[ResourceFileType(ResourceFileType.TuningMarkup)] SimTemplateTuning = 212125579
```

Sim Template Tuning

```
[ResourceFileType(ResourceFileType.TuningMarkup)] SituationGoalSetTuning = 2649944562
```

Situation Goal Set Tuning

```
[ResourceFileType(ResourceFileType.TuningMarkup)] SituationGoalTuning = 1502554343
```

Situation Goal Tuning

```
[ResourceFileType(ResourceFileType.TuningMarkup)] SituationJobTuning = 2617738591
```

Situation Job Tuning

```
[ResourceFileType(ResourceFileType.TuningMarkup)] SituationTuning = 4223905515
```

Situation Tuning

```
[ResourceFileType(ResourceFileType.Binary)] Skintone = 55867754
```

Skintone

```
[ResourceFileType(ResourceFileType.Binary)] SkyBoxTextureData = 1906592201
```

Sky Box Texture Data

```
[ResourceFileType(ResourceFileType.Binary)] Slot = 3540272417
```

Slot

```
[ResourceFileType(ResourceFileType.TuningMarkup)] SlotTypeSetTuning = 1058419973
```

Slot Type Set Tuning

```
[ResourceFileType(ResourceFileType.TuningMarkup)] SlotTypeTuning = 1772477092
```

Slot Type Tuning

```
[ResourceFileType(ResourceFileType.TuningMarkup)] SnippetTuning = 2113017500
```

Snippet Tuning

[ResourceFileType(ResourceFileType.TuningMarkup)] SocialGroupTuning = 776446212

Social Group Tuning

[ResourceFileType(ResourceFileType.TuningMarkup)] SoundMixProperties = 1091959253

Sound Mix Properties

[ResourceFileType(ResourceFileType.TuningMarkup)] SoundModifierMapping = 2776023783

Sound Modifier Mapping

[ResourceFileType(ResourceFileType.TuningMarkup)] SoundProperties = 455450660

Sound Properties

Spandrel = 1057772186

Spandrel

Spawner = 1220708729

Spawner

[ResourceFileType(ResourceFileType.TuningMarkup)] SpellTuning = 523506649

Spell Tuning

Stairs = 2585840924

Stairs

[ResourceFileType(ResourceFileType.TuningMarkup)] StaticCommodityTuning = 1359443523

Static Commodity Tuning

[ResourceFileType(ResourceFileType.TuningMarkup)] StatisticTuning = 865846717

Statistic Tuning

[ResourceFileType(ResourceFileType.TuningMarkup)] StoryArcTuning = 1613438381

Story Arc Tuning

[ResourceFileType(ResourceFileType.TuningMarkup)] StoryChapterTuning = 1250314810

Story Chapter Tuning

[ResourceFileType(ResourceFileType.TuningMarkup)] StrategySetTuning = 1646578134

Strategy Set Tuning

[ResourceFileType(ResourceFileType.TuningMarkup)] StreetTuning = 4142189312

Street Tuning

[ResourceFileType(ResourceFileType.Binary)] StringTable = 570775514

String Table

Style = 2673671952

Style

StyledLook = 1908258978

Styled Look

[ResourceFileType(ResourceFileType.TuningMarkup)] SubrootTuning = 3086978965

Subroot Tuning

SyncPointSchema = 757559827

Sync Point Schema

[ResourceFileType(ResourceFileType.TuningMarkup)] TagCategoriesMetadataTuning = 470996056

Tag Categories Metadata Tuning

[ResourceFileType(ResourceFileType.TuningMarkup)] TagSetTuning = 1228493570

Tag Set Tuning

[ResourceFileType(ResourceFileType.TuningMarkup)] TagTraitGroupMetadataTuning = 2302558876

Tag Trait Group Metadata Tuning

[ResourceFileType(ResourceFileType.TuningMarkup)] TagTuning = 3708124620

Tag Tuning

```
[ResourceFileType(ResourceFileType.TuningMarkup)] TagsMetadataTuning = 2646182397
```

Tags Metadata Tuning

```
[ResourceFileType(ResourceFileType.TuningMarkup)]
```

```
TelemetryMemoryUsageTelemetrySettingsTuning = 2312527733
```

Telemetry Memory Usage Telemetry Settings Tuning

```
[ResourceFileType(ResourceFileType.TuningMarkup)] TelemetryTuning = 2900743302
```

Telemetry Tuning

```
[ResourceFileType(ResourceFileType.TuningMarkup)] TemplateChooserTuning =
```

```
1220728301
```

Template Chooser Tuning

```
[ResourceFileType(ResourceFileType.Binary)] TerrainBlendMap = 1032205008
```

Terrain Blend Map

```
[ResourceFileType(ResourceFileType.Binary)] TerrainData = 2422433293
```

Terrain Data

```
[ResourceFileType(ResourceFileType.PortableNetworkGraphic)] TerrainHeightMap =
```

```
718378482
```

Terrain Height Map

```
[ResourceFileType(ResourceFileType.Binary)] TerrainKDTree = 54209382
```

Terrain KD Tree

```
[ResourceFileType(ResourceFileType.Binary)] TerrainMesh = 2922985887
```

Terrain Mesh

```
TerrainPaint = 3955994988
```

Terrain Paint

```
TerrainSizeInfo = 2422361115
```

Terrain Size Info

`TerrainTool = 338149641`

Terrain Tool

`[ResourceFileType(ResourceFileType.TuningMarkup)] TestBasedScoreTuning = 1332976878`

Test Based Score Tuning

`TheSimsResourceWorkshopProprietary = 20281636`

A proprietary resource used by The Sims Resource Workshop

`[ResourceFileType(ResourceFileType.TuningMarkup)] ThriftStoreTuning = 3841838852`

Thrift Store Tuning

`[ResourceFileType(ResourceFileType.Binary)] ThumbnailCache = 3107625237`

Thumbnail Cache

`ThumbnailExtra1 = 382364612`

Thumbnail Extra 1

`ThumbnailExtra2 = 2953939989`

Thumbnail Extra 2

`[ResourceFileType(ResourceFileType.TuningMarkup)] ThumbnailTuning = 3787619543`

Thumbnail Tuning

`TimelineEvents = 111944783`

Timeline Events

`[ResourceFileType(ResourceFileType.TuningMarkup)] TopicTuning = 1938713686`

Topic Tuning

`TrackMask = 53633251`

Track Mask

`[ResourceFileType(ResourceFileType.TuningMarkup)] TraitTuning = 3412057543`

Trait Tuning

TrayItem = 713711138

Tray Item

[ResourceFileType(ResourceFileType.Binary)] Trim = 1992095756

Trim

TrueTypeFont = 661431481

TrueType Font

[ResourceFileType(ResourceFileType.TuningMarkup)] Tuning = 62078431

Tuning

[ResourceFileType(ResourceFileType.TuningMarkup)] TutorialTipTuning = 2410930353

Tutorial Tip Tuning

[ResourceFileType(ResourceFileType.TuningMarkup)] TutorialTuning = 3762955427

Tutorial Tuning

UIControlEventMap = 3185058337

UI Control Event Map

[ResourceFileType(ResourceFileType.TuningMarkup)] UIEventModeMapping = 2581168265

UI Event Mode Mapping

[ResourceFileType(ResourceFileType.TuningMarkup)] UniversityCourseDataTuning = 689745854

University Course Data Tuning

[ResourceFileType(ResourceFileType.TuningMarkup)] UniversityCourseMajorTuning = 660124491

University Course Major Tuning

[ResourceFileType(ResourceFileType.TuningMarkup)] UniversityTuning = 3646477745

University Tuning

**UnknownWorld2** = 398508673

Unknown World 2

**UnknownWorld3** = 2528165143

Unknown World 3

**UnknownWorld5** = 3162565977

Unknown World 5

**Unspecified** = 0

Unspecified

[ResourceFileType(ResourceFileType.TuningMarkup)] **UserInterfaceInfoTuning** = 3099531875

User Interface Info Tuning

[ResourceFileType(ResourceFileType.TuningMarkup)] **VenueTuning** = 3871070174

Venue Tuning

[ResourceFileType(ResourceFileType.TuningMarkup)] **VideoGlobalTuning** = 2441613679

Video Global Tuning

[ResourceFileType(ResourceFileType.TuningMarkup)] **VideoPlaylistTuning** = 1086277067

Video Playlist Tuning

**VisualEffects** = 454631497

Visual Effects

**VisualEffectsInstanceMap** = 454631498

Visual Effects Instance Map

**VisualEffectsMerged** = 3931183280

Visual Effects Merged

[ResourceFileType(ResourceFileType.TuningMarkup)] **VoiceEffect** = 1942696642

Voice Effect

[ResourceFileType(ResourceFileType.TuningMarkup)] VoicePlugin = 3313685243

Voice Plugin

WMRFReference = 551031958

WMRF Reference

[ResourceFileType(ResourceFileType.TuningMarkup)] WalkByTuning = 1070998590

Walk By Tuning

[ResourceFileType(ResourceFileType.TuningMarkup)] Walkstyle = 666901909

Walkstyle

Wall = 3589339425

Wall

WallTrim = 4264756878

Wall Trim

WaterManifest = 386491005

Water Manifest

WaterMask = 1207820220

Water Mask

WaterMaskList = 2763687493

Water Mask List

[ResourceFileType(ResourceFileType.TuningMarkup)] WeatherEventTuning = 1476851130

Weather Event Tuning

[ResourceFileType(ResourceFileType.TuningMarkup)] WeatherForecastTuning = 1233072753

Weather Forecast Tuning

[ResourceFileType(ResourceFileType.TuningMarkup)] WhimTuning = 1956251190

Whim Tuning

WindowSet = 2834806039

Window Set

WorldCameraInfo = 2301381514

World Camera Info

WorldCameraMesh = 4211986434

World Camera Mesh

[ResourceFileType(ResourceFileType.Binary)] WorldConditionalData = 3494881377

World Conditional Data

[ResourceFileType(ResourceFileType.Binary)] WorldData = 2164920365

World Data

[ResourceFileType(ResourceFileType.Binary)] WorldDescription = 2793466443

World Description

WorldFileHeader = 4033034633

World File Header

WorldLandingStrip = 1332898750

World Landing Strip

WorldLightsInfo = 418629235

World Lights Info

WorldLotArchitecture = 311764532

World Lot Architecture

WorldLotObjects = 2438369240

World Lot Objects

WorldLotParameterInfo = 1006173574

World Lot ParameterInfo

WorldManifest = 2026421476

World Manifest

[ResourceFileType(ResourceFileType.Binary)] WorldMap = 482361971

World Map

[ResourceFileType(ResourceFileType.Binary)] WorldObjectData = 4239499748

World Object Data

WorldOffLotMesh = 170032079

World Off Lot Mesh

WorldRoadPolys = 1866496362

World Road Polys

[ResourceFileType(ResourceFileType.Binary)] WorldTimelineColor = 422580512

World Timeline Color

WorldVisualEffectsInfo = 1541576451

World Visual Effects Info

WorldWaterManifest = 356327961

World Water Manifest

[ResourceFileType(ResourceFileType.PortableNetworkGraphic)] WorldmapLotThumbnail = 2717855684

Worldmap Lot Thumbnail

[ResourceFileType(ResourceFileType.TuningMarkup)] ZoneDirectorTuning = 4183335058

Zone Director Tuning

[ResourceFileType(ResourceFileType.TuningMarkup)] ZoneModifierTuning = 1008568217

## Zone Modifier Tuning

# Class SmartSimUtilities

Namespace: [LlamaLogic.Packages](#)

Assembly: LlamaLogic.Packages.dll

Provides utility methods specific to The Sims 4

```
public static class SmartSimUtilities
```

## Inheritance

[object](#) ← SmartSimUtilities

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

## Methods

### GetResourceKeyForLocale(ResourceKey, CultureInfo)

Gets the [ResourceKey](#) for this string table resource with the specified [locale](#)

```
public static ResourceKey GetResourceKeyForLocale(this ResourceKey key, CultureInfo locale)
```

#### Parameters

key [ResourceKey](#)

locale [CultureInfo](#)

#### Returns

[ResourceKey](#)

#### Exceptions

[ArgumentOutOfRangeException](#)

The locale is not supported by The Sims 4

## GetStringTableLocale(ResourceKey)

Gets the locale of a string table resource in The Sims 4 based on its [key](#)

```
public static CultureInfo GetStringTableLocale(this ResourceKey key)
```

Parameters

[key](#) [ResourceKey](#)

Returns

[CultureInfo](#)

Exceptions

[ArgumentOutOfRangeException](#)

The high order byte of the instance does not correspond to a locale supported by The Sims 4

## GetStringTableResourceKey(CultureInfo, uint, ulong)

Gets the [ResourceKey](#) of a string table resource in The Sims 4 based on its [locale](#), [group](#), and [fullInstance](#)

```
public static ResourceKey GetStringTableResourceKey(this CultureInfo locale, uint group,  
ulong fullInstance)
```

Parameters

[locale](#) [CultureInfo](#)

[group](#) [uint](#)

[fullInstance](#) [ulong](#)

Returns

[ResourceKey](#)

## Exceptions

### [ArgumentOutOfRangeException](#)

The locale is not supported by The Sims 4

# Namespace LlamaLogic.Packages.Cryptography

## Classes

### [Fnv24](#)

Represents the 24-bit implementation of the Fowler–Noll–Vo hash algorithm

### [Fnv32](#)

Represents the 32-bit implementation of the Fowler–Noll–Vo hash algorithm

### [Fnv56](#)

Represents the 56-bit implementation of the Fowler–Noll–Vo hash algorithm

### [Fnv64](#)

Represents the 64-bit implementation of the Fowler–Noll–Vo hash algorithm

### [FnvHash](#)

Represents the base class from which this library's implementations of the Fowler–Noll–Vo hash algorithms derive

# Class Fnv24

Namespace: [LlamaLogic.Packages.Cryptography](#)

Assembly: LlamaLogic.Packages.dll

Represents the 24-bit implementation of the Fowler–Noll–Vo hash algorithm

```
public sealed class Fnv24 : Fnv32, ICryptoTransform, IDisposable
```

## Inheritance

[object](#) ← [HashAlgorithm](#) ← [FnvHash](#) ← [Fnv32](#) ← [Fnv24](#)

## Implements

[ICryptoTransform](#), [IDisposable](#)

## Inherited Members

[Fnv32.HashSize](#), [FnvHash.ComputeHash\(string\)](#), [FnvHash.Initialize\(\)](#), [HashAlgorithm.Clear\(\)](#),  
[HashAlgorithm.ComputeHash\(byte\[\]\)](#), [HashAlgorithm.ComputeHash\(byte\[\], int, int\)](#),  
[HashAlgorithm.ComputeHash\(Stream\)](#),  
[HashAlgorithm.ComputeHashAsync\(Stream, CancellationToken\)](#), [HashAlgorithm.Create\(\)](#),  
[HashAlgorithm.Create\(string\)](#), [HashAlgorithm.Dispose\(\)](#),  
[HashAlgorithm.TransformBlock\(byte\[\], int, int, byte\[\], int\)](#),  
[HashAlgorithm.TransformFinalBlock\(byte\[\], int, int\)](#),  
[HashAlgorithm.TryComputeHash\(ReadOnlySpan<byte>, Span<byte>, out int\)](#),  
[HashAlgorithm.CanReuseTransform](#), [HashAlgorithm.CanTransformMultipleBlocks](#),  
[HashAlgorithm.InputBlockSize](#), [HashAlgorithm.OutputBlockSize](#), [object.Equals\(object\)](#),  
[object.Equals\(object, object\)](#), [object.GetHashCode\(\)](#), [object.GetType\(\)](#),  
[object.ReferenceEquals\(object, object\)](#), [object.ToString\(\)](#)

## Fields

### EmptyStringHash

Gets the hash value for an empty string

```
public static readonly uint EmptyStringHash
```

## Field Value

[uint](#)

## Properties

### Hash

Gets the value of the computed hash code.

```
public override byte[]? Hash { get; }
```

#### Property Value

[byte](#)[]

The current value of the computed hash code.

#### Exceptions

[CryptographicUnexpectedOperationException](#)

[HashValue](#) is [null](#).

[ObjectDisposedException](#)

The object has already been disposed.

## Methods

### GetHash(string?)

Computes the hash value for the specified string

```
public static uint GetHash(string? text)
```

#### Parameters

[text](#) [string](#)

#### Returns

[uint](#)

## SetHighBit(uint)

Sets the high bit of a 24-bit Fowler–Noll–Vo hash

```
public static uint SetHighBit(uint hash)
```

Parameters

hash [uint](#)

Returns

[uint](#)

# Class Fnv32

Namespace: [LlamaLogic.Packages.Cryptography](#)

Assembly: LlamaLogic.Packages.dll

Represents the 32-bit implementation of the Fowler–Noll–Vo hash algorithm

```
public class Fnv32 : FnvHash, ICryptoTransform, IDisposable
```

## Inheritance

[object](#) ← [HashAlgorithm](#) ← [FnvHash](#) ← Fnv32

## Implements

[ICryptoTransform](#), [IDisposable](#)

## Derived

[Fnv24](#)

## Inherited Members

[FnvHash.hash](#), [FnvHash.ComputeHash\(string\)](#), [FnvHash.HashCore\(byte\[\], int, int\)](#), [FnvHash.HashFinal\(\)](#),  
[FnvHash.Initialize\(\)](#), [HashAlgorithm.HashSizeValue](#), [HashAlgorithm.HashValue](#),  
[HashAlgorithm.State](#), [HashAlgorithm.Clear\(\)](#), [HashAlgorithm.ComputeHash\(byte\[\]\)](#),  
[HashAlgorithm.ComputeHash\(byte\[\], int, int\)](#), [HashAlgorithm.ComputeHash\(Stream\)](#),  
[HashAlgorithm.ComputeHashAsync\(Stream, CancellationToken\)](#), [HashAlgorithm.Create\(\)](#),  
[HashAlgorithm.Create\(string\)](#), [HashAlgorithm.Dispose\(\)](#), [HashAlgorithm.Dispose\(bool\)](#),  
[HashAlgorithm.HashCore\(ReadOnlySpan<byte>\)](#),  
[HashAlgorithm.TransformBlock\(byte\[\], int, int, byte\[\], int\)](#),  
[HashAlgorithm.TransformFinalBlock\(byte\[\], int, int\)](#),  
[HashAlgorithm.TryComputeHash\(ReadOnlySpan<byte>, Span<byte>, out int\)](#),  
[HashAlgorithm.TryHashFinal\(Span<byte>, out int\)](#), [HashAlgorithm.CanReuseTransform](#),  
[HashAlgorithm.CanTransformMultipleBlocks](#), [HashAlgorithm.InputBlockSize](#),  
[HashAlgorithm.OutputBlockSize](#), [object.Equals\(object\)](#), [object.Equals\(object, object\)](#),  
[object.GetHashCode\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#),  
[object.ReferenceEquals\(object, object\)](#), [object.ToString\(\)](#)

# Constructors

[Fnv32\(\)](#)

Initializes a new instance of the [Fnv32](#) class

```
public Fnv32()
```

## Fields

### EmptyStringHash

Gets the hash value for an empty string

```
public static readonly uint EmptyStringHash
```

Field Value

[uint](#)

## Properties

### Hash

Gets the value of the computed hash code.

```
public override byte[]? Hash { get; }
```

Property Value

[byte](#)[]

The current value of the computed hash code.

### Exceptions

[CryptographicUnexpectedOperationException](#)

[HashValue](#) is [null](#).

[ObjectDisposedException](#)

The object has already been disposed.

## HashSize

Gets the size, in bits, of the computed hash code.

```
public override int HashSize { get; }
```

Property Value

[int](#)

The size, in bits, of the computed hash code.

## Methods

### GetHash(string?)

Computes the hash value for the specified string

```
public static uint GetHash(string? text)
```

Parameters

[text](#) [string](#)

Returns

[uint](#)

### SetHighBit(uint)

Sets the high bit of a 32-bit Fowler–Noll–Vo hash

```
public static uint SetHighBit(uint hash)
```

## Parameters

**hash** [uint](#) ↗

## Returns

[uint](#) ↗

# Class Fnv56

Namespace: [LlamaLogic.Packages.Cryptography](#)

Assembly: LlamaLogic.Packages.dll

Represents the 56-bit implementation of the Fowler–Noll–Vo hash algorithm

```
public sealed class Fnv56 : Fnv64, ICryptoTransform, IDisposable
```

## Inheritance

[object](#) ← [HashAlgorithm](#) ← [FnvHash](#) ← [Fnv64](#) ← [Fnv56](#)

## Implements

[ICryptoTransform](#), [IDisposable](#)

## Inherited Members

[Fnv64.HashSize](#), [FnvHash.ComputeHash\(string\)](#), [FnvHash.Initialize\(\)](#), [HashAlgorithm.Clear\(\)](#),  
[HashAlgorithm.ComputeHash\(byte\[\]\)](#), [HashAlgorithm.ComputeHash\(byte\[\], int, int\)](#),  
[HashAlgorithm.ComputeHash\(Stream\)](#),  
[HashAlgorithm.ComputeHashAsync\(Stream, CancellationToken\)](#), [HashAlgorithm.Create\(\)](#),  
[HashAlgorithm.Create\(string\)](#), [HashAlgorithm.Dispose\(\)](#),  
[HashAlgorithm.TransformBlock\(byte\[\], int, int, byte\[\], int\)](#),  
[HashAlgorithm.TransformFinalBlock\(byte\[\], int, int\)](#),  
[HashAlgorithm.TryComputeHash\(ReadOnlySpan<byte>, Span<byte>, out int\)](#),  
[HashAlgorithm.CanReuseTransform](#), [HashAlgorithm.CanTransformMultipleBlocks](#),  
[HashAlgorithm.InputBlockSize](#), [HashAlgorithm.OutputBlockSize](#), [object.Equals\(object\)](#),  
[object.Equals\(object, object\)](#), [object.GetHashCode\(\)](#), [object.GetType\(\)](#),  
[object.ReferenceEquals\(object, object\)](#), [object.ToString\(\)](#)

## Fields

### EmptyStringHash

Gets the hash value for an empty string

```
public static readonly ulong EmptyStringHash
```

## Field Value

[ulong](#)

## Properties

### Hash

Gets the value of the computed hash code.

```
public override byte[]? Hash { get; }
```

#### Property Value

[byte](#)[]

The current value of the computed hash code.

#### Exceptions

[CryptographicUnexpectedOperationException](#)

[HashValue](#) is [null](#).

[ObjectDisposedException](#)

The object has already been disposed.

## Methods

### GetHash(string?)

Computes the hash value for the specified string

```
public static ulong GetHash(string? text)
```

#### Parameters

[text](#) [string](#)

#### Returns

[ulong ↗](#)

## SetHighBit(ulong)

Sets the high bit of a 56-bit Fowler–Noll–Vo hash

```
public static ulong SetHighBit(ulong hash)
```

Parameters

hash [ulong ↗](#)

Returns

[ulong ↗](#)

# Class Fnv64

Namespace: [LlamaLogic.Packages.Cryptography](#)

Assembly: LlamaLogic.Packages.dll

Represents the 64-bit implementation of the Fowler–Noll–Vo hash algorithm

```
public class Fnv64 : FnvHash, ICryptoTransform, IDisposable
```

## Inheritance

[object](#) ← [HashAlgorithm](#) ← [FnvHash](#) ← [Fnv64](#)

## Implements

[ICryptoTransform](#), [IDisposable](#)

## Derived

[Fnv56](#)

## Inherited Members

[FnvHash.hash](#), [FnvHash.ComputeHash\(string\)](#), [FnvHash.HashCore\(byte\[\], int, int\)](#), [FnvHash.HashFinal\(\)](#),  
[FnvHash.Initialize\(\)](#), [HashAlgorithm.HashSizeValue](#), [HashAlgorithm.HashValue](#),  
[HashAlgorithm.State](#), [HashAlgorithm.Clear\(\)](#), [HashAlgorithm.ComputeHash\(byte\[\]\)](#),  
[HashAlgorithm.ComputeHash\(byte\[\], int, int\)](#), [HashAlgorithm.ComputeHash\(Stream\)](#),  
[HashAlgorithm.ComputeHashAsync\(Stream, CancellationToken\)](#), [HashAlgorithm.Create\(\)](#),  
[HashAlgorithm.Create\(string\)](#), [HashAlgorithm.Dispose\(\)](#), [HashAlgorithm.Dispose\(bool\)](#),  
[HashAlgorithm.HashCore\(ReadOnlySpan<byte>\)](#),  
[HashAlgorithm.TransformBlock\(byte\[\], int, int, byte\[\], int\)](#),  
[HashAlgorithm.TransformFinalBlock\(byte\[\], int, int\)](#),  
[HashAlgorithm.TryComputeHash\(ReadOnlySpan<byte>, Span<byte>, out int\)](#),  
[HashAlgorithm.TryHashFinal\(Span<byte>, out int\)](#), [HashAlgorithm.CanReuseTransform](#),  
[HashAlgorithm.CanTransformMultipleBlocks](#), [HashAlgorithm.InputBlockSize](#),  
[HashAlgorithm.OutputBlockSize](#), [object.Equals\(object\)](#), [object.Equals\(object, object\)](#),  
[object.GetHashCode\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#),  
[object.ReferenceEquals\(object, object\)](#), [object.ToString\(\)](#)

# Constructors

[Fnv64\(\)](#)

Initializes a new instance of the [Fnv64](#) class

```
public Fnv64()
```

## Fields

### EmptyStringHash

Gets the hash value for an empty string

```
public static readonly ulong EmptyStringHash
```

Field Value

[ulong](#)

## Properties

### Hash

Gets the value of the computed hash code.

```
public override byte[]? Hash { get; }
```

Property Value

[byte](#)[]

The current value of the computed hash code.

### Exceptions

[CryptographicUnexpectedOperationException](#)

[HashValue](#) is [null](#).

[ObjectDisposedException](#)

The object has already been disposed.

## HashSize

Gets the size, in bits, of the computed hash code.

```
public override int HashSize { get; }
```

Property Value

[int](#)

The size, in bits, of the computed hash code.

## Methods

### GetHash(string?)

Computes the hash value for the specified string

```
public static ulong GetHash(string? text)
```

Parameters

[text](#) [string](#)

Returns

[ulong](#)

### SetHighBit(ulong)

Sets the high bit of a 64-bit Fowler–Noll–Vo hash

```
public static ulong SetHighBit(ulong hash)
```

Parameters

**hash** [ulong](#) ↗

Returns

[ulong](#) ↗

# Class FnvHash

Namespace: [LlamaLogic.Packages.Cryptography](#)

Assembly: LlamaLogic.Packages.dll

Represents the base class from which this library's implementations of the Fowler–Noll–Vo hash algorithms derive

```
public abstract class FnvHash : HashAlgorithm, ICryptoTransform, IDisposable
```

## Inheritance

[object](#) ← [HashAlgorithm](#) ← FnvHash

## Implements

[ICryptoTransform](#), [IDisposable](#)

## Derived

[Fnv32](#), [Fnv64](#)

## Inherited Members

[HashAlgorithm.HashSizeValue](#), [HashAlgorithm.HashValue](#), [HashAlgorithm.State](#),  
[HashAlgorithm.Clear\(\)](#), [HashAlgorithm.ComputeHash\(byte\[\]\)](#),  
[HashAlgorithm.ComputeHash\(byte\[\], int, int\)](#), [HashAlgorithm.ComputeHash\(Stream\)](#),  
[HashAlgorithm.ComputeHashAsync\(Stream, CancellationToken\)](#), [HashAlgorithm.Create\(\)](#),  
[HashAlgorithm.Create\(string\)](#), [HashAlgorithm.Dispose\(\)](#), [HashAlgorithm.Dispose\(bool\)](#),  
[HashAlgorithm.HashCore\(ReadOnlySpan<byte>\)](#),  
[HashAlgorithm.TransformBlock\(byte\[\], int, int, byte\[\], int\)](#),  
[HashAlgorithm.TransformFinalBlock\(byte\[\], int, int\)](#),  
[HashAlgorithm.TryComputeHash\(ReadOnlySpan<byte>, Span<byte>, out int\)](#),  
[HashAlgorithm.TryHashFinal\(Span<byte>, out int\)](#), [HashAlgorithm.CanReuseTransform](#),  
[HashAlgorithm.CanTransformMultipleBlocks](#), [HashAlgorithm.Hash](#), [HashAlgorithm.HashSize](#),  
[HashAlgorithm.InputBlockSize](#), [HashAlgorithm.OutputBlockSize](#), [object.Equals\(object\)](#),  
[object.Equals\(object, object\)](#), [object.GetHashCode\(\)](#), [object.GetType\(\)](#),  
[object.MemberwiseClone\(\)](#), [object.ReferenceEquals\(object, object\)](#), [object.ToString\(\)](#)

## Constructors

[FnvHash\(ulong, ulong\)](#)

Represents the base class from which this library's implementations of the Fowler–Noll–Vo hash algorithms derive

```
protected FnvHash(ulong prime, ulong offset)
```

## Parameters

prime [ulong](#)

offset [ulong](#)

## Fields

### hash

The hash aggregate

```
protected ulong hash
```

## Field Value

[ulong](#)

## Methods

### ComputeHash(string?)

Computes the hash value for the specified string

```
public byte[] ComputeHash(string? value)
```

## Parameters

value [string](#)

## Returns

[byte](#)[]

## HashCore(byte[], int, int)

When overridden in a derived class, routes data written to the object into the hash algorithm for computing the hash.

```
protected override void HashCore(byte[] array, int ibStart, int cbSize)
```

### Parameters

**array** [byte](#)[]

The input to compute the hash code for.

**ibStart** [int](#)

The offset into the byte array from which to begin using data.

**cbSize** [int](#)

The number of bytes in the byte array to use as data.

## HashFinal()

When overridden in a derived class, finalizes the hash computation after the last data is processed by the cryptographic hash algorithm.

```
protected override byte[] HashFinal()
```

### Returns

[byte](#)[]

The computed hash code.

## Initialize()

Resets the hash algorithm to its initial state.

```
public override void Initialize()
```

# Namespace LlamaLogic.Packages.Models

## Classes

### [Model](#)

Represents a model for a raw resource

### [StringTableModel](#)

Represents a [StringTable](#) resource (  )

### [UnexpectedFormatDecodingException](#)

Represents an exception thrown by an [IModel<TSelf>](#) when it attempts to decode a resource that is not in the expected format

## Interfaces

### [IModel](#)

Represents a model for a resource

### [IModel<TSelf>](#)

Represents a model for a raw resource

# Interface IModel

Namespace: [LlamaLogic.Packages.Models](#)

Assembly: LlamaLogic.Packages.dll

Represents a model for a resource

```
public interface IModel
```

## Properties

### ResourceName

Gets the name of this resource if it has one

```
string? ResourceName { get; }
```

### Property Value

[string](#)

### SupportedTypes

Gets a list of resource types that this model supports

```
public static abstract ISet<ResourceType> SupportedTypes { get; }
```

### Property Value

[ISet](#)<[ResourceType](#)>

## Methods

### Encode()

Encodes the resource model into raw format (   )

`ReadOnlyMemory<byte> Encode()`

Returns

[ReadOnlyMemory](#) <[byte](#)>

## GetName(Stream)

Gets the name of a resource from its raw data, if it has one (   )

`public static abstract string? GetName(Stream stream)`

Parameters

`stream Stream`

Returns

[string](#)

## GetNameAsync(Stream, CancellationToken)

Gets the name of a resource from its raw data, if it has one, asynchronously (   )

`public static abstract Task<string?> GetNameAsync(Stream stream, CancellationToken cancellationToken = default)`

Parameters

`stream Stream`

`cancellationToken CancellationToken`

Returns

[Task](#) <[string](#)>



# Interface `IModel<TSelf>`

Namespace: [LlamaLogic.Packages.Models](#)

Assembly: LlamaLogic.Packages.dll

Represents a model for a raw resource

```
public interface IModel<TSelf> : IModel where TSelf : IModel<TSelf>
```

Type Parameters

`TSelf`

Inherited Members

[IModel.SupportedTypes](#) , [IModel.GetName\(Stream\)](#) , [IModel.GetNameAsync\(Stream, CancellationToken\)](#) ,  
[IModel.ResourceName](#) , [IModel.Encode\(\)](#)

## Methods

### `Decode(ReadOnlyMemory<byte>)`

Decodes the resource in raw format to produce an operable model (   )

```
public static abstract TSelf Decode(ReadOnlyMemory<byte> data)
```

Parameters

`data` [ReadOnlyMemory<byte>](#)

Returns

`TSelf`

# Class Model

Namespace: [LlamaLogic.Packages.Models](#)

Assembly: LlamaLogic.Packages.dll

Represents a model for a raw resource

```
public abstract class Model : IModel
```

Inheritance

[object](#) ← Model

Implements

[IModel](#)

Derived

[DataModel](#), [ModFileManifestModel](#), [StringTableModel](#)

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

## Properties

### ResourceName

Gets the name of this resource if it has one

```
public abstract string? ResourceName { get; }
```

Property Value

[string](#)

### SupportedTypes

Gets a list of resource types that this model supports

```
public static ISet<ResourceType> SupportedTypes { get; }
```

## Property Value

[ISet](#)< [ResourceType](#)>

## Methods

### Encode()

Encodes the resource model into raw format (   )

```
public abstract ReadOnlyMemory<byte> Encode()
```

## Returns

[ReadOnlyMemory](#)<[byte](#)>

### GetName(ResourceType, Stream)

Gets the name of a resource from its raw data, if it has one

```
public static string? GetName(ResourceType type, Stream stream)
```

## Parameters

**type**  [ResourceType](#)

**stream**  [Stream](#)

## Returns

[string](#)

### GetName(Stream)

Gets the name of a resource from its raw data, if it has one (   )

```
public static string? GetName(Stream stream)
```

Parameters

stream [Stream](#)

Returns

[string](#)

## GetNameAsync(ResourceType, Stream, CancellationToken)

Gets the name of a resource from its raw data, if it has one

```
public static Task<string?> GetNameAsync(ResourceType type, Stream stream, CancellationToken cancellationToken = default)
```

Parameters

type  [ResourceType](#)

stream [Stream](#)

cancellationToken  [CancellationToken](#)

Returns

[Task](#)<[string](#)>

## GetNameAsync(Stream, CancellationToken)

Gets the name of a resource from its raw data, if it has one, asynchronously (   )

```
public static Task<string?> GetNameAsync(Stream stream, CancellationToken cancellationToken = default)
```

## Parameters

stream [Stream](#)

cancellationToken [CancellationToken](#)

## Returns

[Task](#) <[string](#)>

# Class StringTableModel

Namespace: [LlamaLogic.Packages.Models](#)

Assembly: LlamaLogic.Packages.dll

Represents a [StringTable](#) resource (  )

```
public class StringTableModel : Model, IModel<StringTableModel>, IModel
```

## Inheritance

[object](#) ← [Model](#) ← StringTableModel

## Implements

[IModel<StringTableModel>](#), [IModel](#)

## Inherited Members

[Model.GetName\(ResourceType, Stream\)](#) ,  
[Model.GetNameAsync\(ResourceType, Stream, CancellationToken\)](#) , [object.Equals\(object\)](#) ,  
[object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

# Properties

## Count

Gets the number of entries in the string table

```
public int Count { get; }
```

## Property Value

[int](#)

## this[uint]

Gets/sets the string associated with a [keyHash](#)

```
public string this[uint keyHash] { get; set; }
```

## Parameters

keyHash [uint](#)

## Property Value

[string](#)

## KeyHashes

Gets the key hashes in the string table

```
public IReadOnlyList<uint> KeyHashes { get; }
```

## Property Value

[IReadOnlyList](#)<[uint](#)>

## ResourceName

Gets the name of this resource if it has one

```
public override string? ResourceName { get; }
```

## Property Value

[string](#)

## SupportedTypes

Gets a list of resource types that this model supports

```
public static ISet<ResourceType> SupportedTypes { get; }
```

## Property Value

[ISet](#) <ResourceType>

## Methods

### AddNew(string)

Adds `string` to the string table, returning its key hash

```
public uint AddNew(string @string)
```

#### Parameters

`string` [string](#)

#### Returns

[uint](#)

#### Exceptions

[ArgumentException](#)

The key hash generated for `string` already existed in the string table

### Decode(ReadOnlyMemory<byte>)

Decodes the resource in raw format to produce an operable model (   )

```
public static StringTableModel Decode(ReadOnlyMemory<byte> data)
```

#### Parameters

`data` [ReadOnlyMemory](#) <`byte`>

#### Returns

## Delete(uint)

Deletes the string associated with a `keyHash`, returning `true` if the key hash existed in the string table

```
public bool Delete(uint keyHash)
```

Parameters

`keyHash` [uint](#)

Returns

[bool](#)

## Encode()

Encodes the resource model into raw format ( )

```
public override ReadOnlyMemory<byte> Encode()
```

Returns

[ReadOnlyMemory](#) <[byte](#)>

## Get(uint)

Gets the string associated with a `keyHash`

```
public string Get(uint keyHash)
```

Parameters

`keyHash` [uint](#)

Returns

[string](#)

## GetName(Stream)

Gets the name of a resource from its raw data, if it has one ( [🔗](#) )

```
public static string? GetName(Stream stream)
```

Parameters

[stream](#) [Stream](#)

Returns

[string](#)

## GetNameAsync(Stream, CancellationToken)

Gets the name of a resource from its raw data, if it has one, asynchronously ( [🔗](#) )

```
public static Task<string?> GetNameAsync(Stream stream, CancellationToken cancellationToken  
= default)
```

Parameters

[stream](#) [Stream](#)

[cancellationToken](#) [CancellationToken](#)

Returns

[Task](#)<[string](#)>

## Set(uint, string)

Sets the `string` associated with a `keyHash`, returning `true` if the key hash did not already exist in the string table

```
public bool Set(uint keyHash, string @string)
```

Parameters

`keyHash` `uint`

`string` `string`

Returns

`bool`

# Class UnexpectedFormatException

Namespace: [LlamaLogic.Packages.Models](#)

Assembly: LlamaLogic.Packages.dll

Represents an exception thrown by an [IModel<TSelf>](#) when it attempts to decode a resource that is not in the expected format

```
public sealed class UnexpectedFormatException : Exception, ISerializable
```

## Inheritance

[object](#) ← [Exception](#) ← UnexpectedFormatException

## Implements

[ISerializable](#)

## Inherited Members

[Exception.GetBaseException\(\)](#) , [Exception.GetObjectData\(SerializationInfo, StreamingContext\)](#) ,  
[Exception.GetType\(\)](#) , [Exception.ToString\(\)](#) , [Exception.Data](#) , [Exception.HelpLink](#) ,  
[Exception.HResult](#) , [Exception.InnerException](#) , [Exception.Message](#) , [Exception.Source](#) ,  
[Exception.StackTrace](#) , [Exception.TargetSite](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.GetHashCode\(\)](#) , [object.ReferenceEquals\(object, object\)](#)

## Constructors

### UnexpectedFormatException()

Initializes a new [UnexpectedFormatException](#)

```
public UnexpectedFormatException()
```

### UnexpectedFormatException(Exception?)

Initializes a new [UnexpectedFormatException](#) with the specified [innerException](#)

```
public UnexpectedFormatException(Exception? innerException)
```

## Parameters

innerException [Exception](#)

## UnexpectedFormatException(string)

Initializes a new [UnexpectedFormatException](#) with the specified message

```
public UnexpectedFormatException(string message)
```

## Parameters

message [string](#)

## UnexpectedFormatException(string, Exception?)

Initializes a new [UnexpectedFormatException](#) with the specified message and innerException

```
public UnexpectedFormatException(string message, Exception? innerException)
```

## Parameters

message [string](#)

innerException [Exception](#)

# Namespace LlamaLogic.Packages.Models.Data

## Classes

### [DataModel](#)

Represents a [SimData](#) or [CombinedTuning](#) resource (  )

### [DataModelObject](#)

Represents an [OBJECT](#) reference in a [SimData](#) or [CombinedTuning](#) resource (  )

### [DataModelReference](#)

Represents a reference in a [SimData](#) or [CombinedTuning](#) resource (  )

### [DataModelString](#)

Represents a [STRING8](#) or [HASHEDSTRING8](#) reference in a [SimData](#) or [CombinedTuning](#) resource (  )

### [DataModelTable](#)

Represents a table in a [SimData](#) or [CombinedTuning](#) resource (  )

### [DataModelVariant](#)

Represents a [VARIANT](#) reference in a [SimData](#) or [CombinedTuning](#) resource (  )

### [DataModelVector](#)

Represents a [VECTOR](#) reference in a [SimData](#) or [CombinedTuning](#) resource (  )

## Enums

### [DataModelType](#)

A type of data within a data resource

### [DataModelVersion](#)

Known versions of the format for [SimData](#) and [CombinedTuning](#) resources

# Class DataModel

Namespace: [LlamaLogic.Packages.Models.Data](#)

Assembly: LlamaLogic.Packages.dll

Represents a [SimData](#) or [CombinedTuning](#) resource (  )

```
public sealed class DataModel : Model, IModel<DataModel>, IModel
```

## Inheritance

[object](#) ↗ ← [Model](#) ← DataModel

## Implements

[IModel<DataModel>](#), [IModel](#)

## Inherited Members

[Model.GetName\(ResourceType, Stream\)](#) ,  
[Model.GetNameAsync\(ResourceType, Stream, CancellationToken\)](#) , [object.Equals\(object\)](#) ↗ ,  
[object.Equals\(object, object\)](#) ↗ , [object.GetHashCode\(\)](#) ↗ , [object.GetType\(\)](#) ↗ ,  
[object.ReferenceEquals\(object, object\)](#) ↗ , [object.ToString\(\)](#) ↗

## Remarks

### Synchronous API Only

Due to performance considerations, this object model provides no asynchronous API. Thus, the caller will be blocked until the entire [SimData](#) or [CombinedTuning](#) resource is decoded by [Decode\(ReadOnlyMemory<byte>\)](#) or encoded by [Encode\(\)](#). Front-end developers are advised to wrap calls to either method with [Run\(Func<Task>\)](#) to avoid blocking the UI thread. The asynchronous API of [DataBasePackedFile](#) will do this automatically.

### Eager Loading

When a [SimData](#) or [CombinedTuning](#) resource is passed to [Decode\(ReadOnlyMemory<byte>\)](#), the entire resource is decoded to compose a complete graph of all tables in the resource, including their metadata and records. For some [CombinedTuning](#) resources shipped by Maxis, this may result in a large number of tables and records being loaded into memory and take a significant amount of time.

### Thread Safety

For performance reasons, this class and its supporting classes in this namespace are not thread-safe. If a caller has need for parallelized use of a [DataModel](#), they are expected to manage synchronization on

their own.

## Exposed Reference Structure

To grant callers maximum control, the structure of references from the [SimData](#) or [CombinedTuning](#) resource is preserved and unabstracted.

While the classes which implement [DataModelReference](#) have properties to produce useful compositions (e.g. the [Value](#) property of [DataModelString](#) will convert the null-terminated sequence of ASCII characters it references in a [DataModelTable](#) of [CHAR8](#) values to a [string](#)), these conveniences are processed on-demand and only so long as the reference has remained valid. The original data being referenced remains stored by instances of [DataModelTable](#) as it appeared in the resource at the time it was decoded. All instances of [DataModelReference](#) monitor the row (or rows) of the [DataModelTable](#) they are referencing and will invalidate themselves if any constituent element is altered by the caller.

[Encode\(\)](#) reacts to invalid references by writing the [RELOFFSET\\_NULL](#) value in their place. Callers can check to see if something they've done has invalidated a reference by checking its [IsValid](#) property.

## Properties

### this[int]

Gets the table at the specified [index](#)

```
public DataModelTable this[int index] { get; }
```

#### Parameters

[index](#) [int](#)

### Property Value

[DataModelTable](#)

### this[string]

Gets the table with the specified [name](#)

```
public DataModelTable this[string name] { get; }
```

## Parameters

**name** [string](#)

## Property Value

[DataModelTable](#)

## ResourceName

Gets the name of this resource if it has one

```
public override string? ResourceName { get; }
```

## Property Value

[string](#)

## SupportedTypes

Gets a list of resource types that this model supports

```
public static ISet<ResourceType> SupportedTypes { get; }
```

## Property Value

[ISet](#)< [ResourceType](#)>

## Tables

Gets the tables in this resource

```
public Collection<DataModelTable> Tables { get; }
```

## Property Value

[Collection](#)< [DataModelTable](#)>

# Version

Gets/sets the [DataModelVersion](#) of this resource

```
public DataModelVersion Version { get; set; }
```

## Property Value

[DataModelVersion](#)

# Methods

## Decode(ReadOnlyMemory<byte>)

Decodes the resource in raw format to produce an operable model (   )

```
public static DataModel Decode(ReadOnlyMemory<byte> data)
```

## Parameters

**data** [ReadOnlyMemory](#)<[byte](#)>

## Returns

[DataModel](#)

## Exceptions

[UnexpectedFormatDecodingException](#)

The data appears to be a [CombinedTuning](#) resource in XML format

## Encode()

Encodes the resource model into raw format (   )

```
public override ReadOnlyMemory<byte> Encode()
```

Returns

[ReadOnlyMemory](#)<[byte](#)>

## GetName(Stream)

Gets the name of a resource from its raw data, if it has one (   )

```
public static string? GetName(Stream stream)
```

Parameters

[stream](#) [Stream](#)

Returns

[string](#)

## GetNameAsync(Stream, CancellationToken)

Gets the name of a resource from its raw data, if it has one, asynchronously (   )

```
public static Task<string?> GetNameAsync(Stream stream, CancellationToken cancellationToken = default)
```

Parameters

[stream](#) [Stream](#)

[cancellationToken](#) [CancellationToken](#)

Returns

[Task](#)<[string](#)>

## GetTable(string)

Gets the table with the specified `name`

```
public DataModelTable GetTable(string name)
```

Parameters

`name` [string](#)

Returns

[DataModelTable](#)

# Class DataModelObject

Namespace: [LlamaLogic.Packages.Models.Data](#)

Assembly: LlamaLogic.Packages.dll

Represents an [OBJECT](#) reference in a [SimData](#) or [CombinedTuning](#) resource (  )

```
public class DataModelObject : DataModelReference
```

## Inheritance

[object](#) ← [DataModelReference](#) ← DataModelObject

## Derived

[DataModelVariant](#)

## Inherited Members

[DataModelReference.Column](#) , [DataModelReference.ColumnIndex](#) , [DataModelReference.IsValid](#) ,  
[DataModelReference.Row](#) , [DataModelReference.RowIndex](#) , [DataModelReference.Table](#) ,  
[DataModelReference.Type](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

## Constructors

DataModelObject(DataModelTable, Index, Index, DataModelType)

Initializes a new [DataModelObject](#) (   )

```
public DataModelObject(DataModelTable table, Index rowIndex, Index columnIndex,  
DataModelType type)
```

## Parameters

table [DataModelTable](#)

The table containing the object being referenced

## **RowIndex** [Index ↗](#)

The row index within the containing table of the object being referenced

## **columnIndex** [Index ↗](#)

The column index within the containing table of the object being referenced

## **type** [DataModelType](#)

The type of the object being referenced

# Properties

## NullReference

Gets the null reference [DataModelObject](#)

```
public static DataModelObject NullReference { get; }
```

## Property Value

[DataModelObject](#)

## Value

Gets/sets the value referenced by the object ( [z ↗](#) )

```
public object? Value { get; set; }
```

## Property Value

[object ↗](#)

# Class DataModelReference

Namespace: [LlamaLogic.Packages.Models.Data](#)

Assembly: LlamaLogic.Packages.dll

Represents a reference in a [SimData](#) or [CombinedTuning](#) resource (  )

```
public abstract class DataModelReference
```

## Inheritance

[object](#) ← DataModelReference

## Derived

[DataModelObject](#), [DataModelString](#), [DataModelVector](#)

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

## Constructors

DataModelReference(DataModelTable, Index, Index,  
DataModelType)

Initializes a new [DataModelReference](#) (   )

```
protected DataModelReference(DataModelTable table, Index rowIndex, Index columnIndex,  
DataModelType type)
```

## Parameters

**table** [DataModelTable](#)

The table containing the structure being referenced

**rowIndex** [Index](#)

The row index within the containing table of the structure being referenced

**columnIndex** [Index ↗](#)

The column index within the containing table of the structure being referenced

**type** [DataModelType](#)

The type of the structure being referenced

## Properties

### Column

Gets the column index within the table containing the reference of the reference

```
public int Column { get; }
```

### Property Value

[int ↗](#)

### ColumnIndex

Gets the column index within the table containing the reference of the reference

```
public Index ColumnIndex { get; }
```

### Property Value

[Index ↗](#)

### IsValid

Gets whether the reference has remained valid

```
public virtual bool IsValid { get; }
```

### Property Value

[bool](#)

## Row

Gets the row index within the table containing the reference of the reference (or -1 if the row has been removed)

```
public int Row { get; }
```

### Property Value

[int](#)

## RowIndex

Gets the row index within the table containing the reference of the reference (or -1 if the row has been removed)

```
public Index RowIndex { get; }
```

### Property Value

[Index](#)

## Table

Gets the table containing the structure being referenced

```
public DataModelTable Table { get; }
```

### Property Value

[DataModelTable](#)

## Type

Gets the type of the structure being referenced

```
public DataModelType Type { get; }
```

Property Value

[DataModelType](#)

# Class DataModelString

Namespace: [LlamaLogic.Packages.Models.Data](#)

Assembly: LlamaLogic.Packages.dll

Represents a [STRING8](#) or [HASHEDSTRING8](#) reference in a [SimData](#) or [CombinedTuning](#) resource (  )

```
public sealed class DataModelString : DataModelReference
```

## Inheritance

[object](#) ↗ ← [DataModelReference](#) ← DataModelString

## Inherited Members

[DataModelReference.Column](#) , [DataModelReference.ColumnIndex](#) , [DataModelReference.IsValid](#) ,  
[DataModelReference.Row](#) , [DataModelReference.RowIndex](#) , [DataModelReference.Table](#) ,  
[DataModelReference.Type](#) , [object.Equals\(object\)](#) ↗ , [object.Equals\(object, object\)](#) ↗ ,  
[object.GetHashCode\(\)](#) ↗ , [object.GetType\(\)](#) ↗ , [object.ReferenceEquals\(object, object\)](#) ↗ ,  
[object.ToString\(\)](#) ↗

## Constructors

### DataModelString(DataModelTable, Index)

Initializes a new [DataModelString](#) (   )

```
public DataModelString(DataModelTable table, Index rowIndex)
```

## Parameters

### table [DataModelTable](#)

The table, a sub sequence of which, is the null-terminated string of [CHAR8](#) values

### rowIndex [Index](#) ↗

The row index within the containing table of the string being referenced at which the string begins

# Properties

## Length

Gets the length of the string in ASCII characters, including the null terminator

```
public int Length { get; }
```

### Property Value

[int](#)

## NullReference

Gets the null reference [DataModelString](#)

```
public static DataModelString NullReference { get; }
```

### Property Value

[DataModelString](#)

## Range

Gets the range rows of the string

```
public Range Range { get; }
```

### Property Value

[Range](#)

## Value

Gets/sets the value of the string ([zz](#))

```
public string Value { get; set; }
```

Property Value

[string](#) ↗

# Class DataModelTable

Namespace: [LlamaLogic.Packages.Models.Data](#)

Assembly: LlamaLogic.Packages.dll

Represents a table in a [SimData](#) or [CombinedTuning](#) resource (  )

```
public sealed class DataModelTable
```

## Inheritance

[object](#) ← DataModelTable

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

## Constructors

**DataModelTable(string?, string?, uint?)**

Represents a table in a [SimData](#) or [CombinedTuning](#) resource (  )

```
public DataModelTable(string? name, string? schemaName, uint? schemaHash)
```

## Parameters

**name** [string](#)?

**schemaName** [string](#)?

**schemaHash** [uint](#)?

## Properties

### ColumnCount

Gets the number of columns

```
public int ColumnCount { get; }
```

Property Value

[int](#)

## this[Index]

Gets/sets the values of the specified [rowIndex](#)

```
public object? this[Index rowIndex] { get; set; }
```

Parameters

[rowIndex](#) [Index](#)

Property Value

[object](#)

## this[Index, Index]

Gets/sets the value at the specified [rowIndex](#) and [columnIndex](#)

```
public object? this[Index rowIndex, Index columnIndex] { get; set; }
```

Parameters

[rowIndex](#) [Index](#)

[columnIndex](#) [Index](#)

Property Value

[object](#)

## this[Index, string]

Gets/sets the value of the specified `rowIndex` for column with the specified `columnName`

```
public object? this[Index rowIndex, string columnName] { get; set; }
```

### Parameters

`rowIndex` [Index](#)

`columnName` [string](#)

### Property Value

[object](#)

## Name

Gets/sets the name of the table

```
public string? Name { get; set; }
```

### Property Value

[string](#)

## RawValues

Gets the raw values of the table

```
public IEnumerable<object?> RawValues { get; }
```

### Property Value

[IEnumerable](#) <[object](#)>

## RowCount

Gets the number of rows

```
public int RowCount { get; }
```

Property Value

[int](#)

## SchemaHash

Gets/sets the unique identifier for the schema of the table

```
public uint? SchemaHash { get; set; }
```

Property Value

[uint](#)?

## SchemaName

Gets/sets the name of the schema of the table

```
public string? SchemaName { get; set; }
```

Property Value

[string](#)

## Methods

AddColumn(string?, DataModelType, ushort,  
IEnumerable<object?>?)

Adds a column with the specified `name`, `type`, `flags`, and `values`, and returns the index of the column

```
public int AddColumn(string? name, DataModelType type, ushort flags = 0, IEnumerable<object?> values = null)
```

## Parameters

name [string](#)

type [DataModelType](#)

flags [ushort](#)

values [IEnumerable<object>](#)

## Returns

[int](#)

## AddRow(object?)

Adds the specified [row](#) and returns the index of the row

```
public int AddRow(object? row)
```

## Parameters

row [object](#)

## Returns

[int](#)

## Get(Index)

Gets the values of the row at the specified [rowIndex](#)

```
public object? Get(Index rowIndex)
```

## Parameters

`rowIndex` [Index ↗](#)

Returns

[object ↗](#)

## Get(Index, Index)

Gets the value at the specified `rowIndex` and `columnIndex`

```
public object? Get(Index rowIndex, Index columnIndex)
```

Parameters

`rowIndex` [Index ↗](#)

`columnIndex` [Index ↗](#)

Returns

[object ↗](#)

## Get(Index, string)

Gets the value at the specified `rowIndex` for the column with the specified `columnName`

```
public object? Get(Index rowIndex, string columnName)
```

Parameters

`rowIndex` [Index ↗](#)

`columnName` [string ↗](#)

Returns

[object ↗](#)

## GetColumnFlags(Index)

Gets the flags of the column at the specified `index`

```
public ushort GetColumnFlags(Index index)
```

Parameters

`index` [Index](#)

Returns

[ushort](#)

## GetColumnFlags(string)

Gets the flags of the column with the specified `columnName`

```
public ushort GetColumnFlags(string columnName)
```

Parameters

`columnName` [string](#)

Returns

[ushort](#)

## GetColumnIndex(string)

Gets the index of the column with the specified `columnName`, or [null](#) if the column does not exist

```
public int? GetColumnIndex(string columnName)
```

Parameters

`columnName` [string](#)

Returns

[int](#)?

## GetColumnName(Index)

Gets the name of the column at the specified [index](#)

```
public string? GetColumnName(Index index)
```

Parameters

[index](#) [Index](#)

Returns

[string](#)

## GetColumnType(Index)

Gets the [DataModelType](#) of the column at the specified [index](#)

```
public DataModelType GetColumnType(Index index)
```

Parameters

[index](#) [Index](#)

Returns

[DataModelType](#)

## GetColumnType(string)

Gets the [DataModelType](#) of the column with the specified [columnName](#)

```
public DataModelType GetColumnType(string columnName)
```

Parameters

columnName [string](#)

Returns

[DataModelType](#)

## GetColumnValues(Index)

Gets all the values for the column at the specified index

```
public IReadOnlyList<object?> GetColumnValues(Index index)
```

Parameters

index [Index](#)

Returns

[IReadOnlyList](#)<[object](#)>

## GetColumnValues(Index, Index, Predicate<object?>)

Gets the values for the specified columnIndex starting at the specified startingRowIndex for as long as the takeWhile predicate returns true

```
public IReadOnlyList<object?> GetColumnValues(Index columnIndex, Index startingRowIndex,  
Predicate<object?> takeWhile)
```

Parameters

columnIndex [Index](#)

startingRowIndex [Index](#)

`takeWhile` [Predicate](#)<[object](#)>

Returns

[IReadOnlyList](#)<[object](#)>

## GetColumnValues(Index, Range)

Gets the values for the column at the specified `columnIndex` for the specified `rows`

```
public IReadOnlyList<object?> GetColumnValues(Index columnIndex, Range rows)
```

Parameters

`columnIndex` [Index](#)

`rows` [Range](#)

Returns

[IReadOnlyList](#)<[object](#)>

## GetColumnValues(string)

Gets all the values for the column with the specified `columnName`

```
public IReadOnlyList<object?> GetColumnValues(string columnName)
```

Parameters

`columnName` [string](#)

Returns

[IReadOnlyList](#)<[object](#)>

## GetColumnValues(string, Index, Predicate<object?>)

Gets the values for the column with the specified `columnName` starting at the specified `startingRowIndex` for as long as the `takeWhile` predicate returns `true`

```
public IReadOnlyList<object?> GetColumnValues(string columnName, Index startingRowIndex,  
Predicate<object?> takeWhile)
```

Parameters

`columnName` [string](#)

`startingRowIndex` [Index](#)

`takeWhile` [Predicate](#)<[object](#)>

Returns

[IReadOnlyList](#)<[object](#)>

## GetColumnValues(string, Range)

Gets the values for the column with the specified `columnName` for the specified `rows`

```
public IReadOnlyList<object?> GetColumnValues(string columnName, Range rows)
```

Parameters

`columnName` [string](#)

`rows` [Range](#)

Returns

[IReadOnlyList](#)<[object](#)>

## GetRawValue(Index)

Gets the value of a raw table for the specified `row`

```
public object? GetRawValue(Index row)
```

Parameters

row [Index](#)

Returns

[object](#)

## GetRawValues(int, int)

Gets the values of a raw table for the specified number of [rows](#) beginning at the specified [rowIndex](#)

```
public IReadOnlyList<object?> GetRawValues(int rowIndex, int rows)
```

Parameters

rowIndex [int](#)

rows [int](#)

Returns

[IReadOnlyList](#) <[object](#)>

## GetRawValues(Range)

Gets the values of a raw table for the specified [rows](#)

```
public IReadOnlyList<object?> GetRawValues(Range rows)
```

Parameters

rows [Range](#)

Returns

[IReadOnlyList](#)<[object](#)>

## InsertColumn(Index, string?, DataModelType, ushort, IEnumerable<object?>?)

Inserts a column with the specified `name`, `type`, and `flags` at the specified `index`

```
public void InsertColumn(Index index, string? name, DataModelType type, ushort flags = 0,  
IEnumerable<object?>? values = null)
```

### Parameters

`index` [Index](#)

`name` [string](#)

`type` [DataModelType](#)

`flags` [ushort](#)

`values` [IEnumerable](#)<[object](#)>

## InsertRow(Index, object?)

Inserts the specified `row` at the specified `index`

```
public void InsertRow(Index index, object? row)
```

### Parameters

`index` [Index](#)

`row` [object](#)

## RemoveColumn(Index)

Removes the column at the specified `index`

```
public void RemoveColumn(Index index)
```

Parameters

index [Index](#)

## RemoveColumn(string)

Removes the column with the specified `columnName` and returns the index at which the column was removed

```
public int RemoveColumn(string columnName)
```

Parameters

columnName [string](#)

Returns

[int](#)

## RemoveRow(Index)

Removes the row at the specified `index`

```
public void RemoveRow(Index index)
```

Parameters

index [Index](#)

## Set(Index, Index, object?)

Sets the value at the specified `rowIndex` and `columnIndex`

```
public void Set(Index rowIndex, Index columnIndex, object? value)
```

## Parameters

rowIndex [Index ↗](#)

columnIndex [Index ↗](#)

value [object ↗](#)

## Set(Index, object?)

Sets the `row` at the specified `index`

```
public void Set(Index index, object? row)
```

## Parameters

index [Index ↗](#)

row [object ↗](#)

## Set(Index, string, object?)

Sets the value of the specified `rowIndex` for column with the specified `columnName`

```
public void Set(Index rowIndex, string columnName, object? value)
```

## Parameters

rowIndex [Index ↗](#)

columnName [string ↗](#)

value [object ↗](#)

## SetColumnFlags(Index, ushort)

Sets the flags of the column at the specified `index`

```
public void SetColumnFlags(Index index, ushort flags)
```

Parameters

`index` [Index](#)

`flags` [ushort](#)

## SetColumnFlags(string, ushort)

Sets the flags of the column with the specified `columnName`

```
public void SetColumnFlags(string columnName, ushort flags)
```

Parameters

`columnName` [string](#)

`flags` [ushort](#)

## SetColumnName(Index, string?)

Sets the name of the column at the specified `index`

```
public void SetColumnName(Index index, string? name)
```

Parameters

`index` [Index](#)

`name` [string](#)

## SetRawValue(Index, object?)

Sets the `value` of a raw table for the specified `row`

```
public void SetRawValue(Index row, object? value)
```

## Parameters

row [Index](#)

value [object](#)

## SetRawValues(Range, IEnumerable<object?>)

Replaces the specified **rows** in a raw table with the specified **values**, returning the length of **values**

```
public int SetRawValues(Range rows, IEnumerable<object?> values)
```

## Parameters

rows [Range](#)

values [IEnumerable](#)<[object](#)>

## Returns

[int](#)

# Enum DataModelType

Namespace: [LlamaLogic.Packages.Models.Data](#)

Assembly: LlamaLogic.Packages.dll

A type of data within a data resource

```
public enum DataModelType
```

## Fields

BOOL = 0

A [bool](#) expressed with an entire [byte](#) (0 is [false](#), any other value is [true](#))

CHAR8 = 1

A single ASCII character

FLOAT = 10

A [float](#)

FLOAT2 = 15

You might think it's a [double](#), but nope: two [float](#)s

FLOAT3 = 16

An obvious trend is developing, for this is three [float](#)s

FLOAT4 = 17

(You know when, every once in a while, there's a great trilogy and someone messes it up?) Here are your four [float](#)s

HASHEDSTRING8 = 12

An [int](#) offset to a null-terminated ASCII [string](#) and its accompanying [Fnv32](#) hash

INT16 = 4

A [short](#)

**INT32** = 6

An [int](#)

**INT64** = 8

A [long](#)

**INT8** = 2

An [sbyte](#)

**LOCKEY** = 20

A four-[byte](#) key used to uniquely identify localized strings in The Sims 4 (typically the [Env32](#) hash of the string in its original language)

**OBJECT** = 13

An [int](#) offset to an object

**RESOURCEKEY** = 19

A [ResourceKey](#) laid out by [FullInstance](#), then [Type](#), then [Group](#)

**STRING8** = 11

An [int](#) offset to a null-terminated ASCII [string](#)

**TABLESETREFERENCE** = 18

It's not just about good manners from "How to Serve and Not Be Served", it's really about this [ulong](#)

Seriously, though, why have these when we already have [UINT64](#), you ask?

you're right that table set refs are just uint64s, but it is important to distinguish them because they mean different things. int64s are just literal numbers, table set refs are tuning IDs. so, when the game loads these, it knows that uint64 is just a number, so leave it alone, whereas a table set ref is tuning, so it needs to look that tuning up and use it as the value there rather than the number itself

— frankk, August 16, 2024 in Lot 51's Discord

**UINT16** = 5

A [ushort](#)

**UINT32** = 7

A [uint](#)

**UINT64** = 9

A [ulong](#)

**UINT8** = 3

A [byte](#)

**UNDEFINED** = 22

?

**VARIANT** = 21

An [int](#) offset to an object accompanied by a [uint](#) type hash

**VECTOR** = 14

An [int](#) offset to an array of objects and a [uint](#) that is its size

# Class DataModelVariant

Namespace: [LlamaLogic.Packages.Models.Data](#)

Assembly: LlamaLogic.Packages.dll

Represents a [VARIANT](#) reference in a [SimData](#) or [CombinedTuning](#) resource (  )

```
public sealed class DataModelVariant : DataModelObject
```

## Inheritance

[object](#)  [DataModelReference](#)  [DataModelObject](#)  DataModelVariant

## Inherited Members

[DataModelObject.NullReference](#) , [DataModelObject.Value](#) , [DataModelReference.Column](#) ,  
[DataModelReference.ColumnIndex](#) , [DataModelReference.IsValid](#) , [DataModelReference.Row](#) ,  
[DataModelReference.RowIndex](#) , [DataModelReference.Table](#) , [DataModelReference.Type](#) ,  
[object.Equals\(object\)](#)  , [object.Equals\(object, object\)](#)  , [object.GetHashCode\(\)](#)  , [object.GetType\(\)](#)  ,  
[object.ReferenceEquals\(object, object\)](#)  , [object.ToString\(\)](#) 

## Constructors

DataModelVariant(DataModelTable, Index, Index,  
DataModelType, uint)

Initializes a new [DataModelVariant](#) (   )

```
public DataModelVariant(DataModelTable table, Index rowIndex, Index columnIndex,  
DataModelType type, uint typeHash)
```

## Parameters

**table** [DataModelTable](#)

The table containing the variant being referenced

**rowIndex** [Index](#) 

The row index within the containing table of the variant being referenced

## columnIndex [Index ↗](#)

The column index within the containing table of the variant being referenced

## type [DataModelType](#)

The type of the variant being referenced

## typeHash [uint](#) ↗

The type hash of the variant

# Properties

## TypeHash

Gets/sets the type hash of the variant

```
public uint TypeHash { get; set; }
```

## Property Value

### [uint](#) ↗

# Methods

## CreateNullReference(uint)

Create a [DataModelVariant](#) null reference

```
public static DataModelVariant CreateNullReference(uint typeHash)
```

## Parameters

### typeHash [uint](#) ↗

The type hash of the variant

## Returns



# Class DataModelVector

Namespace: [LlamaLogic.Packages.Models.Data](#)

Assembly: LlamaLogic.Packages.dll

Represents a [VECTOR](#) reference in a [SimData](#) or [CombinedTuning](#) resource (  )

```
public sealed class DataModelVector : DataModelReference
```

## Inheritance

[object](#) ↗ ← [DataModelReference](#) ← DataModelVector

## Inherited Members

[DataModelReference.Column](#) , [DataModelReference.ColumnIndex](#) , [DataModelReference.Row](#) ,  
[DataModelReference.RowIndex](#) , [DataModelReference.Table](#) , [DataModelReference.Type](#) ,  
[object.Equals\(object\)](#) ↗ , [object.Equals\(object, object\)](#) ↗ , [object.GetHashCode\(\)](#) ↗ , [object.GetType\(\)](#) ↗ ,  
[object.ReferenceEquals\(object, object\)](#) ↗ , [object.ToString\(\)](#) ↗

## Constructors

DataModelVector(DataModelTable, Index, DataModelType, int)

Initializes a new [DataModelVector](#) (   )

```
public DataModelVector(DataModelTable table, Index rowIndex, DataModelType type, int count)
```

## Parameters

**table** [DataModelTable](#)

The table, a sub sequence of which, is the vector

**rowIndex** [Index](#) ↗

The row index within the containing table of the vector being referenced at which the vector begins

**type** [DataModelType](#)

The type of the objects being referenced

**count** [int](#)

The number of elements in the vector

## Properties

### Count

Gets the number of elements in the vector

```
public int Count { get; }
```

Property Value

[int](#)

### Indices

Gets the indices of the vector

```
public IEnumerable<int> Indices { get; }
```

Property Value

[IEnumerable](#) <[int](#)>

### IsValid

Gets whether the reference has remained valid

```
public override bool IsValid { get; }
```

Property Value

[bool](#)

## Range

Gets the range rows of the vector

```
public Range Range { get; }
```

Property Value

[Range](#)

## Values

Gets/sets the sequence of values referenced by the vector ( [zz](#) )

```
public IEnumerable<object?> Values { get; set; }
```

Property Value

[IEnumerable](#) <[object](#)>

## Methods

### CreateNullReference(int)

Create a [DataModelVector](#) null reference

```
public static DataModelVector CreateNullReference(int count = 0)
```

Parameters

**count** [int](#)

The number of elements (probably should be [0](#))

Returns

[DataModelVector](#)

# Enum DataModelVersion

Namespace: [LlamaLogic.Packages.Models.Data](#)

Assembly: LlamaLogic.Packages.dll

Known versions of the format for [SimData](#) and [CombinedTuning](#) resources

```
public enum DataModelVersion : uint
```

## Fields

**SmartSimOriginalEdition = 256**

The edition of the format which shipped with the original release of The Sims 4 on September 2, 2014

**VariantSupport = 257**

The edition of the format which shipped with The Sims 4 (v1.5.139) on February 17, 2015, introducing support for [VARIANT](#) and allowing the creation of [CombinedTuning](#) resources

# Namespace LlamaLogic.Packages.Models.ModFileManifest

## Classes

### [ModFileManifestModel](#)

Represents a mod file manifest [SnippetTuning](#) resource (  )

### [ModFileManifestModelRequiredMod](#)

Represents a required mod in a mod file manifest [SnippetTuning](#) resource (  )

## Enums

### [ModFileManifestResourceHashStrategy](#)

Represents a strategy for generating hashes for package mod files

# Class ModFileManifestModel

Namespace: [LlamaLogic.Packages.Models.ModFileManifest](#)

Assembly: LlamaLogic.Packages.dll

Represents a mod file manifest [SnippetTuning](#) resource (  )

```
public sealed class ModFileManifestModel : Model, IModel<ModFileManifestModel>,  
IModel, IXmlSerializable
```

## Inheritance

[object](#) ↗ ← [Model](#) ← ModFileManifestModel

## Implements

[IModel<ModFileManifestModel>](#), [IModel](#), [IXmlSerializable](#) ↗

## Inherited Members

[Model.GetName\(ResourceType, Stream\)](#) ,  
[Model.GetNameAsync\(ResourceType, Stream, CancellationToken\)](#) , [object.Equals\(object\)](#) ↗ ,  
[object.Equals\(object, object\)](#) ↗ , [object.GetHashCode\(\)](#) ↗ , [object.GetType\(\)](#) ↗ ,  
[object.ReferenceEquals\(object, object\)](#) ↗

## Remarks

This model allows callers to easily create, read, and update a mod file manifest. These manifests are a format sponsored by the Llama Logic team to permit creators to specify the dependency requirements of their mods.

## Use in Mod Package (.package) files

Place one and only one mod file manifest [SnippetTuning](#) resource in the package. If an external application encounters multiple [SnippetTuning](#) resources in a package, it should use the first one (ordered by [Group](#), then by [FullInstance](#)) and ignore the rest.

Avoid using [Group 0x00000000](#) as this is informally reserved for Maxis. It would be a bizarre miracle if they started incorporating dependency information using a community format. Regardless, let's not make them feel unwelcome and hold their spot for them.

## Use in Script Mod Archive (.ts4script) files

Because these files are just ZIP archives with a different extension and their contents are not flatly merged by the game as the contents of package files are, there is no risk of collision. Also, since they

need not contend with game resource management, they need not be in an arcane format like XML, so they are in YAML instead. The static parsing methods of this type expect YAML representations of manifests and the [ToString\(\)](#) method produces them. Name your mod file manifest resource `manifest.yml` and put it in the root of your `.ts4script` archive.

## Properties

### Creators

Gets the names of the creators of the mod

```
[YamlMember(Order = 2, DefaultValuesHandling = DefaultValuesHandling.OmitEmptyCollections)]  
public Collection<string> Creators { get; }
```

### Property Value

[Collection](#)<[string](#)>

### ElectronicArtsPromoCode

Gets/sets the promo code it is suggested the player use during check out in the EA Store if purchasing a pack for use with this mod

```
[YamlMember(Order = 12, DefaultValuesHandling = DefaultValuesHandling.OmitNull)]  
public string? ElectronicArtsPromoCode { get; set; }
```

### Property Value

[string](#)

### Exclusivities

Gets the globally unique names of the exclusivities of this mod, causing it to be incompatible with other mods which share one or more of them

```
[YamlMember(Order = 10, DefaultValuesHandling = DefaultValuesHandling.OmitEmptyCollections)]
```

```
public Collection<string> Exclusivities { get; }
```

## Property Value

[Collection](#)<[string](#)>

## Features

Gets the names of the features unique to this mod which it offers to other mods as a dependency

```
[YamlMember(Order = 9, DefaultValuesHandling = DefaultValuesHandling.OmitEmptyCollections)]  
public Collection<string> Features { get; }
```

## Property Value

[Collection](#)<[string](#)>

## Hash

Gets/sets the hash of the mod file

```
[YamlMember(Order = 5, DefaultValuesHandling = DefaultValuesHandling.OmitDefaults)]  
public ImmutableArray<byte> Hash { get; set; }
```

## Property Value

[ImmutableArray](#)<[byte](#)>

## HashResourceKeys

Gets the resource keys omitted from the [ResourceHashStrategy](#) (this is typically not used)

```
[YamlMember(Order = 7, DefaultValuesHandling = DefaultValuesHandling.OmitEmptyCollections)]  
public HashSet<ResourceKey> HashResourceKeys { get; }
```

## Property Value

[HashSet](#)<[ResourceKey](#)>

## IncompatiblePacks

Gets the list of pack codes identifying the packs incompatible with this mod (e.g. "EP01" for Get to Work)

```
[YamlMember(Order = 13, DefaultValuesHandling = DefaultValuesHandling.OmitEmptyCollections)]  
public Collection<string> IncompatiblePacks { get; }
```

### Property Value

[Collection](#)<[string](#)>

## Name

Gets/sets the name of the mod

```
[YamlMember(Order = 1, DefaultValuesHandling = DefaultValuesHandling.OmitDefaults)]  
public required string Name { get; set; }
```

### Property Value

[string](#)

## RequiredMods

Gets the list of mods required by this mod

```
[YamlMember(Order = 14, DefaultValuesHandling = DefaultValuesHandling.OmitEmptyCollections)]  
public Collection<ModFileManifestModelRequiredMod> RequiredMods { get; }
```

### Property Value

[Collection](#)<[ModFileManifestModelRequiredMod](#)>

## RequiredPacks

Gets the list of pack codes identifying the packs required by this mod (e.g. "EP01" for Get to Work)

```
[YamlMember(Order = 11, DefaultValuesHandling = DefaultValuesHandling.OmitEmptyCollections)]  
public Collection<string> RequiredPacks { get; }
```

### Property Value

[Collection](#)<[string](#)>

## ResourceHashStrategy

Gets/sets the [ModFileManifestResourceHashStrategy](#) to use when producing this mod file's [Hash](#) if it is a package

```
[YamlMember(Order = 6, DefaultValuesHandling = DefaultValuesHandling.OmitDefaults)]  
public ModFileManifestResourceHashStrategy ResourceHashStrategy { get; set; }
```

### Property Value

[ModFileManifestResourceHashStrategy](#)

## ResourceName

Gets the name of this resource if it has one

```
[YamlIgnore]  
public override string? ResourceName { get; }
```

### Property Value

[string](#)

## SubsumedHashes

Gets the hashes of previous versions of this mod in for which I stand even though my hash is different

```
[YamlMember(Order = 8, DefaultValuesHandling = DefaultValuesHandling.OmitEmptyCollections)]  
public HashSet<ImmutableArray<byte>> SubsumedHashes { get; }
```

## Property Value

[HashSet](#)<[ImmutableArray](#)<[byte](#)>>

## SupportedTypes

Gets a list of resource types that this model supports

```
public static ISet<ResourceType> SupportedTypes { get; }
```

## Property Value

[ISet](#)< [ResourceType](#)>

## TuningFullInstance

Gets the [SnippetTuning](#) decimal conversion of the [FullInstance](#) for the mod file manifest

```
[YamlMember(Order = -2, DefaultValuesHandling = DefaultValuesHandling.OmitDefaults)]  
public ulong TuningFullInstance { get; set; }
```

## Property Value

[ulong](#)

## TuningName

Gets the [SnippetTuning](#) name for the mod file manifest

```
[YamlMember(Order = -1, DefaultValuesHandling = DefaultValuesHandling.OmitNull)]  
public string? TuningName { get; set; }
```

## Property Value

[string](#) ↗

## Url

Gets/sets the URL to which players can go to find more information about this mod

```
[YamlMember(Order = 4, DefaultValuesHandling = DefaultValuesHandling.OmitNull)]
public Uri? Url { get; set; }
```

## Property Value

[Uri](#) ↗

## Version

Gets/sets the version of this mod

```
[YamlMember(Order = 3, DefaultValuesHandling = DefaultValuesHandling.OmitNull)]
public string? Version { get; set; }
```

## Property Value

[string](#) ↗

## Methods

### Decode(ReadOnlyMemory<byte>)

Decodes the resource in raw format to produce an operable model (↗ 🚧)

```
public static ModFileManifestModel Decode(ReadOnlyMemory<byte> data)
```

## Parameters

**data** [ReadOnlyMemory](#) ↗ <[byte](#) ↗ >

Returns

[ModFileManifestModel](#)

## Encode()

Encodes the resource model into raw format (   )

```
public override ReadOnlyMemory<byte> Encode()
```

Returns

[ReadOnlyMemory](#) <[byte](#)>

## GetFileSha256Hash(string)

Gets the SHA 256 hash of the content of the file located at `filePath`

```
public static ImmutableArray<byte> GetFileSha256Hash(string filePath)
```

Parameters

`filePath` [string](#)

Returns

[ImmutableArray](#) <[byte](#)>

## GetFileSha256HashAsync(string, CancellationToken)

Gets the SHA 256 hash of the content of the file located at `filePath` asynchronously

```
public static Task<ImmutableArray<byte>> GetFileSha256HashAsync(string filePath,  
CancellationToken cancellationToken = default)
```

Parameters

`filePath string`

`cancellationToken CancellationToken`

Returns

`Task<ImmutableArray<byte>>`

`GetModFileHash(DataBasePackedFile,  
ModFileManifestResourceHashStrategy,  
HashSet<ResourceKey>)`

Gets the hash for the specified `package` using the specified `strategy` and `hashResourceKeys`

```
public static ImmutableArray<byte> GetModFileHash(DataBasePackedFile package,  
ModFileManifestResourceHashStrategy strategy, HashSet<ResourceKey> hashResourceKeys)
```

Parameters

`package DataBasePackedFile`

`strategy ModFileManifestResourceHashStrategy`

`hashResourceKeys HashSet<ResourceKey>`

Returns

`ImmutableArray<byte>`

`GetModFileHash(ZipArchive)`

Gets the hash for the specified `scriptMod`

```
public static ImmutableArray<byte> GetModFileHash(ZipArchive scriptMod)
```

Parameters

scriptMod [ZipArchive](#)

Returns

[ImmutableArray](#) <[byte](#)>

**GetModFileHashAsync(DataBasePackedFile, ModFileManifestResourceHashStrategy, HashSet<ResourceKey>, CancellationToken)**

Gets the hash for the specified `package` using the specified `strategy` and `hashResourceKeys`, asynchronously

```
public static Task<ImmutableArray<byte>> GetModFileHashAsync(DataBasePackedFile package, ModFileManifestResourceHashStrategy strategy, HashSet<ResourceKey> hashResourceKeys, CancellationToken cancellationToken = default)
```

Parameters

`package`  [DataBasePackedFile](#)

`strategy`  [ModFileManifestResourceHashStrategy](#)

`hashResourceKeys`  [HashSet](#) <[ResourceKey](#)>

`cancellationToken`  [CancellationToken](#)

Returns

[Task](#) <[ImmutableArray](#) <[byte](#)>>

**GetModFileManifest(DataBasePackedFile)**

Gets the mod file manifest for the specified `package`, if it has one

```
public static ModFileManifestModel? GetModFileManifest(DataBasePackedFile package)
```

Parameters

package  [DataBasePackedFile](#)

Returns

[ModFileManifestModel](#)

## GetModFileManifest(ZipArchive)

Gets the mod file manifest for the specified `scriptMod`, if it has one

```
public static ModFileManifestModel? GetModFileManifest(ZipArchive scriptMod)
```

Parameters

`scriptMod` [ZipArchive](#)

Returns

[ModFileManifestModel](#)

## GetModFileManifestAndKey(DataBasePackedFile)

Gets the mod file manifest for the specified `package` and its [ResourceKey](#), if the `package` has one

```
public static (ResourceKey, ModFileManifestModel?)  
GetModFileManifestAndKey(DataBasePackedFile package)
```

Parameters

`package`  [DataBasePackedFile](#)

Returns

([ResourceKey](#), [ModFileManifestModel](#))

## GetModFileManifestAndKeyAsync(DataBasePackedFile)

Gets the mod file manifest for the specified `package`, asynchronously, if the `package` has one

```
public static Task<(ResourceKey, ModFileManifestModel?)>
GetModFileManifestAndKeyAsync(DataBasePackedFile package)
```

Parameters

`package`  [DataBasePackedFile](#)

Returns

[Task](#) <(ResourceKey, ModFileManifestModel)>

## GetModFileManifestAsync(DataBasePackedFile)

Gets the mod file manifest for the specified `package`, asynchronously, if it has one

```
public static Task<ModFileManifestModel?> GetModFileManifestAsync(DataBasePackedFile
package)
```

Parameters

`package`  [DataBasePackedFile](#)

Returns

[Task](#) <ModFileManifestModel>

## GetModFileManifestAsync(ZipArchive)

Gets the mod file manifest for the specified `scriptMod`, asynchronously, if it has one

```
public static Task<ModFileManifestModel?> GetModFileManifestAsync(ZipArchive scriptMod)
```

Parameters

`scriptMod`  [ZipArchive](#)

Returns

[Task](#) <[ModFileManifestModel](#)>

## GetModFileManifests(DataBasePackedFile)

Gets the mod file manifests for the specified [package](#) (for use by editors checking for packages which may have been merged by manifest unaware tooling)

```
public static IReadOnlyDictionary<ResourceKey, ModFileManifestModel>
GetModFileManifests(DataBasePackedFile package)
```

Parameters

[package](#)  [DataBasePackedFile](#)

Returns

[IReadOnlyDictionary](#) <[ResourceKey](#), [ModFileManifestModel](#)>

## GetModFileManifestsAsync(DataBasePackedFile)

Gets the mod file manifests for the specified [package](#), asynchronously (for use by editors checking for packages which may have been merged by manifest unaware tooling)

```
public static Task<IReadOnlyDictionary<ResourceKey, ModFileManifestModel>>
GetModFileManifestsAsync(DataBasePackedFile package)
```

Parameters

[package](#)  [DataBasePackedFile](#)

Returns

[Task](#) <[IReadOnlyDictionary](#) <[ResourceKey](#), [ModFileManifestModel](#)>>

## GetName(Stream)

Gets the name of a resource from its raw data, if it has one (   )

```
public static string? GetName(Stream stream)
```

Parameters

stream [Stream](#)

Returns

[string](#)

## GetNameAsync(Stream, CancellationToken)

Gets the name of a resource from its raw data, if it has one, asynchronously (   )

```
public static Task<string?> GetNameAsync(Stream stream, CancellationToken cancellationToken  
= default)
```

Parameters

stream [Stream](#)

cancellationToken [CancellationToken](#)

Returns

[Task](#) <[string](#)>

## Parse(string)

Parses a string into a [ModFileManifestModel](#)

```
public static ModFileManifestModel Parse(string s)
```

Parameters

s [string](#)

The string to parse

Returns

[ModFileManifestModel](#)

Exceptions

[FormatException](#)

`s` is not in the correct format

## Parse(string, IFormatProvider?)

Parses a string into a [ModFileManifestModel](#)

```
public static ModFileManifestModel Parse(string s, IFormatProvider? provider)
```

Parameters

`s` [string](#)

The string to parse

`provider` [IFormatProvider](#)

An object that provides culture-specific formatting information about `s`

Returns

[ModFileManifestModel](#)

Exceptions

[FormatException](#)

`s` is not in the correct format

## ToString()

Generates the YAML representation of the [ModFileManifestModel](#)

```
public override string ToString()
```

Returns

[string](#)

## TryParse(string?, out ModFileManifestModel)

Tries to parse a string into a [ModFileManifestModel](#)

```
public static bool TryParse(string? s, out ModFileManifestModel result)
```

Parameters

[s](#) [string](#)

The string to parse

[result](#) [ModFileManifestModel](#)

When this method returns, contains the result of successfully parsing [s](#) or an undefined value on failure

Returns

[bool](#)

## TryParse(string?, IFormatProvider?, out ModFileManifestModel)

Tries to parse a string into a [ResourceKey](#)

```
public static bool TryParse(string? s, IFormatProvider? provider, out  
ModFileManifestModel result)
```

Parameters

`s` [string](#)

The string to parse

`provider` [IFormatProvider](#)

An object that provides culture-specific formatting information about `s`

`result` [ModFileManifestModel](#)

When this method returns, contains the result of successfully parsing `s` or an undefined value on failure

Returns

[bool](#)

# Class ModFileManifestModelRequiredMod

Namespace: [LlamaLogic.Packages.Models.ModFileManifest](#)

Assembly: LlamaLogic.Packages.dll

Represents a required mod in a mod file manifest [SnippetTuning](#) resource (  )

```
public sealed class ModFileManifestModelRequiredMod : IXmlSerializable
```

## Inheritance

[object](#) ← ModFileManifestModelRequiredMod

## Implements

[IXmlSerializable](#)

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

# Properties

## Creators

Gets the names of the creators of the dependency mod

```
[YamlMember(Order = 2, DefaultValuesHandling = DefaultValuesHandling OMITEmptyCollections)]  
public Collection<string> Creators { get; }
```

## Property Value

[Collection](#)<[string](#)>

## Hashes

Gets the hashes of the mod files that must be present in order for this dependency to be fulfilled

```
[YamlMember(Order = 8)]  
public HashSet<ImmutableArray<byte>> Hashes { get; }
```

## Property Value

[HashSet](#)<[ImmutableArray](#)<[byte](#)>>

## IgnoreIfHashAvailable

Instructs agents to ignore fulfilling this dependency if this hash is present in the player's catalog of mods

```
[YamlMember(Order = 9, DefaultValuesHandling = DefaultValuesHandling.OmitDefaults)]  
public ImmutableArray<byte> IgnoreIfHashAvailable { get; set; }
```

## Property Value

[ImmutableArray](#)<[byte](#)>

## IgnoreIfHashUnavailable

Instructs agents to ignore fulfilling this dependency if this hash is not present in the player's catalog of mods

```
[YamlMember(Order = 10, DefaultValuesHandling = DefaultValuesHandling.OmitDefaults)]  
public ImmutableArray<byte> IgnoreIfHashUnavailable { get; set; }
```

## Property Value

[ImmutableArray](#)<[byte](#)>

## IgnoreIfPackAvailable

Instructs agents to ignore fulfilling this dependency if the pack identified by this pack code is present (e.g. "EP01" for Get to Work)

```
[YamlMember(Order = 11, DefaultValuesHandling = DefaultValuesHandling.OmitNull)]  
public string? IgnoreIfPackAvailable { get; set; }
```

## Property Value

[string](#) ↗

## IgnoreIfPackUnavailable

Instructs agents to ignore fulfilling this dependency if the pack identified by this pack code is not present (e.g. "EP01" for Get to Work)

```
[YamlMember(Order = 12, DefaultValuesHandling = DefaultValuesHandling.OmitNull)]  
public string? IgnoreIfPackUnavailable { get; set; }
```

## Property Value

[string](#) ↗

## ModManifestKey

Gets/sets the [ResourceKey](#) of the mod manifest [SnippetTuning](#) of the dependency mod (optional)

```
[YamlMember(Order = 5, DefaultValuesHandling = DefaultValuesHandling.OmitNull)]  
public ResourceKey? ModManifestKey { get; set; }
```

## Property Value

[ResourceKey](#)?

## Name

Gets/sets the name of the dependency mod

```
[YamlMember(Order = 1)]  
public string Name { get; set; }
```

## Property Value

[string](#)

## RequiredFeatures

Gets the names of the features of the dependency mod which the dependent mod requires

```
[YamlMember(Order = 6, DefaultValuesHandling = DefaultValuesHandling.OmitEmptyCollections)]
public Collection<string> RequiredFeatures { get; }
```

## Property Value

[Collection](#) <[string](#)>

## RequirementIdentifier

Gets/sets an identifier that, when shared with a group of other dependency mods for a single dependent, indicates that only one member of the group need be present for the dependent to be satisfied

```
[YamlMember(Order = 7, DefaultValuesHandling = DefaultValuesHandling.OmitNull)]
public string? RequirementIdentifier { get; set; }
```

## Property Value

[string](#)

## Url

Gets/sets the URL to which players can go to find more information about this dependency mod

```
[YamlMember(Order = 4, DefaultValuesHandling = DefaultValuesHandling.OmitNull)]
public Uri? Url { get; set; }
```

## Property Value

## Version

Gets/sets the version of this dependency mod

```
[YamlMember(Order = 3, DefaultValuesHandling = DefaultValuesHandling.OmitNull)]
public string? Version { get; set; }
```

Property Value

[string](#)

# Enum ModFileManifestResourceHashStrategy

Namespace: [LlamaLogic.Packages.Models.ModFileManifest](#)

Assembly: LlamaLogic.Packages.dll

Represents a strategy for generating hashes for package mod files

```
[Flags]
public enum ModFileManifestResourceHashStrategy
```

## Fields

**Default = 1**

The strategy selected by default by creator manifest tooling

**None = 0**

Inverts the typical behavior of [HashResourceKeys](#), causing it to become a list of keys for the only resources to be used in generating the hash

**PlayerCustomizationToleranceLenient = Default | UseNonTuningSimDataStringTablesAndImages**

The strategy with lenient tolerance for player customization of mod package files, permitting changes to [DirectDrawSurface](#), [PortableNetworkGraphic](#), and [StringTable](#) resources

**PlayerCustomizationToleranceModerate = PlayerCustomizationToleranceLenient | UseStringTables**

The strategy with moderate tolerance for player customization of mod package files, permitting changes to [DirectDrawSurface](#) and [PortableNetworkGraphic](#) resources

**PlayerCustomizationTolerancePermissive = 1**

The strategy most tolerant of player customization of mod package files, permitting any change so long as it does not impact [SimData](#) or [TuningMarkup](#)

**PlayerCustomizationToleranceStrict = PlayerCustomizationToleranceModerate | UseImages**

The strategy least tolerant of player customization of mod package files, permitting no changes of any kind whatever

**UseImages = 8**

[DirectDrawSurface](#) and [PortableNetworkGraphic](#) resources will be used to generate hashes for mod package files

**UseNonTuningSimDataStringTablesAndImages = 2**

Non [DirectDrawSurface](#), [PortableNetworkGraphic](#), [SimData](#), [StringTable](#) and [TuningMarkup](#) resources will be used to generate hashes for mod package files

**UseStringTables = 4**

[StringTable](#) resources will be used to generate hashes for mod package files

**UseTuningAndSimData = 1**

[SimData](#) and [TuningMarkup](#) resources will be used to generate hashes for mod package files