Under Construction. This Readme is a work in progress. Once it's completed, an English version will be provided.

Introduction

Under Construction. This Readme is a work in progress. Once it's completed, an English version will be provided.

Getting Started

Namespace AkiraNetwork.VirtualStorage Library

Filter by title

Classes

- **AkiraNetwork.VirtualStorage Library**

VirtualCycleDetector (AkiraNetwork.VirtualStorageLibrary.VirtualCycleDetector.html)

Class for detecting cycles in virtual symbolic links. Determines if a given link is part of a cycle.

VirtualDirectory (AkiraNetwork.VirtualStorageLibrary.VirtualDirectory.html)

Represents a virtual directory. A virtual directory functions as a container for other nodes. It can contain VirtualItem<T> (AkiraNetwork.VirtualStorageLibrary.VirtualItem-1.html), VirtualDirectory (AkiraNetwork.VirtualStorageLibrary.VirtualDirectory.html), and VirtualSymbolicLink (AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink.html), providing a centralized way to manage these entities.

This class supports operations such as adding, removing, updating, checking the existence of, and enumerating nodes within the directory. It serves as a fundamental component for efficiently managing the structure of the virtual storage.

Additionally, nodes can be enumerated according to the specified display conditions.

VirtualException (AkiraNetwork.VirtualStorageLibrary.VirtualException.html)

The base class for exceptions that occur within the virtual storage library.

VirtualGroupCondition<T, TKey> (AkiraNetwork.VirtualStorageLibrary.VirtualGroupCondition-2.html)

Represents the conditions for grouping data, holding the property to group by and the order (ascending or descending).

VirtualItem (AkiraNetwork.VirtualStorageLibrary.VirtualItem.html)

Represents a virtual item. This class serves as the base class for the VirtualItem<T> (AkiraNetwork.VirtualStorageLibrary.VirtualItem-1.html) type and is used solely for type checking of VirtualItem (AkiraNetwork.VirtualStorageLibrary.VirtualItem.html).

VirtualItem<T> (AkiraNetwork.VirtualStorageLibrary.VirtualItem-1.html)

Represents a virtual item. A virtual item is an object used to manage item data within the virtual storage. It holds item data based on the user-defined type T, facilitating the management and manipulation of various types of content.

ActionNodeDelegate (AkiraNetwork.VirtualStorageLibrary)

IVirtualDeepCloneable<T> (AkiraNetwork.VirtualStorageLibrary.IVirtualDeepCloneable-1.html)

IVirtualWildcardMatcher (AkiraNetwork.VirtualStorageLibrary)

NotifyNodeDelegate (AkiraNetwork.VirtualStorageLibrary)

PatternMatch (AkiraNetwork.VirtualStorageLibrary)

VirtualCycleDetector (AkiraNetwork.VirtualStorageLibrary)

VirtualDirectory (AkiraNetwork.VirtualStorageLibrary)

VirtualException (AkiraNetwork.VirtualStorageLibrary)

VirtualGroupCondition<T, TKey> (AkiraNetwork.VirtualStorageLibrary.VirtualGroupCondition-2.html)

VirtualID (AkiraNetwork.VirtualStorageLibrary)

VirtualItem (AkiraNetwork.VirtualStorageLibrary)

VirtualItem<T> (AkiraNetwork.VirtualStorageLibrary.VirtualItem-1.html)

VirtualNode (AkiraNetwork.VirtualStorageLibrary)

VirtualNodeContext (AkiraNetwork.VirtualStorageLibrary)

VirtualNodeExtensions (AkiraNetwork.VirtualStorageLibrary)

VirtualNodeListConditions (AkiraNetwork.VirtualStorageLibrary)

VirtualNodeName (AkiraNetwork.VirtualStorageLibrary)

VirtualNodeNotFoundException

Virtual items ensure that the actual data is distinctly present. If the user-defined type T implements the IVirtual DeepCloneable<T> (AkiraNetwork.VirtualStorageLibrary.IVirtualDeepCloneable-1.html) interface, a deep clone of the item data is created, ensuring separate instances. If this interface is not implemented, shallow copies may result in multiple virtual items referencing the same instance of the item data.

VirtualNode (AkiraNetwork.VirtualStorageLibrary.VirtualNode.html)

Represents an abstract class for nodes.

- **AkiraNetwork.VirtualStorageLibrary**
  - ActionNodeDelegate (AkiraNetwork.VirtualStorageLibrary.html)
  - IVirtualDeepCloneable<T> (AkiraNetwork.VirtualStorageLibrary-1.html)
  - IVirtualWildcardMatcher (AkiraNetwork.VirtualStorageLibrary.html)
  - NotifyNodeDelegate (AkiraNetwork.VirtualStorageLibrary.html)
  - PatternMatch (AkiraNetwork.VirtualStorageLibrary.html)
  - VirtualCycleDetector (AkiraNetwork.VirtualStorageLibrary.html)
  - VirtualDirectory (AkiraNetwork.VirtualStorageLibrary.html)
  - VirtualException (AkiraNetwork.VirtualStorageLibrary.html)
  - VirtualGroupCondition<T, TKey> (AkiraNetwork.VirtualStorageLibrary-2.html)
  - VirtualID (AkiraNetwork.VirtualStorageLibrary.html)
  - VirtualItem (AkiraNetwork.VirtualStorageLibrary.html)
  - VirtualItem<T> (AkiraNetwork.VirtualStorageLibrary-1.html)
  - VirtualNode (AkiraNetwork.VirtualStorageLibrary.html)
  - VirtualNodeContext (AkiraNetwork.VirtualStorageLibrary.html)
  - VirtualNodeExtensions (AkiraNetwork.VirtualStorageLibrary.html)
  - VirtualNodeListConditions (AkiraNetwork.VirtualStorageLibrary.html)
  - VirtualNodeName (AkiraNetwork.VirtualStorageLibrary.html)
  - VirtualNodeNotFoundException (AkiraNetwork.VirtualStorageLibrary.html)

VirtualNodeContext (AkiraNetwork.VirtualStorageLibrary.VirtualNodeContext.html)

A class that holds context information for a node. It is returned during or after path traversal, providing information about the node, path, parent directory, depth, index, resolved path, and symbolic link.

VirtualNodeExtensions (AkiraNetwork.VirtualStorageLibrary.VirtualNodeExtensions.html)

Provides extension methods for operations on virtual nodes in the Virtual Storage Library.

VirtualNodeName (AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)

Represents the name of a virtual node. This class handles the generation, validation, and comparison of node names.

VirtualNodeNotFoundException (AkiraNetwork.VirtualStorageLibrary.VirtualNodeNotFoundException.html)

The exception that is thrown when a node is not found in the virtual storage.

VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

Represents a path within the virtual storage. Provides functionality for manipulating, comparing, normalizing, and splitting paths.

VirtualSortCondition<T> (AkiraNetwork.VirtualStorageLibrary.VirtualSortCondition-1.html)

Represents the conditions for sorting data, holding the property to sort by and the order (ascending or descending).

VirtualStorageExtensions (AkiraNetwork.VirtualStorageLibrary.VirtualStorageExtensions.html)

Provides extension methods for various operations on collections in the Virtual Storage Library.

VirtualStorageSettings (AkiraNetwork.VirtualStorageLibrary.VirtualStorageSettings.html)

Manages settings for the virtual storage. This class implements the singleton pattern and holds default settings and various parameters.

VirtualStorageState (AkiraNetwork.VirtualStorageLibrary.VirtualStorageState.html)

Manages the state of the virtual storage. This class implements the singleton pattern and maintains the current settings and state.

- AkiraNetwork.VirtualStorageLibrary

VirtualStorage<T> (AkiraNetwork.VirtualStorageLibrary.VirtualStorage-1.html)

Manages virtual storage containing user-defined type T item data.

VirtualSymbolicLink (AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink.html)

Represents a virtual symbolic link. A virtual symbolic link provides a virtual reference to other nodes (directories, items, or other symbolic links). This class represents symbolic links within the virtual storage library and manages paths to other nodes.

This virtual symbolic link supports the concept of NULL links. When the TargetPath is null, the link resolution is not performed during path traversal. This feature can be useful when the target is undefined or dynamically determined.

Structs

VirtualID (AkiraNetwork.VirtualStorageLibrary.VirtualID.html)

A structure representing a unique identifier within the virtual storage library. This identifier is based on a randomly generated UUID version 4 (UUIDv4). UUIDv4 is a 128-bit value that ensures high uniqueness and has strong cryptographic properties.

VirtualNodeListConditions (AkiraNetwork.VirtualStorageLibrary.VirtualNodeListConditions.html)

Represents the conditions for creating a list of virtual nodes.

Interfaces

IVirtualDeepCloneable<T> (AkiraNetwork.VirtualStorageLibrary.IVirtualDeepCloneable-1.html)

Provides functionality to create deep clones of type T.

IVirtualWildcardMatcher (AkiraNetwork.VirtualStorageLibrary.IVirtualWildcardMatcher.html)

Provides functionality for wildcard matching.

ActionNodeDelegate (AkiraNetwork.VirtualStorageLibrary.
VirtualDeepCloneable<T> (AkiraNetwork.VirtualStorageLibrary.1.html)
IVirtualWildcardMatcher (AkiraNetwork.VirtualStorageLibrary.
NotifyNodeDelegate (AkiraNetwork.VirtualStorageLibrary.
PatternMatch (AkiraNetwork.VirtualStorageLibrary.
VirtualCycleDetector (AkiraNetwork.VirtualStorageLibrary.
VirtualDirectory (AkiraNetwork.VirtualStorageLibrary.
VirtualException (AkiraNetwork.VirtualStorageLibrary.
VirtualGroupCondition<T, TKey> (AkiraNetwork.VirtualStorageLibrary.2.html)
VirtualID (AkiraNetwork.VirtualStorageLibrary.
VirtualItem (AkiraNetwork.VirtualStorageLibrary.
VirtualItem<T> (AkiraNetwork.VirtualStorageLibrary.1.html)
VirtualNode (AkiraNetwork.VirtualStorageLibrary.
VirtualNodeContext (AkiraNetwork.VirtualStorageLibrary.
VirtualNodeExtensions (AkiraNetwork.VirtualStorageLibrary.
VirtualNodeMatcher (AkiraNetwork.VirtualStorageLibrary.
VirtualNodeName (AkiraNetwork.VirtualStorageLibrary.
VirtualNodeNotFoundException

Enums

VirtualNodeType (AkiraNetwork.VirtualStorageLibrary.VirtualNodeType.html)

Specifies the types of virtual nodes.

VirtualNodeTypeFilter (AkiraNetwork.VirtualStorageLibrary.VirtualNodeTypeFilter.html)

Specifies the filters for virtual node types.

Delegates

ActionNodeDelegate (AkiraNetwork.VirtualStorageLibrary.ActionNodeDelegate.html)

A delegate used to perform an action on a specific node during node traversal.

NotifyNodeDelegate (AkiraNetwork.VirtualStorageLibrary.NotifyNodeDelegate.html)

A delegate used to notify the status of a specific node during node traversal.

PatternMatch (AkiraNetwork.VirtualStorageLibrary.PatternMatch.html)

A delegate used to determine if a node name matches a pattern.

This documentation uses the DocFX, which is © 2019 Oscar Vásquez.
It is licensed under the MIT License. For more details, please see the
docfx GitHub repository (https://github.com/dotnet/docfx).

This documentation uses the DocFX Material theme, which is © 2019 Oscar Vásquez.
It is licensed under the MIT License. For more details, please see the
DocFX Material GitHub repository (https://github.com/ovasquez/docfx-material).