

# Namespace AkiraNetwork.VirtualStorage Library

▼ Filter by title

## Classes

### - **AkiraNetwork.VirtualStorage**

#### **Library**

#### **VirtualCycleDetector**

(AkiraNetwork.VirtualStorageLibrary.VirtualCycleDetector.html)

ActionNodeDelegate

(AkiraNetwork.VirtualStorageLibrary.

#### **VirtualDirectory (AkiraNetwork.VirtualStorageLibrary.VirtualDirectory.html)**

Represents a virtual directory. A virtual directory functions as a container for other nodes. It can contain Virtual

Item (AkiraNetwork.VirtualStorageLibrary.VirtualItem-1.html), VirtualDirectory

(AkiraNetwork.VirtualStorageLibrary.VirtualDirectory.html), and VirtualSymbolicLink

(AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink.html), providing a centralized way to manage these

entities.

NotifyNodeDelegate

(AkiraNetwork.VirtualStorageLibrary.

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.

VirtualCycleDetector

(AkiraNetwork.VirtualStorageLibrary.

Additional methods are generated according to the specified display conditions.

VirtualDirectory

(AkiraNetwork.VirtualStorageLibrary.

#### **VirtualDirectoryAdapter<T>**

(AkiraNetwork.VirtualStorageLibrary.VirtualDirectoryAdapter-1.html)

1.html)

VirtualException

#### **VirtualException (AkiraNetwork.VirtualStorageLibrary.VirtualException.html)**

(AkiraNetwork.VirtualStorageLibrary.

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

(AkiraNetwork.VirtualStorageLibrary.

2.html)

#### **VirtualGroupCondition<T, TKey>**

(AkiraNetwork.VirtualStorageLibrary.VirtualGroupCondition-2.html)

(AkiraNetwork.VirtualStorageLibrary.

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

descending).

(AkiraNetwork.VirtualStorageLibrary.

VirtualItemAdapter<T>

(AkiraNetwork.VirtualStorageLibrary.

#### **VirtualItem (AkiraNetwork.VirtualStorageLibrary.VirtualItem.html)**

(AkiraNetwork.VirtualStorageLibrary.

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).

VirtualNode

(AkiraNetwork.VirtualStorageLibrary.

#### **VirtualItemAdapter<T> (AkiraNetwork.VirtualStorageLibrary.VirtualItemAdapter-1.html)**

(AkiraNetwork.VirtualStorageLibrary.

VirtualNodeContext

(AkiraNetwork.VirtualStorageLibrary.

VirtualNodeExtensions

## **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.

Virtual items ensure that the actual data is distinctly present. If the user-defined type T implements the IVirtualDeepCloneable<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.

## - **AkiraNetwork.VirtualStorageLibrary**

### **VirtualNode (AkiraNetwork.VirtualStorageLibrary.VirtualStorageLibrary.VirtualNode.html)**

Represents a node delegate class for nodes.

(AkiraNetwork.VirtualStorageLibrary)

IVirtualDeepCloneable<T>

## **VirtualNodeContext**

### **(AkiraNetwork.VirtualStorageLibrary.VirtualNodeContext.html)**

A VirtualNodeContext object contains information for a node. It is returned during or after path traversal, providing information about VirtualStorageLibrary, parent directory, depth, index, resolved path, and symbolic link.

NotifyNodeDelegate

(AkiraNetwork.VirtualStorageLibrary)

## **VirtualNodeExtensions**

### **(AkiraNetwork.VirtualStorageLibrary.VirtualNodeExtensions.html)**

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

(AkiraNetwork.VirtualStorageLibrary)

VirtualDirectory

## **VirtualnodeName (AkiraNetwork.VirtualStorageLibrary.VirtualnodeName.html)**

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

(AkiraNetwork.VirtualStorageLibrary)

1.html

VirtualException

## **VirtualNodeNotFoundException**

### **(AkiraNetwork.VirtualStorageLibrary.VirtualNodeNotFoundException.html)**

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

(AkiraNetwork.VirtualStorageLibrary)

2.html

VirtualID

## **VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)**

VirtualItem

(AkiraNetwork.VirtualStorageLibrary)

## **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).

(AkiraNetwork.VirtualStorageLibrary)

1.html

VirtualStorageExtensions

### **(AkiraNetwork.VirtualStorageLibrary.VirtualStorageExtensions.html)**

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

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeExtensions

## **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

### **- AkiraNetwork.VirtualStorage**

#### **Library**

##### **(AkiraNetwork.VirtualStorage1**

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

ActionNodeDelegate

(AkiraNetwork.VirtualStorageLibrar

VirtualPathGraphLink<T>

##### **VirtualSymbolicLink (AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink.html)**

1.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.

NotifyNodeDelegate

The **AkiraNetwork.VirtualStorage** 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.

VirtualCycleDetector

(AkiraNetwork.VirtualStorageLibrar

##### **VirtualSymbolicLinkAdapter<T>**

##### **(AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLinkAdapter-1.html)**

VirtualDirectoryAdapter<T>

##### **Structs (AkiraNetwork.VirtualStorageLibrary.Structs.html)**

1.html)

VirtualException

##### **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.

VirtualID

(AkiraNetwork.VirtualStorageLibrar

##### **VirtualNodeListConditions**

##### **(AkiraNetwork.VirtualStorageLibrary.VirtualNodeListConditions.html)**

VirtualItemAdapter<T>

Represents the conditions for creating a list of virtual nodes.

(AkiraNetwork.VirtualStorageLibrar

1.html)

##### **Interfaces (AkiraNetwork.VirtualStorageLibrary.Interfaces.html)**

VirtualItem<T>

(AkiraNetwork.VirtualStorageLibrar

1.html)

##### **IVirtualDeepCloneable<T>**

##### **(AkiraNetwork.VirtualStorageLibrary.IVirtualDeepCloneable-1.html)**

Provides functionality to create deep clones of type T.

(AkiraNetwork.VirtualStorageLibrar

VirtualNodeExtensions

## **IVirtualWildcardMatcher**

**(AkiraNetwork.VirtualStorageLibrary.IVirtualWildcardMatcher.html)**

Provides functionality for wildcard matching.

## **Enums**



### **VirtualNodeType (AkiraNetwork.VirtualStorageLibrary.VirtualNodeType.html)**

**AkiraNetwork.VirtualStorageLibrary**  
Specifies the types of virtual nodes.

**(AkiraNetwork.VirtualStorageLibrary)**

### **VirtualNodeTypeFilter**

**(AkiraNetwork.VirtualStorageLibrary.VirtualNodeTypeFilter.html)**

**VirtualDeepCloneable**  
Specifies the filters for virtual node types.

**(AkiraNetwork.VirtualStorageLibrary)**

**1.html**

## **Delegates**

**VirtualWildcardMatcher**

**(AkiraNetwork.VirtualStorageLibrary)**

**NotifyNodeDelegate**

**AkiraNetwork.VirtualStorageLibrary**  
**PatternMatch**

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

**VirtualCycleDetector**

**(AkiraNetwork.VirtualStorageLibrary)**

**VirtualDirectory**

**(AkiraNetwork.VirtualStorageLibrary)**

**VirtualDirectoryAdapter**

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

**(AkiraNetwork.VirtualStorageLibrary)**

**1.html**

### **PatternMatch (AkiraNetwork.VirtualStorageLibrary.PatternMatch.html)**

**VirtualException**

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

**VirtualGroupCondition**<T, TKey>

**(AkiraNetwork.VirtualStorageLibrary)**

**2.html**

**VirtualID**

**(AkiraNetwork.VirtualStorageLibrary)**

**VirtualItem**

**(AkiraNetwork.VirtualStorageLibrary)**

**VirtualItemAdapter**<T>

**(AkiraNetwork.VirtualStorageLibrary)**

**1.html**

**VirtualItem**<T>

**(AkiraNetwork.VirtualStorageLibrary)**

**1.html**

**VirtualNode**

**(AkiraNetwork.VirtualStorageLibrary)**

**VirtualNodeContext**

**(AkiraNetwork.VirtualStorageLibrary)**

**VirtualNodeExtensions**

# Delegate ActionNodeDelegate

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

Filter by title

Namespace: AkiraNetwork (AkiraNetwork.html).VirtualStorageLibrary (AkiraNetwork.VirtualStorageLibrary.html)

Assemblies: AkiraNetwork.dll

Symbols

## (AkiraNetwork.VirtualStorage)

```
public delegate bool ActionNodeDelegate(VirtualDirectory parentDirectory, VirtualNodeName nodeName, VirtualPath nodePath)
(AkiraNetwork.VirtualStorageLibrary)
```

IVirtualDeepCloneable<T>

Parameters

Type	VirtualWildcardMatcher	Name	Description
VirtualDirectory	(AkiraNetwork.VirtualStorageLibrary)	parentDirectory	The parent directory of the target node.
VirtualNodeDelegate	(AkiraNetwork.VirtualStorageLibrary.VirtualDirectory.html)		
VirtualDirectoryName	(AkiraNetwork.VirtualStorageLibrary)	nodeName	The name of the target node.
VirtualPathMatch	(AkiraNetwork.VirtualStorageLibrary.VirtualDirectoryName.html)		
VirtualCycleDetector	(AkiraNetwork.VirtualStorageLibrary)	nodePath	The path of the target node.
VirtualPath	(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)		
VirtualDirectory	(AkiraNetwork.VirtualStorageLibrary)		

Returns

Type	AkiraNetwork.VirtualStorageLibrary	Description
bool	(AkiraNetwork.VirtualStorageLibrary)	Returns true to continue node traversal, or false to stop node traversal.

## Extension Methods

VirtualTextFormatter.GenerateSingleTableDebugText<T>(T)	(AkiraNetwork.VirtualStorageLibrary)
VirtualItem	(AkiraNetwork.VirtualStorageLibrary.Utilities.VirtualTextFormatter.html#AkiraNetwork_VirtualStorageLibrary_Utilities_VirtualTextFormatter_GenerateSingleTableDebugText_1__0_)
VirtualItemAdapter<T>	(AkiraNetwork.VirtualStorageLibrary)
VirtualItem	(AkiraNetwork.VirtualStorageLibrary)
VirtualItem<T>	(AkiraNetwork.VirtualStorageLibrary)
VirtualNode	(AkiraNetwork.VirtualStorageLibrary)
VirtualNodeContext	(AkiraNetwork.VirtualStorageLibrary)
VirtualNodeExtensions	(AkiraNetwork.VirtualStorageLibrary)

# Interface IVirtualDeepCloneable<T>

Provides functionality to create deep clones of type T.

Filter by title

Namespace: AkiraNetwork ([AkiraNetwork.html](#)).VirtualStorageLibrary ([AkiraNetwork.VirtualStorageLibrary.html](#))

Assembly: VirtualStorageLibrary.dll

Syntax ([AkiraNetwork.VirtualStorageLibrary.html](#))

```
public interface IVirtualDeepCloneable<T>
{
    IVirtualDeepCloneable<T>
}
```

Type Parameters ([1.html](#))

Name	Description
T	(AkiraNetwork.VirtualStorageLibrary) The type of the object to clone.

NotifyNodeDelegate ([AkiraNetwork.VirtualStorageLibrary.html](#))

```
(AkiraNetwork.VirtualStorageLibrary)
```

PatternMatch

```
(AkiraNetwork.VirtualStorageLibrary)
```

VirtualCycleDetector

```
(AkiraNetwork.VirtualStorageLibrary)
```

VirtualDirectory

```
(AkiraNetwork.VirtualStorageLibrary)
```

## Methods

DeepClone(bool)

```
(AkiraNetwork.VirtualStorageLibrary)
```

VirtualDirectoryAdapter<T>

```
(AkiraNetwork.VirtualStorageLibrary)
```

## Declaration

VirtualException

```
T(AkiraNetwork.VirtualStorageLibrary) false)
```

VirtualGroupCondition<T, TKey>

```
(AkiraNetwork.VirtualStorageLibrary)
```

## Parameters ([2.html](#))

Type	virtualID	Name	Description
bool		recursive	Indicates whether to recursively clone nested objects. If true, a recursive clone is performed; if false, only the object itself is cloned. This parameter is only relevant for VirtualDirectory, and is ignored for VirtualItem{T} and VirtualSymbolicLink.

[1.html](#))

## Returns [VirtualItem<T>](#)

Type	Description
T	The cloned object.

[\(AkiraNetwork.VirtualStorageLibrary.html\)](#)

VirtualNodeContext

```
(AkiraNetwork.VirtualStorageLibrary)
```

VirtualNodeExtensions

```
(AkiraNetwork.VirtualStorageLibrary)
```

# Extension Methods

VirtualTextFormatter.GenerateSingleTableDebugText<T>(T)  
(AkiraNetwork.VirtualStorageLibrary.Utilities.VirtualTextFormatter.html#AkiraNetwork\_VirtualStorageLibrary\_Utils  
s\_VirtualTextFormatter\_GenerateSingleTableDebugText\_1\_\_0\_)



- **AkiraNetwork.VirtualStorage**  
**Library**  
**(AkiraNetwork.VirtualStorageL**

ActionNodeDelegate  
(AkiraNetwork.VirtualStorageLibrar  
IVirtualDeepCloneable<T>  
(AkiraNetwork.VirtualStorageLibrar  
1.html)  
IVirtualWildcardMatcher  
(AkiraNetwork.VirtualStorageLibrar  
NotifyNodeDelegate  
(AkiraNetwork.VirtualStorageLibrar  
PatternMatch  
(AkiraNetwork.VirtualStorageLibrar  
VirtualCycleDetector  
(AkiraNetwork.VirtualStorageLibrar  
VirtualDirectory  
(AkiraNetwork.VirtualStorageLibrar  
VirtualDirectoryAdapter<T>  
(AkiraNetwork.VirtualStorageLibrar  
1.html)  
VirtualException  
(AkiraNetwork.VirtualStorageLibrar  
VirtualGroupCondition<T, TKey>  
(AkiraNetwork.VirtualStorageLibrar  
2.html)  
VirtualID  
(AkiraNetwork.VirtualStorageLibrar  
VirtualItem  
(AkiraNetwork.VirtualStorageLibrar  
VirtualItemAdapter<T>  
(AkiraNetwork.VirtualStorageLibrar  
1.html)  
VirtualItem<T>  
(AkiraNetwork.VirtualStorageLibrar  
1.html)  
VirtualNode  
(AkiraNetwork.VirtualStorageLibrar  
VirtualNodeContext  
(AkiraNetwork.VirtualStorageLibrar  
VirtualNodeExtensions  
(AkiraNetwork.VirtualStorageLibrar

# Interface IVirtualWildcardMatcher

Provides functionality for wildcard matching.

 Filter by title

Namespace: AkiraNetwork (AkiraNetwork.html).VirtualStorageLibrary (AkiraNetwork.VirtualStorageLibrary.html)

# Library

Assembly: VirtualStorageLibrary.dll  
**(AkiraNetwork.VirtualStorage)**

## Syntax

```
  ActionNodeDelegate  
  (AkiraNetwork.VirtualStorageLibrary  
  public interface IVirtualWildcardMatcher  
  VirtualDepthOneable>
```

(AkiraNetwork.VirtualStorageLibrary  
1.html)

## IVirtualWildcardMatcher

# Properties

## NotifyNodeDelegate (AkiraNetwork.VirtualStorageLibrary) Public Methods

# Count<sup>Patt<sup>#</sup></sup>

(AkiraNetwork.VirtualStorageLibrary)

## Gets Virtual Memory Defragmentation

# (AkiraNetwork.VirtualStorageLibrary) Declaration VirtualDirectory

(AkiraN)

`int VirtGANTDirectoryAdapter<T>  
(AkiraNetwork VirtualStorageLibrary)`

property Value  
VirtualException

## Property Value

## VirtualException

Type	Description
int VirtualGroupCondition<T, TKey> (AkiraNetwork.VirtualStorageLibrary ( <a href="https://learn.microsoft.com/dotnet/api/system.int32_2.html">https://learn.microsoft.com/dotnet/api/system.int32_2.html</a> )	An integer representing the number of wildcards available.

## **Patterns**

(AkiraNetwork.VirtualStorageLibrary)  
Gets a collection of available patterns.  
VirtualItemAdapter<T>

## Declaration

```
1.html)  
VirtualItem<T>  
TEnumerable<string> Patterns {
```

Pattern 18  
(AkiraNetwork.VirtualStorageLibrary)  
1.html)

**Property Value**  
(AkiraNetwork.VirtualStorageLibrary)

## VirtualNodeContext (AkiraNetwork VirtualStorageLibrary)

## VirtualNodeExtensions (AkiraNetwork.VirtualStorageLibrary)

(AkiraNetwork.VirtualStorageLibrary  
VirtualNetwork.GuestList)

Type	Description
IEnumerable ( <a href="https://learn.microsoft.com/dotnet/api/system.collections.generic.ienumerable-1">https://learn.microsoft.com/dotnet/api/system.collections.generic.ienumerable-1</a> ) <string ( <a href="https://learn.microsoft.com/dotnet/api/system.string">https://learn.microsoft.com/dotnet/api/system.string</a> )>	An enumerable collection of strings representing the available patterns.

## Library

# Windows Dictionary Virtual Storage

**ActionNodeDelegate**  
Gets a dictionary of wildcards and their corresponding regex patterns.  
(AkiraNetwork.VirtualStorageLibrary)

## Declaration Virtual DeepCloneable<T>

(AkiraNetwork.VirtualStorageLibrary)

```
1.html) ReadOnlyDictionary<string, string> WildcardDictionary { get; }
```

## IVirtualWildcardMatcher

(AkiraNetwork VirtualStorage) library

## Property Value

Type	Description
AkiraNetwork.VirtualStorageLibrary.PatternMatch AkiraNetwork.VirtualStorageLibrary.ReadOnlyDictionary ( <a href="https://learn.microsoft.com/dotnet/api/system.collections.objectmodel.VirtualCycleDetector.readonlydictionary?view=dotnet-5.0">https://learn.microsoft.com/dotnet/api/system.collections.objectmodel.VirtualCycleDetector.readonlydictionary?view=dotnet-5.0</a> ) <string ( <a href="https://learn.microsoft.com/dotnet/api/system.string">https://learn.microsoft.com/dotnet/api/system.string</a> ), VirtualDirectory string ( <a href="https://learn.microsoft.com/dotnet/api/system.string">https://learn.microsoft.com/dotnet/api/system.string</a> )> (AkiraNetwork.VirtualStorageLibrary)	A read-only dictionary containing wildcards as keys and their corresponding regex patterns as values.

# Wildcards<sup>1.html</sup>

**VirtualException**

Gets (Akita Network Viral) Stowaway Blame

#### VirtualGroupCondition<T, TKey>

VirtualGroupCreation(1, 1Key)  
**Declaration**: AkiraNetwork VirtualStorage library

3.html

```
TEnumerable<string> Wildcards { get; }
```

(AlciraNetwork VirtualStorage) library

(AkitaNet  
VirtualItens)

## VirtualItem

Type	Description
(AkiraNetwork.VirtualStorageLibrary.VirtualItemAdapter<T>)	
IEnumerable<(AkiraNetwork.VirtualStorageLibrary.VirtualItemAdapter<T>)>	An enumerable collection of strings representing the available wildcards.
<string> (https://learn.microsoft.com/dotnet/api/system.string)	

# VirtualNode Methods

**PatternMatcher(string, string)**

Determines whether the specified node name matches the specified pattern.

## Declaration

```
bool PatternMatcher(string nodeName, string pattern)
```



### Parameters

Library	Name	Description
Type <a href="#">(AkiraNetwork.VirtualStorageLibrary)</a>		
string ( <a href="https://learn.microsoft.com/dotnet/api/system.string">https://learn.microsoft.com/dotnet/api/system.string</a> ) ActionNodeDelegate	nodeName	The node name to check.

Type <a href="#">(AkiraNetwork.VirtualStorageLibrary)</a>	pattern	The pattern to compare against.
--	---------	---------------------------------

Returns <a href="#">1.html</a>
-----------------------------------

Type	Description
<a href="#">VirtualWildcardMatcher</a>	True if the node name matches the pattern; otherwise, false.

## Extension Methods

[VirtualDirectory](#).GenerateSingleTableDebugText<T>(T)

([AkiraNetwork.VirtualStorageLibrary.Utilities.VirtualTextFormatter.html#AkiraNetwork\\_VirtualStorageLibrary\\_Utilsilites\\_VirtualTextFormatter\\_GenerateSingleTableDebugText\\_1\\_\\_0\\_](#))

([AkiraNetwork.VirtualStorageLibrary](#)  
1.html)

[VirtualException](#)

([AkiraNetwork.VirtualStorageLibrary](#)  
VirtualGroupCondition<T, TKey>

([AkiraNetwork.VirtualStorageLibrary](#)  
2.html)

[VirtualID](#)

([AkiraNetwork.VirtualStorageLibrary](#)

[VirtualItem](#)

([AkiraNetwork.VirtualStorageLibrary](#)

[VirtualItemAdapter<T>](#)

([AkiraNetwork.VirtualStorageLibrary](#)  
1.html)

[VirtualItem<T>](#)

([AkiraNetwork.VirtualStorageLibrary](#)  
1.html)

[VirtualNode](#)

([AkiraNetwork.VirtualStorageLibrary](#)

[VirtualNodeContext](#)

([AkiraNetwork.VirtualStorageLibrary](#)

[VirtualNodeExtensions](#)

([AkiraNetwork.VirtualStorageLibrary](#)

[VirtualNodeListConditions](#)



# Delegate PatternMatch

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

Filter by title

**Namespace:** AkiraNetwork (AkiraNetwork.html).VirtualStorageLibrary (AkiraNetwork.VirtualStorageLibrary.html)

**Assembly:** VirtualStorageLibrary.dll

(AkiraNetwork.VirtualStorageLibrary)

**Syntax:**

```
VirtualDeepCloneable<T>
(AkiraNetwork.VirtualStorageLibrary)
public delegate bool PatternMatch(string nodeName, string pattern)
1.html)
```

IVirtualWildcardMatcher

(AkiraNetwork.VirtualStorageLibrary)

**Parameters**

NotifyNodeDelegate

Type (AkiraNetwork.VirtualStorageLibrary)

**Name**

**Description**

PatternMatch

string (AkiraNetwork.VirtualStorageLibrary)
(https://learn.microsoft.com/dotnet/api/system.string)

*nodeName*

The name of the node to be matched.

VirtualCycleDetector

string (AkiraNetwork.VirtualStorageLibrary)

(https://learn.microsoft.com/dotnet/api/system.string)

*pattern*

The pattern to match against.

VirtualDirectory

Return VirtualDirectoryAdapter<T>

Type (AkiraNetwork.VirtualStorageLibrary)
1.html)

**Description**

bool VirtualException

(AkiraNetwork.VirtualStorageLibrary)
(https://learn.microsoft.com/dotnet/api/system.boolean)

True if the node name matches the pattern; otherwise, false.

) VirtualGroupCondition<T, TKey>

(AkiraNetwork.VirtualStorageLibrary)

2.html)

## Extension Methods

VirtualID (AkiraNetwork.VirtualStorageLibrary)

VirtualTextFormatter.GenerateSingleTableDebugText<T>(T)

(AkiraNetwork.VirtualStorageLibrary)
(AkiraNetwork.VirtualStorageLibrary.Utilities.VirtualTextFormatter.html#AkiraNetwork\_VirtualStorageLibrary\_Utilsilites\_VirtualTextFormatter\_GenerateSingleTableDebugText\_1\_\_0\_)

VirtualItemAdapter<T> (AkiraNetwork.VirtualStorageLibrary)

(AkiraNetwork.VirtualStorageLibrary)
1.html)

VirtualItem<T>

(AkiraNetwork.VirtualStorageLibrary)

1.html)

VirtualNode

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeContext

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeExtensions

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeListConditions

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeName

# Class VirtualCycleDetector

## Inheritance

↳ [Object](#) (<https://learn.microsoft.com/dotnet/api/system.object>)

↳ [VirtualCycleDetector](#)  
([AkiraNetwork.VirtualStorageLibrary](#))

## Inherited Members

[VirtualDeepCloneable<T>](#)

[Object.Equals](#) ([https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals\(system-object-object\)html](https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals(system-object-object)html))

[Object.Equals\(Object\)](#) ([https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals\(system-object-object\)html](https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals(system-object-object)html))

[Object.GetHashCode](#) (<https://learn.microsoft.com/dotnet/api/system.object.gethashcode>)

[Object.GetType](#) (<https://learn.microsoft.com/dotnet/api/system.object.gettype>)

[Object.MemberwiseClone](#) () (<https://learn.microsoft.com/dotnet/api/system.object.memberwiseclone>)

[Object.ReferenceEquals](#) ([VirtualCycleDetector](#)) (<https://learn.microsoft.com/dotnet/api/system.object.referenceequals>)

[Object.ToString](#) () (<https://learn.microsoft.com/dotnet/api/system.object.tostring>)

([AkiraNetwork.VirtualStorageLibrary](#))

**Namespace:** [AkiraNetwork](#) ([AkiraNetwork.html](#)).VirtualStorageLibrary ([AkiraNetwork.VirtualStorageLibrary.html](#))

VirtualDirectory

**Assembly:** [VirtualStorageLibrary.dll](#)  
([AkiraNetwork.VirtualStorageLibrary](#))

## Syntax

```
VirtualDirectoryAdapter<T>
  (AkiraNetwork.VirtualStorageLibrary)
public class VirtualCycleDetector
  VirtualException
  (AkiraNetwork.VirtualStorageLibrary)
  VirtualGroupCondition<T, TKey>
  (AkiraNetwork.VirtualStorageLibrary)
  2.html)
```

## Constructors

**VirtualCycleDetector()**

([AkiraNetwork.VirtualStorageLibrary](#))

## Declaration

```
VirtualItem
  (AkiraNetwork.VirtualStorageLibrary)
VirtualItemAdapter<T>
  public VirtualCycleDetector()
  (AkiraNetwork.VirtualStorageLibrary)
  1.html)
```

```
VirtualItem<T>
  (AkiraNetwork.VirtualStorageLibrary)
```

## Properties

VirtualNode

([AkiraNetwork.VirtualStorageLibrary](#))

**Count**

VirtualNodeContext

([AkiraNetwork.VirtualStorageLibrary](#))

**Declaration**

```
VirtualNodeExtensions
  (AkiraNetwork.VirtualStorageLibrary)
public int Count { get; }
```

```
VirtualNodeListConditions
```

([AkiraNetwork.VirtualStorageLibrary](#))

**PropertyName**

([AkiraNetwork.VirtualStorageLibrary](#))

Type	Description
int ( <a href="https://learn.microsoft.com/dotnet/api/system.int32">https://learn.microsoft.com/dotnet/api/system.int32</a> )	

## CycleDictionary

### Declaration

```
(AkiraNetwork.VirtualStorageLibrary)
    IVirtualDeepCloneable<T>
    public Dictionary<VirtualID, VirtualSymbolicLink> CycleDictionary { get; }
(AkiraNetwork.VirtualStorageLibrary)
1.html)
```

### Properties

Type	Description
NotifyNodeDelegate	
Dictionary<VirtualID, VirtualSymbolicLink>	<a href="https://learn.microsoft.com/dotnet/api/system.collections.generic.dictionary-2">https://learn.microsoft.com/dotnet/api/system.collections.generic.dictionary-2</a> <VirtualID, VirtualSymbolicLink>
VirtualCycleDetector	<a href="https://learn.microsoft.com/dotnet/api/AkiraNetwork.VirtualStorageLibrary.VirtualCycleDetector">https://learn.microsoft.com/dotnet/api/AkiraNetwork.VirtualStorageLibrary.VirtualCycleDetector</a>

## Methods

### Clear()

VirtualException

Declaration	Description
(AkiraNetwork.VirtualStorageLibrary)	
VirtualGroupCondition<T, TKey>	
(AkiraNetwork.VirtualStorageLibrary)	

### IsNodeInCycle(VirtualSymbolicLink)

Declaration	Description
(AkiraNetwork.VirtualStorageLibrary)	
VirtualItemAdapter<T>	
(AkiraNetwork.VirtualStorageLibrary)	

Parameters	Name	Description
1.html		

Type	Name	Description
VirtualNode		

Parameters	Name	Description
VirtualSymbolicLink ( <a href="https://learn.microsoft.com/dotnet/api/AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink.html">https://learn.microsoft.com/dotnet/api/AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink.html</a> )	link	

Returns	Name	Description
VirtualNodeExtensions		

Type	Description
AkiraNetwork.VirtualStorageLibrary	

Parameters	Name	Description
VirtualNodeListConditions		

Type	Description
bool ( <a href="https://learn.microsoft.com/dotnet/api/system.boolean">https://learn.microsoft.com/dotnet/api/system.boolean</a> )	

Type	Description
VirtualnodeName	

Type	Description
(AkiraNetwork.VirtualStorageLibrary)	

# Extension Methods

VirtualTextFormatter.GenerateSingleTableDebugText<T>(T)  
(AkiraNetwork.VirtualStorageLibrary.Utilities.VirtualTextFormatter.html#AkiraNetwork\_VirtualStorageLibrary\_Utils  
s\_VirtualTextFormatter\_GenerateSingleTableDebugText\_1\_\_0\_)



(AkiraNetwork.VirtualStorageLibrary  
IVirtualDeepCloneable<T>  
(AkiraNetwork.VirtualStorageLibrary  
1.html)  
IVirtualWildcardMatcher  
(AkiraNetwork.VirtualStorageLibrary  
NotifyNodeDelegate  
(AkiraNetwork.VirtualStorageLibrary  
PatternMatch  
(AkiraNetwork.VirtualStorageLibrary  
VirtualCycleDetector  
(AkiraNetwork.VirtualStorageLibrary  
VirtualDirectory  
(AkiraNetwork.VirtualStorageLibrary  
VirtualDirectoryAdapter<T>  
(AkiraNetwork.VirtualStorageLibrary  
1.html)  
VirtualException  
(AkiraNetwork.VirtualStorageLibrary  
VirtualGroupCondition<T, TKey>  
(AkiraNetwork.VirtualStorageLibrary  
2.html)  
VirtualID  
(AkiraNetwork.VirtualStorageLibrary  
VirtualItem  
(AkiraNetwork.VirtualStorageLibrary  
VirtualItemAdapter<T>  
(AkiraNetwork.VirtualStorageLibrary  
1.html)  
VirtualItem<T>  
(AkiraNetwork.VirtualStorageLibrary  
1.html)  
VirtualNode  
(AkiraNetwork.VirtualStorageLibrary  
VirtualNodeContext  
(AkiraNetwork.VirtualStorageLibrary  
VirtualNodeExtensions  
(AkiraNetwork.VirtualStorageLibrary  
VirtualNodeListConditions  
(AkiraNetwork.VirtualStorageLibrary  
VirtualnodeName  
(AkiraNetwork.VirtualStorageLibrary

# Class VirtualDirectory

Represents a virtual directory. A virtual directory functions as a container for other nodes. It can contain Virtual Item

[Filter API title](#) (AkiraNetwork.VirtualStorageLibrary.VirtualItem-1.html), VirtualDirectory

(AkiraNetwork.VirtualStorageLibrary.VirtualDirectory.html), and VirtualSymbolicLink

(AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink.html), providing a centralized way to manage these entities.

[AkiraNetwork.VirtualStorageLibrary](#)

1.html)

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.

NotifyNodeDelegate

Adds [AkiraNetwork.VirtualStorageLibrary](#) according to the specified display conditions.

PatternMatch

[AkiraNetwork.VirtualStorageLibrary](#)

VirtualCycleDetector

↳ object (<https://learn.microsoft.com/dotnet/api/system.object>)

↳ VirtualNode ([AkiraNetwork.VirtualStorageLibrary.VirtualNode.html](#))

VirtualDirectory

↳ VirtualDirectory

[AkiraNetwork.VirtualStorageLibrary](#)

Implements

DirectoryAdapter<T>

[AkiraNetwork.VirtualStorageLibrary](#) [VirtualDeepCloneable-1.html](#) <[VirtualNode](#)

(AkiraNetwork.VirtualStorageLibrary.VirtualNode.html)>

VirtualException

(AkiraNetwork.VirtualStorageLibrary.VirtualNode.html)>

VirtualGroupCondition (<https://learn.microsoft.com/dotnet/api/system.collections.ienumerable>)

[AkiraNetwork.VirtualStorageLibrary](#)

Inherited Members

2.html)

VirtualNodeName

(AkiraNetwork.VirtualStorageLibrary.VirtualNode.html#AkiraNetwork\_VirtualStorageLibrary\_VirtualNode\_Name)

VirtualNodeCreatedDate

(AkiraNetwork.VirtualStorageLibrary.VirtualNode.html#AkiraNetwork\_VirtualStorageLibrary\_VirtualNode\_CreatedDate)

VirtualItemAdapter<T>

[AkiraNetwork.VirtualStorageLibrary](#)

(AkiraNetwork.VirtualStorageLibrary.VirtualNode.html#AkiraNetwork\_VirtualStorageLibrary\_VirtualNode\_UpdatedDate)

VirtualItem<T>

[AkiraNetwork.VirtualStorageLibrary](#)

(AkiraNetwork.VirtualStorageLibrary.VirtualNode.html#AkiraNetwork\_VirtualStorageLibrary\_VirtualNode\_VID)

VirtualNodeIsReferencedInStorage

(AkiraNetwork.VirtualStorageLibrary.VirtualNode.html#AkiraNetwork\_VirtualStorageLibrary\_VirtualNode\_IsReferencedInStorage)

VirtualNodeContext

object ([&lt;T>.Equals\(system-object\)](https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals(system-object)))

VirtualNodeExtensions

object ([&lt;T>.Equals\(system-object\)](https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals(system-object)))

object ([GetHashCode\(\)](https://learn.microsoft.com/dotnet/api/system.object.gethashcode))

object ([GetType\(\)](https://learn.microsoft.com/dotnet/api/system.object.gettype))

object ([MemberwiseClone\(\)](https://learn.microsoft.com/dotnet/api/system.object.memberwiseclone))

VirtualNodeNotFoundException

object.ReferenceEquals(object, object) (<https://learn.microsoft.com/dotnet/api/system.object.referenceequals>)

**Namespace:** AkiraNetwork (AkiraNetwork.html).VirtualStorageLibrary (AkiraNetwork.VirtualStorageLibrary.html)

**Assembly:** VirtualStorageLibrary.dll

## Syntax

```
public class VirtualDirectory : VirtualNode, IVirtualDeepCloneable<VirtualNode>, IEnumerable<VirtualNode>, IEnumerable<IVirtualDeepCloneable<T>>
```

(AkiraNetwork.VirtualStorageLibrary.html)

VirtualNodeMatcher

(AkiraNetwork.VirtualStorageLibrary.html)

NotifyNodeDelegate

## VirtualDirectory()

(AkiraNetwork.VirtualStorageLibrary.html)

PatternMatch

Initializes a new instance of the VirtualDirectory (AkiraNetwork.VirtualStorageLibrary.VirtualDirectory.html) class.  
(AkiraNetwork.VirtualStorageLibrary.html)

The node name is automatically generated by the GenerateNodeName(string)

VirtualCycleDetector

(AkiraNetwork.VirtualStorageLibrary.VirtualnodeName.html#AkiraNetwork\_VirtualStorageLibrary\_VirtualnodeName\_GeneratenodeName\_System\_String\_.method.

VirtualDirectory

**Declaration**

(AkiraNetwork.VirtualStorageLibrary.html)

VirtualDirectoryAdapter<T>

(AkiraNetwork.VirtualStorageLibrary.html)

1.html

VirtualException

(AkiraNetwork.VirtualStorageLibrary.html)

## VirtualDirectory(VirtualNodeName)

(AkiraNetwork.VirtualStorageLibrary.html)

Initializes a new instance of the VirtualDirectory (AkiraNetwork.VirtualStorageLibrary.VirtualDirectory.html) class  
with the specified name.

VirtualName

(AkiraNetwork.VirtualStorageLibrary.html)

VirtualItem

(AkiraNetwork.VirtualStorageLibrary.html)

public VirtualDirectory(VirtualNodeName name)

VirtualItemAdapter<T>

(AkiraNetwork.VirtualStorageLibrary.html)

## Parameters

Type	Name	Description
VirtualItem<T> (AkiraNetwork.VirtualStorageLibrary.html)	name	The name of the directory.

(AkiraNetwork.VirtualStorageLibrary.html)

VirtualNodeContext

(AkiraNetwork.VirtualStorageLibrary.html)

## VirtualDirectory(VirtualNodeName, DateTime, DateTime)

(AkiraNetwork.VirtualStorageLibrary.html)

Initializes a new instance of the VirtualDirectory (AkiraNetwork.VirtualStorageLibrary.VirtualDirectory.html) class  
with the specified name, creation date, and update date.

(AkiraNetwork.VirtualStorageLibrary.html)

## Declaration

VirtualNodeName

(AkiraNetwork.VirtualStorageLibrary.html)

VirtualNodeNotFoundException



Gets the number of directories based on the current display conditions.

## Declaration

```
public int DirectoryViewCount { get; }
```



### Property Value

Type	Description
virtualDeepCloneable<T> (AkiraNetwork.VirtualStorageLibrary.html) int ( <a href="https://learn.microsoft.com/dotnet/api/system.int32">https://learn.microsoft.com/dotnet/api/system.int32</a> ) 1.html)	

IVirtualWildcardMatcher  
(AkiraNetwork.VirtualStorageLibrary.html)

### this[VirtualNodeName]

(AkiraNetwork.VirtualStorageLibrary.html)

Gets or sets the node with the specified name using the indexer.

(AkiraNetwork.VirtualStorageLibrary.html)  
VirtualCycleDetector  
(AkiraNetwork.VirtualStorageLibrary.html)

```
public VirtualNode this[VirtualNodeName name] { get; set; }
```

VirtualDirectory  
(AkiraNetwork.VirtualStorageLibrary.html)  
VirtualDirectoryAdapter<T>

Type	Name	Description
(AkiraNetwork.VirtualStorageLibrary.html)	name	The name of the node.

VirtualException  
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)  
(AkiraNetwork.VirtualStorageLibrary.html)

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

Type	Description
(AkiraNetwork.VirtualStorageLibrary.html)	The node with the specified name.

VirtualItem  
(AkiraNetwork.VirtualStorageLibrary.html)

VirtualItemAdapter<T>  
(AkiraNetwork.VirtualStorageLibrary.html)

Type	Condition
(AkiraNetwork.VirtualStorageLibrary.html)	Thrown if the node is not found.

VirtualItem<T>  
(AkiraNetwork.VirtualStorageLibrary.html)

VirtualNode  
(AkiraNetwork.VirtualStorageLibrary.html)

Gets the number of items within the directory.

VirtualNodeContext  
(AkiraNetwork.VirtualStorageLibrary.html)

VirtualNodeExtensions  
(AkiraNetwork.VirtualStorageLibrary.html)

```
public int ItemCount { get; }
```

VirtualNodeListConditions  
(AkiraNetwork.VirtualStorageLibrary.html)

VirtualStorageLibrary  
(AkiraNetwork.VirtualStorageLibrary.html)

VirtualNodeName  
(AkiraNetwork.VirtualStorageLibrary.html)

VirtualNodeNotFoundException  
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)

Type	Description
int ( <a href="https://learn.microsoft.com/dotnet/api/system.int32">https://learn.microsoft.com/dotnet/api/system.int32</a> )	

## ItemCount

Gets the number of items based on the current display conditions.

IVirtualDeepCloneable<T>

Declaration

[AkiraNetwork.VirtualStorageLibrary](#)

1.html)

public int ItemCount { get; }

(AkiraNetwork.VirtualStorageLibrary)

NotifyNodeDelegate

Property Value

(AkiraNetwork.VirtualStorageLibrary)

Type

PatternMatch

(AkiraNetwork.VirtualStorageLibrary)

int (<https://learn.microsoft.com/dotnet/api/system.int32>)

Description

Type	Description
PatternMatch	

VirtualWildcardMatcher

(AkiraNetwork.VirtualStorageLibrary)

VirtualCycleDetector

(AkiraNetwork.VirtualStorageLibrary)

VirtualDirectory

(AkiraNetwork.VirtualStorageLibrary)

NodeNames

VirtualDirectoryAdapter<T>

Enum

(AkiraNetwork.VirtualStorageLibrary)

VirtualDirectoryName

(AkiraNetwork.VirtualStorageLibrary)

1.html)

Declaration

VirtualException

(AkiraNetwork.VirtualStorageLibrary)

public IEnumerable<VirtualNodeName> NodeNames { get; }

(AkiraNetwork.VirtualStorageLibrary)

2.html)

Property Value

VirtualID

Type

(AkiraNetwork.VirtualStorageLibrary)

VirtualItem

(AkiraNetwork.VirtualStorageLibrary)

IEnumerable

(<https://learn.microsoft.com/dotnet/api/system.collections.generic.ienumerable-1>)<

(AkiraNetwork.VirtualStorageLibrary)

VirtualnodeName

(AkiraNetwork.VirtualStorageLibrary.VirtualnodeName.html)

VirtualItemAdapter<T>

(AkiraNetwork.VirtualStorageLibrary)

1.html)

NodeType

VirtualItem<T>

(AkiraNetwork.VirtualStorageLibrary)

Gets the type of the node. This class always returns Directory.

1.html)

Declaration

VirtualNode

(AkiraNetwork.VirtualStorageLibrary)

public override VirtualNodeType NodeType { get; }

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeExtensions

Property Value

(AkiraNetwork.VirtualStorageLibrary)

Type	Description
VirtualNodeListConditions	

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeType.html)

(AkiraNetwork.VirtualStorageLibrary)

Overrides

VirtualNodeNotFoundException

Type	Description
VirtualNodeNotFoundException	

VirtualNode.NodeType  
(AkiraNetwork.VirtualStorageLibrary.VirtualNode.html#AkiraNetwork\_VirtualStorageLibrary\_VirtualNode\_NodeType)

## Nodes

Enumerates all nodes within the directory.

**Declaration**  
IVirtualDeepCloneable<T>  
(AkiraNetwork.VirtualStorageLibrary  
1.html)  
public IEnumerable<VirtualNode> Nodes { get; }  
IVirtualWildcardMatcher  
(AkiraNetwork.VirtualStorageLibrary

### Property Value

Type	Description
IEnumerable<VirtualNode> (AkiraNetwork.VirtualStorageLibrary VirtualNode.html)	

## NodesView

Gets a view of nodes based on the current display conditions.  
1.html)

### Declaration

Exception  
(AkiraNetwork.VirtualStorageLibrary  
VirtualGroupCondition<VirtualNode> NodesView { get; }  
(AkiraNetwork.VirtualStorageLibrary  
2.html)

### Property Value

Type	Description
VirtualItem IEnumerable<VirtualItem> (AkiraNetwork.VirtualStorageLibrary VirtualNode.html)	

(AkiraNetwork.VirtualStorageLibrary  
VirtualItemAdapter<T>  
1.html)

## NodesViewCount

Gets the number of nodes based on the current display conditions.  
1.html)

### Declaration

VirtualNode  
(AkiraNetwork.VirtualStorageLibrary  
VirtualNodeContext  
public int NodesViewCount { get; }  
(AkiraNetwork.VirtualStorageLibrary  
VirtualNodeExtensions

### Property Value

Type	Description
VirtualNodeListConditions int (https://learn.microsoft.com/dotnet/api/system.int32)	

## SymbolicLinkCount

Gets the number of symbolic links within the directory.

## Declaration

```
    public int SymbolicLinkCount { get; }
```

Type	Value	Description
VirtualDeepCloneable<T> (AkiraNetwork.VirtualStorageLibrary.html)		
VirtualWildcardMatcher int ( <a href="https://learn.microsoft.com/dotnet/api/system.int32">https://learn.microsoft.com/dotnet/api/system.int32</a> ) (AkiraNetwork.VirtualStorageLibrary.html)		

(AkiraNetwork.VirtualStorage)  
NotifyNodeDelegate  
(AkiraNetwork.VirtualStorage)  
**SymbolicLinkViewCount**  
[Later update]

(AkiraNetwork.VirtualStorageLibrary)  
Gets the number of symbolic links based on the current display conditions.  
VirtualCycleDetector

## Declaration (Akira Network Virtual Storage Library)

```
VirtualDirectory  
(AkiraNetwork.VirtualStorageLibrary  
public int SymbolicLinkViewCount { get; }  
VirtualDirectoryAdapter<T>  
(AkiraNetwork.VirtualStorageLibrary
```

Property	Value
Type	VirtualException (AkiraNetwork.VirtualStorageLibrary) int ( <a href="https://learn.microsoft.com/dotnet/api/system.int32">https://learn.microsoft.com/dotnet/api/system.int32</a> )
VirtualGroupCondition	<Key>

(AkiraNetwo

## Methods

**VirtualItem**

(AkiraNe)

`VirtualNode<bool>`

Add(**VirtualStorage**, **host**)  
(AkiraNetwork.VirtualStorage)

Adds a node to the directory.

`VirtualItem<T>`

## Declaration (AkiraNetworks)

(AkiraNetwork.VirtualStorageLibrary  
1.1.1.1)

```
1.html)  
public VirtualNode Add(VirtualN
```

VirtualN

(AkiraN)

## Virtual Node Parameters

(AkiraNetwork.VirtualStorageLibrary)

Type VirtualNodeExtensions

Virtu(AkiraN) Network VirtualStorage library

(AkiVirtualNordicListConditionsLibrary VirtualNo

(AkiraNetwork.VirtualStorageLibrary.VII  
to 1/AkiraNetwork.VirtualStorageLibrary

Virtuaal Network

**VirtualNodeName**  
(AlpineNetworks)VirtualStorageLibraryName

(AkilaNetwork.VirtualStorageLibrary)

Type	Name	Description
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeExtensions)		
(AkiraNetwork.VirtualStorageLibrary.VirtualNode)	<i>node</i>	The node to add.
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeListConditions)		
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeNotFoundException)		

Type	Name	Description
bool ( <a href="https://learn.microsoft.com/dotnet/api/system.boolean">https://learn.microsoft.com/dotnet/api/system.boolean</a> )	<i>allowOverwrite</i>	If true, allows overwriting an existing node with the same name.

## Returns

Type	Description
<code>VirtualDeepCloneable&lt;T&gt;</code> ( <a href="#">AkiraNetwork.VirtualStorageLibrary.VirtualNode.html</a> )	The added node.

Exceptions  
IVirtualWildcardMatcher  
([AkiraNetwork.VirtualStorageLibrary](#))

Type	Condition
<code>NotifyNodeDelegate</code> ( <a href="#">AkiraNetwork.VirtualStorageLibrary</a> ) <code>ArgumentException</code> ( <a href="#">PatternMatch</a> ( <a href="https://learn.microsoft.com/dotnet/api/system.argumentexception">https://learn.microsoft.com/dotnet/api/system.argumentexception</a> ) ( <a href="#">AkiraNetwork.VirtualStorageLibrary</a> )	Thrown if an invalid node name is specified.
<code>InvalidOperationException</code> ( <a href="#">VirtualDirectory</a> ( <a href="https://learn.microsoft.com/dotnet/api/system.invalidoperationexception">https://learn.microsoft.com/dotnet/api/system.invalidoperationexception</a> ) <code>VirtualDirectory</code> ( <a href="#">AkiraNetwork.VirtualStorageLibrary</a> ) <code>VirtualDirectoryAdapter&lt;T&gt;</code> ( <a href="#">AkiraNetwork.VirtualStorageLibrary</a> )	Thrown if a node with the same name already exists.

## AddDirectory(VirtualNodeName, bool)

`VirtualException`  
Adds a directory with the specified name.  
([AkiraNetwork.VirtualStorageLibrary](#))

Declaration  
`GroupCondition<T, TKey>`

```
public VirtualDirectory AddDirectory(VirtualNodeName name, bool allowOverwrite = false)
```

Parameters  
`VirtualItem`

Type	Name	Description
<code>VirtualItemAdapter&lt;T&gt;</code> <code>VirtualNodeName</code> ( <a href="#">AkiraNetwork.VirtualStorageLibrary</a> ) ( <a href="#">AkiraNetwork.VirtualStorageLibrary</a> ) ( <a href="#">VirtualNodeName</a> )	<i>name</i>	The name of the directory.
<code>bool</code> ( <a href="https://learn.microsoft.com/dotnet/api/system.boolean">https://learn.microsoft.com/dotnet/api/system.boolean</a> )	<i>allowOverwrite</i>	If true, allows overwriting an existing node with the same name.

Returns  
`VirtualNodeContext`  
([AkiraNetwork.VirtualStorageLibrary](#))

Type	Description
<code>VirtualNodeExtensions</code> ( <a href="#">AkiraNetwork.VirtualStorageLibrary</a> ) <code>VirtualDirectory</code> ( <a href="#">AkiraNetwork.VirtualStorageLibrary</a> ) <code>VirtualNodeListConditions</code> ( <a href="#">AkiraNetwork.VirtualStorageLibrary</a> ) <code>VirtualNodeName</code> ( <a href="#">AkiraNetwork.VirtualStorageLibrary</a> )	The added directory.

## AddItem<T>(VirtualNodeName, T?, bool)

`VirtualNodeNotFoundException`

Adds an item with the specified name.

## Declaration

```
public VirtualItem<T> AddItem<T>(VirtualNodeName name, T? itemData = default, bool allowOverwrite = false)
```



### Parameters

VirtualDeepCloneable<T>	
<b>Type</b> AkiraNetwork.VirtualStorageLibrary	<b>Name</b>
VirtualNodeName VirtualWildcardMatcher (AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html) NotifyNodeDelegate	<i>name</i>
T (AkiraNetwork.VirtualStorageLibrary)	<i>itemData</i>

PatternMatch  
bool  
(AkiraNetwork.VirtualStorageLibrary  
(https://learn.microsoft.com/dotnet/api/system.boolean))  
VirtualCycleDetector

(AkiraNetwork.VirtualStorageLibrary)

VirtualDirectory

**Return** AkiraNetwork.VirtualStorageLibrary

VirtualDirectoryAdapter<T>	
(AkiraNetwork.VirtualStorageLibrary VirtualItem (AkiraNetwork.VirtualStorageLibrary.VirtualItem-1.html)<T> .html)	The added item.

VirtualException

**Type** AkiraNetwork.VirtualStorageLibrary

VirtualGroupCondition<T, TKey>	
(AkiraNetwork.VirtualStorageLibrary T 2.html)	The data type of the item.
VirtualID (AkiraNetwork.VirtualStorageLibrary VirtualItem (AkiraNetwork.VirtualStorageLibrary))	

## AddSymbolicLink(VirtualNodeName, VirtualPath, bool)

VirtualItem Adapter<T>

Adds a symbolic link with the specified name.

(AkiraNetwork.VirtualStorageLibrary)

## Declaration

VirtualItem<T>

```
public void AddSymbolicLink(VirtualNodeName name, VirtualPath targetPath, bool allowOverwrite = false)
```

VirtualNode

(AkiraNetwork.VirtualStorageLibrary)

### Parameters

VirtualNodeContext	
<b>Type</b> AkiraNetwork.VirtualStorageLibrary	<b>Name</b>
VirtualNodeExtensions VirtualNodeName (AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html) VirtualNodeListConditions	<i>name</i>
(AkiraNetwork.VirtualStorageLibrary VirtualnodeName.html)	
VirtualNodeName (AkiraNetwork.VirtualStorageLibrary VirtualNodeNotFoundException)	



```
public bool DirectoryExists(VirtualNodeName name)
```

#### Parameters

Type		Name	Description
VirtualNodeName (AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)  (AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)		name	The name of the directory to check.

#### Returns

Type	IVirtualWildcardMatcher (AkiraNetwork.VirtualStorageLibrary.VirtualWildcardMatcher.html)	Description
bool (http://dotnet.developersoftware.com/dotnet/api/system.boolean)  (AkiraNetwork.VirtualStorageLibrary.VirtualBoolean.html)		True if the directory exists; otherwise, false.

## Get(VirtualNodeName, bool)

Gets the node with the specified name.

VirtualDirectory

Declaration

```
public VirtualItem Get(VirtualNodeName name, bool exceptionEnabled = true)  
1.html)
```

VirtualException

Parameters

Type	VirtualGroupCondition<T, TKey> (AkiraNetwork.VirtualStorageLibrary.VirtualGroupCondition.html) VirtualNodeName (AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html) VirtualID (AkiraNetwork.VirtualStorageLibrary.VirtualID.html) (AkiraNetwork.VirtualStorageLibrary.VirtualItem.html)	Name	Description
bool (http://dotnet.developersoftware.com/dotnet/api/system.boolean) VirtualItemAdapter<T> (AkiraNetwork.VirtualStorageLibrary.VirtualItemAdapter.html) 1.html)	exceptionEnabled		If true, throws an exception if the node is not found.

#### Returns

Type	VirtualItem<T> (AkiraNetwork.VirtualStorageLibrary.VirtualItem.html) VirtualNode (AkiraNetwork.VirtualStorageLibrary.VirtualNode.html) (AkiraNetwork.VirtualStorageLibrary.VirtualNode.html)	Description
VirtualNodeContext (AkiraNetwork.VirtualStorageLibrary.VirtualNodeContext.html)		The node with the specified name, or null if the node does not exist.

## GetDirectory(VirtualNodeName)

Gets the directory with the specified name.

VirtualNodeListConditions  
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeListConditions.html)

Declaration

```
public VirtualDirectory GetDirectory(VirtualNodeName name)  
1.html)
```

VirtualNodeNotFoundException  
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeNotFoundException.html)

```
public VirtualDirectory GetDirectory(VirtualNodeName name)
```

#### Parameters

Type	Name	Description
VirtualNodeName (AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)  (AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)	name	The name of the directory.

#### Returns

Type	Description
IVirtualWildcardMatcher (AkiraNetwork.VirtualStorageLibrary.VirtualWildcardMatcher.html)	The directory with the specified name.

#### Exceptions

Type	Condition
VirtualCycleDetector (AkiraNetwork.VirtualStorageLibrary.VirtualCycleDetector.html)	
VirtualNotFoundException (AkiraNetwork.VirtualStorageLibrary.VirtualNotFoundException.html)	Thrown if the node is not found.
VirtualDirectoryAdapter<T> InvalidOperationException (AkiraNetwork.VirtualStorageLibrary.InvalidOperationException.html)	Thrown if the specified node is not a directory.

#### GetEnumerator()

#### Returns

Gets an enumerator for nodes based on the current display conditions.

#### Declaration

```
(AkiraNetwork.VirtualStorageLibrary)
```

VirtualItem

```
public System.Collections.IEnumerable GetEnumerator()
```

VirtualItemAdapter<T>

```
(AkiraNetwork.VirtualStorageLibrary)
```

#### Returns

1.html

#### Type

VirtualItem<T>

```
(AkiraNetwork.VirtualStorageLibrary)
```

#### 1.html

(https://learn.microsoft.com/dotnet/api/system.collections.generic.ienumerator-1)

<VirtualNode>(AkiraNetwork.VirtualStorageLibrary.VirtualNode.html)>

```
(AkiraNetwork.VirtualStorageLibrary)
```

VirtualNodeContext

```
(AkiraNetwork.VirtualStorageLibrary)
```

#### GetItemByName(VirtualNodeName)

```
(AkiraNetwork.VirtualStorageLibrary)
```

Gets the item with the specified name.

VirtualNodeListConditions

```
(AkiraNetwork.VirtualStorageLibrary)
```

VirtualNodeName

```
(AkiraNetwork.VirtualStorageLibrary)
```

VirtualNodeNotFoundException

```
public VirtualItem<T> GetItem<T>(VirtualNodeName name)
```

## Parameters

Type	Name	Description
VirtualNodeName (AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html) IVirtualDeepCloneable<T>	name	The name of the item.

Return Value

Type	Description
Virt (AkiraNetwork.VirtualStorageLibrary.VirtualItem-1.html)<T>	The item with the specified name.

Type Parameters

Type	Parameters	Description
	PatternMatch	(AkiraNetwork.VirtualStorageLibrary)
T	VirtualCycleDetector	The data type of the item. (AkiraNetwork.VirtualStorageLibrary)

Exceptions

Type	Condition
VirtualDirectoryAdapter<T>	Thrown if the node is not found. (AkiraNetwork.VirtualStorageLibrary)
VirtualNotFoundException 1.html (AkiraNetwork.VirtualStorageLibrary.VirtualNodeNotFoundException.html)	Thrown if the specified node is not an item. (https://learn.microsoft.com/dotnet/api/system.invalidoperationexception) (AkiraNetwork.VirtualStorageLibrary)

## GetNodesView()

VirtualItem

Gets a view of nodes based on the current display conditions.

(AkiraNetwork.VirtualStorageLibrary)

Declaration

VirtualItemAdapter<T>

(AkiraNetwork.VirtualStorageLibrary)

```
public IEnumerable<VirtualNode> GetNodesView()
```

VirtualItem<T>

(AkiraNetwork.VirtualStorageLibrary)

Returns

T.html

Type	Description
VirtualNode	An enumeration of nodes based on the display conditions. (AkiraNetwork.VirtualStorageLibrary) IEnumerable VirtualNodeContext (https://learn.microsoft.com/dotnet/api/system.collections.generic.IEnumerable-1) VirtualNodeExtensions <VirtualNode (AkiraNetwork.VirtualStorageLibrary.VirtualNode.html)> (AkiraNetwork.VirtualStorageLibrary)

VirtualNodeListConditions

(AkiraNetwork.VirtualStorageLibrary)

## GetSymbolicLink(VirtualNodeName)

(AkiraNetwork.VirtualStorageLibrary)

Gets the symbolic link with the specified name.

VirtualNodeNotFoundException

## Declaration

```
public VirtualSymbolicLink GetSymbolicLink(VirtualNodeName name)
```

## Parameters

Type		Name	Description
VirtualNodeName (AkiraNetworkVirtualStorageLibraryVirtualNodeName.html)	VirtualDeepCloneable<T> (AkiraNetworkVirtualStorageLibraryVirtualNodeName.html)	<i>name</i>	The name of the symbolic link.

## **BetuVirtualWildcardMatcher**

Type	Description
(AkiraNetwork.VirtualStorageLibrary.NotifyNodeDelegate)	
Virt(AkiraNetwork.VirtualStorageLibrary)	The symbolic link with the specified name.
(AkiraPatternMatVirtualStorageLibrary.VirtualSymbolicLink.html)	

#### **VirtualCycleDetector**

Type	Condition
(AkiraNetwork.VirtualStorageLibrary.VirtualDirectoryException)	Thrown if the node is not found.
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeNotFoundException.htm l) (AkiraNetwork.VirtualStorageLibrary.VirtualDirectoryAdapter.T)	Thrown if the specified node is not a symbolic link.
(https://learn.microsoft.com/dotnet/api/system.invalidoperationexception)	

**ItemExists(VirtualNodeName)**

(AkiraNetwork.VirtualStorageLibrary)

Check **Virtuwalther**, an item with the specified name exists.

CHICKS WING  
(AkiraN)

```
Declaration VirtualItemAdapter<T>
(AkiraNetwork.VirtualStorageLibrary)
public bool ItemExists(VirtualNodeName name)
```

## Parameters (AkiraN)

Type	Name	Description
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)		
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)	<i>name</i>	The name of the item to check.
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeContext.html)		

## **VirtualNetwork.Virtual**

**Returns**: `VirtualNodeExtensions` object.

Type	Description
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeListConditions)	True if the item exists; otherwise, false.

### **NodeExists(VirtualNodeName)**

Checks whether a node with the specified name exists.

## Declaration

```
    public bool NodeExists(VirtualnodeName name)
```

Parameters	<code>VirtualDeepCloneable&lt;I&gt;</code> (AkiraNetwork.VirtualStorageLibrary.html)	Name	Description
<code>VirtualWildcardMatcher</code> <code>VirtualNodeName</code> (AkiraNetwork.VirtualStorageLibrary.VirtualnodeName.html)	<code>name</code>	The name of the node to check.	

<b>Returns</b>	(AkiraNetwork.VirtualStorageLibrary.PatternMatch)
<b>Type</b>	AkiraNetwork.VirtualStorageLibrary.PatternMatch

## Remove Virtual Node

(AkiraNetwork.VirtualStorageLibrary)

**RemoveNode**(*node*)  
Removes the specified node from the directory.

## VirtualException Declaration

(AkiraNetwork VirtualStorage) library

```
VirtualGroupCondition<T, TKey>
public void Remove(VirtualNode node)
(AkiraNetwork VirtualStorage) library
```

2.html

Parameter ID	Type	Name	Description
(AkiraNetwork.VirtualStorageLibrary.VirtualItem)			
(AkiraNetwork.VirtualStorageLibrary.VirtualNode)		node	The node to remove.

Type	Condition
VirtualItem<T>	
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeNotFoundException.html)	Thrown if the node is not found.

## SymbolicLinkExists(VirtualNodeName)

**VirtualNodeExtensions**  
Checks whether a symbolic link with the specified name exists.  
([libNetwork/virtualStorageLibrary](#))

## DeclVirtualNode istConditions

#### **Advanced Network Conditions**

(AkiraNetwork.virtual)

```
public bool SymbolicLinkExists()
```

## (AkiraNetwork.VirtualStorageLib

## VirtualNodeNotFoundException

## Parameters

Type	Name	Description
VirtualNodeName (AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)	name	The name of the symbolic link to check.

## Returns

Type	Description
IVirtualDeepCloneable<T> (AkiraNetwork.VirtualStorageLibrary.VirtualNode.html)	True if the symbolic link exists; otherwise, false.

## IVirtualWildcardMatcher

### (AkiraNetwork.VirtualStorageLibrary.VirtualNode)

>Returns a string representation of the directory.  
PatternMatch

## Declaration

```
(AkiraNetwork.VirtualStorageLibrary.VirtualCycleDetector  
(AkiraNetwork.VirtualStorageLibrary.VirtualDirectory  
public override string ToString()  
VirtualDirectory  
(AkiraNetwork.VirtualStorageLibrary.VirtualDirectoryAdapter<T>
```

## Returns

Type	Description
(AkiraNetwork.VirtualStorageLibrary.VirtualDirectoryAdapter<T>.html)	A string representation of the directory.

## Overrides

```
object.ToString() (https://learn.microsoft.com/dotnet/api/system.object.tostring)  
2.html
```

## VirtualID

## Update(VirtualNode)

### (AkiraNetwork.VirtualStorageLibrary.VirtualItem)

Updates the current directory with the data from the specified node.  
(AkiraNetwork.VirtualStorageLibrary.VirtualItem)

## Declaration

```
(AkiraNetwork.VirtualStorageLibrary.VirtualItemAdapter<T>  
1.html)  
public override void Update(VirtualNode node)  
VirtualItem<T>  
(AkiraNetwork.VirtualStorageLibrary.VirtualItemAdapter<T>)
```

## Parameters

Type	Name	Description
VirtualNode (AkiraNetwork.VirtualStorageLibrary.VirtualNode.html)	node	The node used for the update.

## Overrides

```
VirtualNodeExtensions.UpdateVirtualNode  
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeExtensions)  
VirtualNodeListConditions  
(AkiraNetwork.VirtualStorageLibrary.VirtualNode.html#AkiraNetwork_VirtualStorageLibrary_VirtualNode_Update_A  
kiraNetwork_VirtualStorageLibrary_VirtualNode_)  
VirtualNodeName  
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)
```

## Exceptions

```
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeNotFoundException  
VirtualNodeNotFoundException)
```



## Returns

Type	Description
VirtualDirectory (AkiraNetwork.VirtualStorageLibrary.VirtualDirectory.html)	

## operator -(VirtualDirectory, VirtualNode)

|VirtualDeepCloneable<T>  
Overloads the subtraction operator to remove a node from the directory.  
(AkiraNetwork.VirtualStorageLibrary)

### Declaration

IVirtualWildcardMatcher  
public static VirtualDirectory operator -(VirtualDirectory directory, VirtualNode node)

NotifyNodeDelegate

(AkiraNetwork.VirtualStorageLibrary)

### Parameters

PatternMatch

Type	Name	Description
AkiraNetwork.VirtualStorageLibrary VirtualCycleDetector	directory	The directory from which the node is removed.
AkiraNetwork.VirtualStorageLibrary (AkiraNetwork.VirtualStorageLibrary.VirtualDirectory.html)	node	The node to remove.

### Returns

VirtualException

Type	Description
AkiraNetwork.VirtualStorageLibrary VirtualGroupCondition<T, TKey> VirtualDirectory (AkiraNetwork.VirtualStorageLibrary) (AkiraNetwork.VirtualStorageLibrary.VirtualDirectory.html)	The directory with the removed node.

VirtualID

(AkiraNetwork.VirtualStorageLibrary)

## Implements

IVirtualDeepCloneable<T>	(AkiraNetwork.VirtualStorageLibrary.IVirtualDeepCloneable-1.html)
IEnumerable<T>	(https://learn.microsoft.com/dotnet/api/system.collections.generic.ienumerable-1)
IEnumerable	(https://learn.microsoft.com/dotnet/api/system.collections.ienumerable)
VirtualItem<T>	

## Extension Methods

VirtualTextFormatter.GenerateSingleTableDebugText<T>(T)	(AkiraNetwork.VirtualStorageLibrary.Utilities.VirtualTextFormatter.html#AkiraNetwork_VirtualStorageLibrary_Utils s_VirtualTextFormatter_GenerateSingleTableDebugText_1__0_)
VirtualNodeExtensions.ResolveNodeType(VirtualNode)	(AkiraNetwork.VirtualStorageLibrary.VirtualNodeExtensions.html#AkiraNetwork_VirtualStorageLibrary_VirtualNode Extensions_ResolveNodeType_AkiraNetwork_VirtualStorageLibrary_VirtualNode_)
VirtualTextFormatter.GenerateTableDebugText<T>(IEnumerable<T>)	(AkiraNetwork.VirtualStorageLibrary.Utilities.VirtualTextFormatter.html#AkiraNetwork_VirtualStorageLibrary_Utils s_VirtualTextFormatter_GenerateTableDebugText_1_System_Collections_Generic_IEnumerable__0_)
VirtualStorageExtensions.ApplySortConditions<T>(IEnumerable<T>, List<VirtualSortCondition<T>>?)	(AkiraNetwork.VirtualStorageLibrary.VirtualStorageExtensions.html#AkiraNetwork_VirtualStorageLibrary_VirtualSt orageExtensions_ApplySortConditions_1_System_Collections_Generic_IList__0_)
VirtualNodeNotFoundException	(AkiraNetwork.VirtualStorageLibrary.VirtualStorageExtensions.html#AkiraNetwork_VirtualStorageLibrary_VirtualSt orageExtensions_VirtualNodeNotFoundException)

```
rageExtensions_ApplySortConditions_1_System_Collections_Generic_IEnumerator_0_System_Collections_Generi
c_List_AkiraNetwork_VirtualStorageLibrary_VirtualSortCondition_0__)
VirtualStorageExtensions.GroupAndSort<T>(IEnumerable<T>, VirtualGroupCondition<T, object>?, List<Virtual
SortCondition<T>>?)
(AkiraNetwork.VirtualStorageLibrary.VirtualStorageExtensions.html#AkiraNetwork_VirtualStorageLibrary_VirtualSto
rageExtensions_GroupAndSort_1_System_Collections_Generic_IEnumerator_0_AkiraNetwork_VirtualStorageLibr
ary_VirtualGroupCondition_0_System_Object_System_Collections_Generic_List_AkiraNetwork_VirtualStorageLibr
ary_VirtualSortCondition_0__)
IVirtualDeepCloneable<T>
(AkiraNetwork.VirtualStorageLibrar
1.html)
IVirtualWildcardMatcher
(AkiraNetwork.VirtualStorageLibrar
NotifyNodeDelegate
(AkiraNetwork.VirtualStorageLibrar
PatternMatch
(AkiraNetwork.VirtualStorageLibrar
VirtualCycleDetector
(AkiraNetwork.VirtualStorageLibrar
VirtualDirectory
(AkiraNetwork.VirtualStorageLibrar
VirtualDirectoryAdapter<T>
(AkiraNetwork.VirtualStorageLibrar
1.html)
VirtualException
(AkiraNetwork.VirtualStorageLibrar
VirtualGroupCondition<T, TKey>
(AkiraNetwork.VirtualStorageLibrar
2.html)
VirtualID
(AkiraNetwork.VirtualStorageLibrar
VirtualItem
(AkiraNetwork.VirtualStorageLibrar
VirtualItemAdapter<T>
(AkiraNetwork.VirtualStorageLibrar
1.html)
VirtualItem<T>
(AkiraNetwork.VirtualStorageLibrar
1.html)
VirtualNode
(AkiraNetwork.VirtualStorageLibrar
VirtualNodeContext
(AkiraNetwork.VirtualStorageLibrar
VirtualNodeExtensions
(AkiraNetwork.VirtualStorageLibrar
VirtualNodeListConditions
(AkiraNetwork.VirtualStorageLibrar
VirtualnodeName
(AkiraNetwork.VirtualStorageLibrar
VirtualNodeNotFoundException
```

# Class VirtualDirectoryAdapter<T>

## Inheritance

↳ [Object](#) (<https://learn.microsoft.com/dotnet/api/system.object>)

↳ [VirtualDirectoryAdapter<T>](#)  
([AkiraNetwork.VirtualStorageLibrary](#))

## Inherited Members

[Object.Equals](#) ([https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals\(system-object-system-object\)](https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals(system-object-system-object)))

[Object.Equals](#) ([https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals\(system-object-system-object\)](https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals(system-object-system-object)))

[Object.GetHashCode](#) () (<https://learn.microsoft.com/dotnet/api/system.object.gethashcode>)

[Object.GetType](#) () (<https://learn.microsoft.com/dotnet/api/system.object.gettype>)

[Object.MemberwiseClone](#) () (<https://learn.microsoft.com/dotnet/api/system.object.memberwiseclone>)

[Object.ReferenceEquals](#) ([VirtualStorageLibrary](#)) (<https://learn.microsoft.com/dotnet/api/system.object.referenceequals>)

[Object.ToString](#) () (<https://learn.microsoft.com/dotnet/api/system.object.tostring>)

**NameSpace:** [AkiraNetwork](#) ([AkiraNetwork.html](#)).VirtualStorageLibrary ([AkiraNetwork.VirtualStorageLibrary.html](#))

**Assembly:** [VirtualStorageLibrary.dll](#)

([AkiraNetwork.VirtualStorageLibrary](#))

## Syntax

[VirtualException](#)

[VirtualDirectoryAdapter<T>](#)

[VirtualGroupCondition<T, TKey>](#)

**Type Parameters**  
[2.html](#)

Name	Description
<a href="#">T</a>	( <a href="#">AkiraNetwork.VirtualStorageLibrary</a> ) <a href="#">VirtualItem</a>

([AkiraNetwork.VirtualStorageLibrary](#))

[VirtualItemAdapter<T>](#)

([AkiraNetwork.VirtualStorageLibrary](#))  
[1.html](#)

[VirtualItem<T>](#)

**VirtualDirectoryAdapter(VirtualStorage<T>)**

([AkiraNetwork.VirtualStorageLibrary](#))  
[1.html](#)

[VirtualNode](#)

([AkiraNetwork.VirtualStorageLibrary](#))  
public [VirtualDirectoryAdapter](#)([VirtualStorage<T>](#) storage)  
[VirtualNodeContext](#)

([AkiraNetwork.VirtualStorageLibrary](#))

**Parameters**  
[VirtualNodeExtensions](#)

Type	Name	Description
( <a href="#">AkiraNetwork.VirtualStorageLibrary</a> ) <a href="#">VirtualNodeListConditions</a>		
<a href="#">VirtualStorage</a> ( <a href="#">AkiraNetwork.VirtualStorageLibrary</a> ) <a href="#">VirtualStorage-1.html</a>	<a href="#">storage</a>	
	<a href="#">VirtualNodeName</a>	
	( <a href="#">AkiraNetwork.VirtualStorageLibrary</a> )	
	<a href="#">VirtualNodeNotFoundException</a>	
	( <a href="#">AkiraNetwork.VirtualStorageLibrary</a> )	

# Properties

## this[VirtualPath, bool]

### Declaration

```
public VirtualDirectory this[VirtualPath path, bool followLinks = true] { get; set; }  
(AkiraNetwork.VirtualStorageLibrary  
1.html)
```

### Parameters

Type	Description
AkiraNetwork.VirtualStorageLibrary.NotifyNodeDelegate	
VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	<i>path</i>

bool (https://learn.microsoft.com/dotnet/api/system.boolean)

Type	Description
VirtualDirectory	
(AkiraNetwork.VirtualStorageLibrary.VirtualDirectory.html)	

VirtualDirectoryAdapter<T>

(AkiraNetwork.VirtualStorageLibrary  
1.html)

## Extension Methods

VirtualTextFormatter.GenerateSingleTableDebugText<T>(T)  
(AkiraNetwork.VirtualStorageLibrary.Uses.VirtualTextFormatter.html#AkiraNetwork\_VirtualStorageLibrary\_Utilities\_VirtualTextFormatter\_GenerateSingleTableDebugText\_1\_\_0\_)

2.html

VirtualID

(AkiraNetwork.VirtualStorageLibrary)

VirtualItem

(AkiraNetwork.VirtualStorageLibrary)

VirtualItemAdapter<T>

(AkiraNetwork.VirtualStorageLibrary)

1.html

VirtualItem<T>

(AkiraNetwork.VirtualStorageLibrary)

1.html

VirtualNode

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeContext

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeExtensions

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeListConditions

(AkiraNetwork.VirtualStorageLibrary)

VirtualnodeName

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeNotFoundException

(AkiraNetwork.VirtualStorageLibrary)

# Class VirtualException

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

Filter by title

## Inheritance

1.html

- ↳ [object](#) (<https://learn.microsoft.com/dotnet/api/system.object>)
- ↳ [VirtualWildcardMatcher](#) ([AkiraNetwork.VirtualStorageLibrary](#))
- ↳ [Exception](#) (<https://learn.microsoft.com/dotnet/api/system.exception>)
- ↳ [VirtualException](#) ([AkiraNetwork.VirtualStorageLibrary](#))
- ↳ [NotifyNodeDelegate](#) ([AkiraNetwork.VirtualStorageLibrary](#))
- ↳ [VirtualNodeNotFoundException](#) ([AkiraNetwork.VirtualStorageLibrary](#))
- ↳ [PatternMatch](#) ([AkiraNetwork.VirtualStorageLibrary.VirtualNodeNotFoundException.html](#))

## Implements

- ↳ [ISerializable](#) (<https://learn.microsoft.com/dotnet/api/system.runtime.serialization.iserializable>)
- ↳ [\(AkiraNetwork.VirtualStorageLibrary\)](#)

## Inherited Members

[VirtualDirectory](#)

- Exception ([AkiraGetBaseException.html](#)) (<https://learn.microsoft.com/dotnet/api/system.exception.getbaseexception>)
- ExceptionType ([AkiraGetType.html](#)) (<https://learn.microsoft.com/dotnet/api/system.exception.gettype>)
- ExceptionToString () ([AkiraToString.html](#)) (<https://learn.microsoft.com/dotnet/api/system.exception.tostring>)
- ExceptionData ([AkiraExceptionData.html](#)) (<https://learn.microsoft.com/dotnet/api/system.exception.data>)
- ExceptionHelpLink ([AkiraHelpLink.html](#)) (<https://learn.microsoft.com/dotnet/api/system.exception.helplink>)
- ExceptionHRESULT ([AkiraHRESULT.html](#)) ([https://learn.microsoft.com/dotnet/api/system.exception\\_hresult](https://learn.microsoft.com/dotnet/api/system.exception_hresult))
- ExceptionInnerException ([AkiraInnerException.html](#)) (<https://learn.microsoft.com/dotnet/api/system.exception.innerexception>)
- ExceptionMessage ([AkiraMessage.html](#)) (<https://learn.microsoft.com/dotnet/api/system.exception.message>)
- ExceptionSource ([AkiraSource.html](#)) (<https://learn.microsoft.com/dotnet/api/system.exception.source>)
- ExceptionStackTrace ([AkiraStackTrace.html](#)) (<https://learn.microsoft.com/dotnet/api/system.exception.stacktrace>)
- ExceptionTargetSite ([AkiraTargetSite.html](#)) (<https://learn.microsoft.com/dotnet/api/system.exception.targetsite>)
- ExceptionSerializeObjectState ([AkiraSerializeObjectState.html](#)) (<https://learn.microsoft.com/dotnet/api/system.exception.serializeobjectstate>)
- object ([AkiraObject.html](#)) ([\(https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals\(system-object\)\)](https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals(system-object))
- VirtualItemAdapter < T > ([AkiraVirtualItemAdapter.html](#)) ([\(https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals\(system-object-system-object\)\)](https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals(system-object-system-object))
- object ([AkiraObject.html](#)) (<https://learn.microsoft.com/dotnet/api/system.object.gethashcode>)
- object ([AkiraMemberwiseClone.html](#)) (<https://learn.microsoft.com/dotnet/api/system.object.memberwiseclone>)
- object ([AkiraReferenceEquals.html](#)) (<https://learn.microsoft.com/dotnet/api/system.object.referenceequals>)

## Namespace

[AkiraNetwork](#) ([AkiraNetwork.html](#)).VirtualStorageLibrary ([AkiraNetwork.VirtualStorageLibrary.html](#))

([AkiraNetwork.VirtualStorageLibrary](#))

## Assembly

[VirtualStorageLibrary.dll](#)

[VirtualNodeContext](#)

## Syntax

[\(AkiraNetwork.VirtualStorageLibrary\)](#)

VirtualNodeExtensions

public abstract class VirtualException : Exception, ISerializable

VirtualNodeListConditions

([AkiraNetwork.VirtualStorageLibrary](#))

VirtualNodeName

([AkiraNetwork.VirtualStorageLibrary](#))

VirtualNodeNotFoundException

([AkiraNetwork.VirtualStorageLibrary](#))

## VirtualException()

[VirtualNodeType](#)

## Constructors

Initializes a new instance of the VirtualException (AkiraNetwork.VirtualStorageLibrary.VirtualException.html) class.

## Declaration

```
public VirtualException()
```



## VirtualException(string)

IVirtualWildcardMatcher

Initializes a new instance of the VirtualException (AkiraNetwork.VirtualStorageLibrary.VirtualException.html) class with a specified error message.

NotifyNodeDelegate

### Declaration

PatternMatch

(AkiraNetwork.VirtualStorageLibrary)

```
public VirtualException(string message)
```

VirtualCycleDetector

(AkiraNetwork.VirtualStorageLibrary)

### Parameters

VirtualDirectory

Type	Name	Description
(AkiraNetwork.VirtualStorageLibrary) VirtualDirectoryAdapter<T>		

string (https://learn.microsoft.com/dotnet/api/system.string)

message

The error message.

1.html)

VirtualException

(AkiraNetwork.VirtualStorageLibrary)

## VirtualException(string, Exception)

VirtualGroupCondition<T, Key>

Initializes a new instance of the VirtualException (AkiraNetwork.VirtualStorageLibrary.VirtualException.html) class with a specified error message and a reference to the inner exception that is the cause of this exception.

VirtualID

### Declaration

(AkiraNetwork.VirtualStorageLibrary)

VirtualItem

```
public VirtualException(string message, Exception innerException)
```

VirtualItemAdapter<T>

(AkiraNetwork.VirtualStorageLibrary)

### Parameters

1.html)

Type	Name	Description
(AkiraNetwork.VirtualStorageLibrary) VirtualItem<T>		

string

(AkiraNetwork.VirtualStorageLibrary)

message

The error message.

1.html)

(https://learn.microsoft.com/dotnet/api/system.string)

VirtualNode

### Exception

(AkiraNetwork.VirtualStorageLibrary)

### InnerException

(AkiraNetwork.VirtualStorageLibrary)

innerException

The exception that is the cause of this exception.

VirtualNodeExtensions

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeListConditions

(AkiraNetwork.VirtualStorageLibrary)

### ISerializable

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeNotImplementedException

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeType

## Implements

# Extension Methods

VirtualTextFormatter.GenerateSingleTableDebugText<T>(T)  
(AkiraNetwork.VirtualStorageLibrary.Utilities.VirtualTextFormatter.html#AkiraNetwork\_VirtualStorageLibrary\_Utils  
s\_VirtualTextFormatter\_GenerateSingleTableDebugText\_1\_\_0\_)

▼

1.html)  
IVirtualWildcardMatcher  
(AkiraNetwork.VirtualStorageLibrary  
NotifyNodeDelegate  
(AkiraNetwork.VirtualStorageLibrary  
PatternMatch  
(AkiraNetwork.VirtualStorageLibrary  
VirtualCycleDetector  
(AkiraNetwork.VirtualStorageLibrary  
VirtualDirectory  
(AkiraNetwork.VirtualStorageLibrary  
VirtualDirectoryAdapter<T>  
(AkiraNetwork.VirtualStorageLibrary  
1.html)  
VirtualException  
(AkiraNetwork.VirtualStorageLibrary  
VirtualGroupCondition<T, TKey>  
(AkiraNetwork.VirtualStorageLibrary  
2.html)  
VirtualID  
(AkiraNetwork.VirtualStorageLibrary  
VirtualItem  
(AkiraNetwork.VirtualStorageLibrary  
VirtualItemAdapter<T>  
(AkiraNetwork.VirtualStorageLibrary  
1.html)  
VirtualItem<T>  
(AkiraNetwork.VirtualStorageLibrary  
1.html)  
VirtualNode  
(AkiraNetwork.VirtualStorageLibrary  
VirtualNodeContext  
(AkiraNetwork.VirtualStorageLibrary  
VirtualNodeExtensions  
(AkiraNetwork.VirtualStorageLibrary  
VirtualNodeListConditions  
(AkiraNetwork.VirtualStorageLibrary  
VirtualnodeName  
(AkiraNetwork.VirtualStorageLibrary  
VirtualNodeNotFoundException  
(AkiraNetwork.VirtualStorageLibrary  
VirtualNodeType

# Class VirtualGroupCondition<T, TKey>

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

## Inheritance

(AkiraNetwork.VirtualStorageLibrary.VirtualWildcardMatcher)  
↳ object (https://learn.microsoft.com/dotnet/api/system.object)  
    ↳ NotifyNodeDelegate  
        ↳ VirtualGroupCondition<T, TKey>  
            ↳ (AkiraNetwork.VirtualStorageLibrary)

## Inherited Members

(AkiraNetwork.VirtualStorageLibrary)  
object.Equals(object) (https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals(system-object))  
VirtualCycleDetector  
(AkiraNetwork.VirtualStorageLibrary)  
object.Equals(object, object) (https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals(system-object-system-object))  
VirtualDirectory  
(AkiraNetwork.VirtualStorageLibrary)  
object.GetHashCode() (https://learn.microsoft.com/dotnet/api/system.object.gethashcode)  
VirtualDirectoryAdapter<T>  
object.GetType() (https://learn.microsoft.com/dotnet/api/system.object.gettype)  
(AkiraNetwork.VirtualStorageLibrary)  
object.MemberwiseClone() (https://learn.microsoft.com/dotnet/api/system.object.memberwiseclone)  
1.html  
object.ReferenceEquals(object, object) (https://learn.microsoft.com/dotnet/api/system.object.referenceequals)  
VirtualException  
object.ToString() (https://learn.microsoft.com/dotnet/api/system.object.tostring)  
(AkiraNetwork.VirtualStorageLibrary)

**Namespace:** AkiraNetwork (AkiraNetwork.html).VirtualStorageLibrary (AkiraNetwork.VirtualStorageLibrary.html)

**Assembly:** AkiraNetwork.VirtualStorageLibrary (AkiraNetwork.VirtualStorageLibrary.dll)

## Syntax

```
VirtualID  
(AkiraNetwork.VirtualStorageLibrary)  
public class VirtualGroupCondition<T, TKey>  
    VirtualItem  
(AkiraNetwork.VirtualStorageLibrary)
```

## Type Parameters

Name	Description
T	VirtualItem<T> The type of the entity to be grouped.
TKey	(AkiraNetwork.VirtualStorageLibrary) The type of the key used for grouping. 1.html

## Constructors

VirtualGroupCondition<Expression<Func<T, TKey>>, bool>  
(AkiraNetwork.VirtualStorageLibrary)  
VirtualNodeContext  
(AkiraNetwork.VirtualStorageLibrary)  
VirtualNodeExtensions  
(AkiraNetwork.VirtualStorageLibrary)

## VirtualGroupCondition<Expression<Func<T, TKey>>, bool>

VirtualNodeListConditions  
Initializes a new instance of the VirtualGroupCondition<T, TKey>  
(AkiraNetwork.VirtualStorageLibrary) class with the specified grouping property and order:  
VirtualNodeName  
(AkiraNetwork.VirtualStorageLibrary)

Declaration  
VirtualNodeNotFoundException  
(AkiraNetwork.VirtualStorageLibrary)  
VirtualNodeType  
(AkiraNetwork.VirtualStorageLibrary)

```
public VirtualGroupCondition(Expression<Func<T, TKey>> groupBy, bool ascending = true)
```

## Parameters

Type	Name	Description
Expression<Func<T, TKey>> ( <a href="https://learn.microsoft.com/dotnet/api/system.linq.expressions.expression">https://learn.microsoft.com/dotnet/api/system.linq.expressions.expression</a> ) Network.VirtualStorageLibrary<Func<T, TKey>> ( <a href="https://learn.microsoft.com/dotnet/api/system.func-2">https://learn.microsoft.com/dotnet/api/system.func-2</a> )<T, TKey> (AkiraNetwork.VirtualStorageLibrary)	groupBy	The property to group by.
bool ( <a href="https://learn.microsoft.com/dotnet/api/system.boolean">https://learn.microsoft.com/dotnet/api/system.boolean</a> ) (AkiraNetwork.VirtualStorageLibrary)	ascending	A value indicating whether the grouping order is ascending.

## Properties

VirtualItemAdapter<T>

(AkiraNetwork.VirtualStorageLibrary)

1.html)

### Ascending

(VirtualException)

(AkiraNetwork.VirtualStorageLibrary)

Gets or sets a value indicating whether the grouping order is ascending. True if the order is ascending; otherwise, false.

(AkiraNetwork.VirtualStorageLibrary)

### Declaration

VirtualID

public bool Ascending { get; set; }

VirtualItem

(AkiraNetwork.VirtualStorageLibrary)

### Property Value

VirtualItemAdapter<T>

(AkiraNetwork.VirtualStorageLibrary)

1.html)

bool

VirtualItem<T>

(<https://learn.microsoft.com/dotnet/api/system.boolean>)

(AkiraNetwork.VirtualStorageLibrary)

lean)

1.html)

VirtualNode

(AkiraNetwork.VirtualStorageLibrary)

### GroupBy

(VirtualNodeContext)

(AkiraNetwork.VirtualStorageLibrary)

Gets the `VirtualNodeContext` used for grouping.

(AkiraNetwork.VirtualStorageLibrary)

### Declaration

VirtualNodeListConditions

(AkiraNetwork.VirtualStorageLibrary)

public Expression<Func<T, TKey>> GroupBy { get; set; }

VirtualNodeName

(AkiraNetwork.VirtualStorageLibrary)

### Property Value

VirtualNodeNotFoundException

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary)

## Description

A boolean value indicating whether the grouping order is ascending.

Type	Description
Expression ( <a href="https://learn.microsoft.com/dotnet/api/system.linq.expressions.expression-1">https://learn.microsoft.com/dotnet/api/system.linq.expressions.expression-1</a> ) <Func ( <a href="https://learn.microsoft.com/dotnet/api/system.func-2">https://learn.microsoft.com/dotnet/api/system.func-2</a> )<T, TKey>>	An expression that specifies the property to use for grouping.

## Extension Methods

IVirtualWildcardMatcher	(AkiraNetwork.VirtualStorageLibrary)
VirtualNodeDelegate	GenerateSingleTableDebugText<T>(T) (AkiraNetwork.VirtualStorageLibrary)
VirtualTextFormatter	Utilities.VirtualTextFormatter.html#AkiraNetwork_VirtualStorageLibrary_Utilities_VirtualTextFormatter_GenerateSingleTableDebugText_1__0_
VirtualTextMatcher	(AkiraNetwork.VirtualStorageLibrary)
VirtualCycleDetector	
VirtualDirectory	(AkiraNetwork.VirtualStorageLibrary)
VirtualDirectoryAdapter<T>	(AkiraNetwork.VirtualStorageLibrary)
VirtualException	1.html
VirtualGroupCondition<T, TKey>	(AkiraNetwork.VirtualStorageLibrary)
VirtualItem	2.html
VirtualItemAdapter<T>	(AkiraNetwork.VirtualStorageLibrary)
VirtualItem<T>	1.html
VirtualNode	(AkiraNetwork.VirtualStorageLibrary)
VirtualNodeContext	(AkiraNetwork.VirtualStorageLibrary)
VirtualNodeExtensions	(AkiraNetwork.VirtualStorageLibrary)
VirtualNodeListConditions	(AkiraNetwork.VirtualStorageLibrary)
VirtualnodeName	(AkiraNetwork.VirtualStorageLibrary)
VirtualNodeNotFoundException	(AkiraNetwork.VirtualStorageLibrary)
VirtualNodeType	(AkiraNetwork.VirtualStorageLibrary)

# Struct VirtualID

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

(AkiraNetwork.VirtualStorageLibrary)

Implements

IEquatable<VirtualID> (<https://learn.microsoft.com/dotnet/api/system.iequatable-1>)

(AkiraNetwork.VirtualStorageLibrary.VirtualID.html)

(AkiraNetwork.VirtualStorageLibrary)

Inherited Members

VirtualCycleDetector

ValueObject.Equals(object) (<https://learn.microsoft.com/dotnet/api/system.valuetype.equals>)

ValueObject.GetHashCode() (<https://learn.microsoft.com/dotnet/api/system.valuetype.gethashcode>)

object.Equals(object) ([https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals\(system-object-system-object\)](https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals(system-object-system-object)))

VirtualDirectoryAdapter<T>

(AkiraNetwork.VirtualStorageLibrary)

object.GetType() (<https://learn.microsoft.com/dotnet/api/system.object.gettype>)

object.ReferenceEquals(object, object) (<https://learn.microsoft.com/dotnet/api/system.object.referenceequals>)

VirtualException

Namespace: AkiraNetwork (AkiraNetwork.html).VirtualStorageLibrary (AkiraNetwork.VirtualStorageLibrary.html)

(AkiraNetwork.VirtualStorageLibrary)

Assembly: VirtualStorageLibrary.dll

VirtualGroupCondition<T, TKey>

Syntax

(AkiraNetwork.VirtualStorageLibrary)

2.html)

public readonly record struct VirtualID : IEquatable<VirtualID>

(AkiraNetwork.VirtualStorageLibrary)

VirtualItem

(AkiraNetwork.VirtualStorageLibrary)

VirtualItemAdapter<T>

(AkiraNetwork.VirtualStorageLibrary)

1.html)

**VirtualID0**

VirtualItem<T>

(AkiraNetwork.VirtualStorageLibrary)

Initializes a new instance of the VirtualID (AkiraNetwork.VirtualStorageLibrary.VirtualID.html) structure, generating a unique identifier.

VirtualNode

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeContext

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeExtensions

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeListConditions

(AkiraNetwork.VirtualStorageLibrary)

Name

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeNotFoundException

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeType

Gets the unique identifier.

(AkiraNetwork.VirtualStorageLibrary)

Declaration

NodeFilter

```
public Guid ID { get; }
```

## Property Value

Type	Description
Guid ( <a href="https://learn.microsoft.com/dotnet/api/system.guid">https://learn.microsoft.com/dotnet/api/system.guid</a> ) AkiraNetwork.VirtualStorageLibrary /system/guid VirtualNodeDelegate	A Guid ( <a href="https://learn.microsoft.com/dotnet/api/system.guid">https://learn.microsoft.com/dotnet/api/system.guid</a> ) value representing the unique identifier.

## Methods

### `ToString()`

Returns a string representation of this instance's unique identifier.  
(<https://learn.microsoft.com/dotnet/api/AkiraNetwork.VirtualStorageLibrary.VirtualID.ToString>)

#### Declaration

```
public override string ToString()  
VirtualGroupCondition<T, TKey>  
(AkiraNetwork.VirtualStorageLibrary)
```

#### Returns

Type	Description
VirtualID ( <a href="https://learn.microsoft.com/dotnet/api/AkiraNetwork.VirtualStorageLibrary.VirtualID">https://learn.microsoft.com/dotnet/api/AkiraNetwork.VirtualStorageLibrary.VirtualID</a> ) string ( <a href="https://learn.microsoft.com/dotnet/api/system.string">https://learn.microsoft.com/dotnet/api/system.string</a> ) (AkiraNetwork.VirtualStorageLibrary)	A string that represents the unique identifier.

#### Overrides

`Value Type.ToString()` (<https://learn.microsoft.com/dotnet/api/system.valuetype.tostring>)  
(<https://learn.microsoft.com/dotnet/api/AkiraNetwork.VirtualStorageLibrary.VirtualGroupCondition.ToString>)

#### Implements

`IEquatable<T>` (<https://learn.microsoft.com/dotnet/api/system.iequatable-1>)  
VirtualNode  
(<https://learn.microsoft.com/dotnet/api/AkiraNetwork.VirtualStorageLibrary.VirtualNode>)

## Extension Methods

`VirtualTextFormatter.GenerateSingleTableDebugText<T>(T)`  
([https://learn.microsoft.com/dotnet/api/AkiraNetwork.VirtualStorageLibrary.Utilities.VirtualTextFormatter.html#AkiraNetwork\\_VirtualStorageLibrary\\_Utilsitie](https://learn.microsoft.com/dotnet/api/AkiraNetwork.VirtualStorageLibrary.Utilities.VirtualTextFormatter.html#AkiraNetwork_VirtualStorageLibrary_Utilsitie))  
`s_VirtualTextFormatter.GenerateSingleTableDebugText_1__0_`  
VirtualNodeListConditions  
(<https://learn.microsoft.com/dotnet/api/AkiraNetwork.VirtualStorageLibrary.VirtualNodeListConditions>)  
VirtualStorageLibrary  
VirtualNodeName  
(<https://learn.microsoft.com/dotnet/api/AkiraNetwork.VirtualStorageLibrary.VirtualNodeName>)  
VirtualNodeNotFoundException  
(<https://learn.microsoft.com/dotnet/api/AkiraNetwork.VirtualStorageLibrary.VirtualNodeNotFoundException>)  
VirtualNodeType  
(<https://learn.microsoft.com/dotnet/api/AkiraNetwork.VirtualStorageLibrary.VirtualNodeType>)  
VirtualNodeTypeFilter  
(<https://learn.microsoft.com/dotnet/api/AkiraNetwork.VirtualStorageLibrary.VirtualNodeTypeFilter>)

# Class VirtualItem

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](#)).

([AkiraNetwork.VirtualStorageLibrary](#))

**Inheritance**

([AkiraNetwork.VirtualStorageLibrary](#))

↳ `object` (<https://learn.microsoft.com/dotnet/api/system.object>)

↳ `VirtualNode` ([AkiraNetwork.VirtualStorageLibrary.VirtualNode.html](#))

↳ `VirtualItem`

↳ `VirtualCycledDetector`

↳ `VirtualItem` ([AkiraNetwork.VirtualStorageLibrary.VirtualItem-1.html](#)) <`T`>

**Implements**

([AkiraNetwork.VirtualStorageLibrary](#))

`IVirtualDeepCloneable` ([AkiraNetwork.VirtualStorageLibrary.IVirtualDeepCloneable-1.html](#)) <`VirtualNode`

↳ `VirtualDirectoryAdapter<T>`

([AkiraNetwork.VirtualStorageLibrary.VirtualNode.html](#)) >

**Inherited Members**

1.html)

`VirtualException`

([AkiraNetwork.VirtualStorageLibrary](#))

`VirtualGroupCondition<T, TKey>`

([AkiraNetwork.VirtualStorageLibrary](#))

`VirtualNode` ([AkiraNetwork.VirtualStorageLibrary.VirtualNode.html](#)) #[AkiraNetwork\\_VirtualStorageLibrary\\_VirtualNode\\_CreatedDate](#) 2.html)

`VirtualNodeUpdatedDate`

([AkiraNetwork.VirtualStorageLibrary](#))

`DateVirtualItem`

([AkiraNetwork.VirtualStorageLibrary](#))

`VirtualNodeAdapter<T>`

([AkiraNetwork.VirtualStorageLibrary](#))

↳ `VirtualNodeTypeFilter`

([AkiraNetwork.VirtualStorageLibrary](#))

`VirtualNode.VID`

([AkiraNetwork.VirtualStorageLibrary](#))

`VirtualNodeUpdateVirtualNode`

([AkiraNetwork.VirtualStorageLibrary](#))

`VirtualNodeUpdateVirtualNodeUpdateAll`

([AkiraNetwork.VirtualStorageLibrary](#))

`VirtualNodeVirtualStorageLibrary_VirtualNode`

([AkiraNetwork.VirtualStorageLibrary](#))

`VirtualNodeIsReferencedInStorage`

([AkiraNetwork.VirtualStorageLibrary](#))

`VirtualNodeExtensions` ([\[https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals\\(system-object-system-object\\)\]\(#\)](https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals(system-object-system-object)))

([AkiraNetwork.VirtualStorageLibrary](#))

`VirtualNodeObjectEqualityComparer` ([\[https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals\\(system-object-system-object\\)\]\(#\)](https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals(system-object-system-object)))

([AkiraNetwork.VirtualStorageLibrary](#))

`VirtualNodeHashCode` ([\[https://learn.microsoft.com/dotnet/api/system.object.gethashcode\]\(#\)](https://learn.microsoft.com/dotnet/api/system.object.gethashcode))

([AkiraNetwork.VirtualStorageLibrary](#))

`VirtualNodeGetType` ([\[https://learn.microsoft.com/dotnet/api/system.object.gettype\]\(#\)](https://learn.microsoft.com/dotnet/api/system.object.gettype))

([AkiraNetwork.VirtualStorageLibrary](#))

`VirtualNodeGetMemberwiseClone` ([\[https://learn.microsoft.com/dotnet/api/system.object.memberwiseclone\]\(#\)](https://learn.microsoft.com/dotnet/api/system.object.memberwiseclone))

([AkiraNetwork.VirtualStorageLibrary](#))

`VirtualNodeReferenceEquals` ([\[https://learn.microsoft.com/dotnet/api/system.object.referenceeequals\]\(#\)](https://learn.microsoft.com/dotnet/api/system.object.referenceeequals))

([AkiraNetwork.VirtualStorageLibrary](#))

`VirtualNodeToString` ([\[https://learn.microsoft.com/dotnet/api/system.object.tostring\]\(#\)](https://learn.microsoft.com/dotnet/api/system.object.tostring))

([AkiraNetwork.VirtualStorageLibrary](#))

**Namespace:** [AkiraNetwork](#) ([AkiraNetwork.html](#)).VirtualStorageLibrary ([AkiraNetwork.VirtualStorageLibrary.html](#))

`VirtualNodeTypeFilter`

## Syntax

```
public abstract class VirtualItem : VirtualNode, IVirtualDeepCloneable<VirtualNode>
```



## Constructors

(AkiraNetwork.VirtualStorageLibrary)

NotifyNodeDelegate

### **VirtualItem(VirtualNodeName)**

PatternMatch

Initializes a new instance of the VirtualItem (AkiraNetwork.VirtualStorageLibrary.VirtualItem.html) class.

(AkiraNetwork.VirtualStorageLibrary)

VirtualCycleDetector

(AkiraNetwork.VirtualStorageLibrary)

VirtualDirectory

```
protected VirtualItem(VirtualNodeName name)
```

(AkiraNetwork.VirtualStorageLibrary)

VirtualDirectoryAdapter<T>

(AkiraNetwork.VirtualStorageLibrary)

1.html

Type VirtualException

(AkiraNetwork.VirtualStorageLibrary)

VirtualGroupCondition<T, TKey>

(AkiraNetwork.VirtualStorageLibrary)

2.html

### **VirtualItem(VirtualNodeName, DateTime)**

(AkiraNetwork.VirtualStorageLibrary)

Initializes a new instance of the VirtualItem (AkiraNetwork.VirtualStorageLibrary.VirtualItem.html) class.

VirtualItem

(AkiraNetwork.VirtualStorageLibrary)

VirtualItemAdapter<T>

```
protected VirtualItem(VirtualNodeName name, DateTime createdDate)
```

1.html

VirtualItem<T>

(AkiraNetwork.VirtualStorageLibrary)

Type 1.html

VirtualNode

VirtualNodeName

(AkiraNetwork.VirtualStorageLibrary)

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)

VirtualNodeContext

(AkiraNetwork.VirtualStorageLibrary)

DateOnly (<https://learn.microsoft.com/dotnet/api/system.datetime>)

VirtualNodeExtensions

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeListConditions

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeName

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeNotFoundException

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeTypeFilter

### **VirtualItem(VirtualNodeName, DateTime, DateTime)**

(AkiraNetwork.VirtualStorageLibrary)

Initializes a new instance of the VirtualItem (AkiraNetwork.VirtualStorageLibrary.VirtualItem.html) class.

VirtualNodeNotFoundException

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeTypeFilter

```
protected VirtualItem(VirtualNodeName name, DateTime createdDate, DateTime updatedDate)
```

## Parameters

Type		Name	Description
VirtualNodeName (AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)		<i>name</i>	The name of the item.
NotifyNodeDelegate DateTime (AkiraNetwork.VirtualStorageLibrary (https://learn.microsoft.com/dotnet/api/system.datetime) PatternMatch		<i>createdDate</i>	The creation date of the item.
DateTime (AkiraNetwork.VirtualStorageLibrary (https://learn.microsoft.com/dotnet/api/system.datetime)		<i>updatedDate</i>	The last update date of the item.

## Methods

**DeepClone(bool)**  
(AkiraNetwork.VirtualStorageLibrary

Creates a deep copy of the item. If the data of type T held by the item implements the IVirtualDeepCloneable<T> (AkiraNetwork.VirtualStorageLibrary.VirtualDeepCloneable-1.html) interface, the DeepClone method of the data type is called.

VirtualID  
**Declaration:**

(AkiraNetwork.VirtualStorageLibrary

VirtualItem

```
public override abstract VirtualNode DeepClone(bool recursive = false)
```

VirtualItemAdapter<T>

(AkiraNetwork.VirtualStorageLibrary

## Parameters

Type		Name	Description
VirtualItem<T> bool (AkiraNetwork.VirtualStorageLibrary (https://learn.microsoft.com/dotnet/api/system.boolean)		<i>recursive</i>	In the case of items, this parameter is ignored.

## Returns

VirtualNodeContext

Type	Description
AkiraNetwork.VirtualStorageLibrary	

VirtualNodeExtensions

VirtualNode  
(AkiraNetwork.VirtualStorageLibrary

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeListConditions

INode.html)

(AkiraNetwork.VirtualStorageLibrary

VirtualNodeName

(AkiraNetwork.VirtualStorageLibrary

VirtualNodeDeepClone(bool)

VirtualNodeDeepCloneException  
(AkiraNetwork.VirtualStorageLibrary

(AkiraNetwork.VirtualStorageLibrary.VirtualNode.html#AkiraNetwork\_VirtualStorageLibrary\_VirtualNode\_DeepClon

e\_System\_Boolean)

VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary

VirtualNodeTypeFilter

# Implements

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

## Extension Methods

VirtualTextFormatter.GenerateSingleTableDebugText<T>(T)  
(AkiraNetwork.VirtualStorageLibrary.Utilities.VirtualTextFormatter.html#AkiraNetwork\_VirtualStorageLibrary\_Utilsilie  
s\_VirtualTextFormatter\_GenerateSingleTableDebugText\_1\_\_0\_)  
NotifyNodeDelegate  
VirtualNodeExtensions.ResolveNodeType(VirtualNode)  
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeExtensions.html#AkiraNetwork\_VirtualStorageLibrary\_VirtualNode  
Extensions\_ResolveNodeType\_AkiraNetwork\_VirtualStorageLibrary\_VirtualNode\_)  
(AkiraNetwork.VirtualStorageLibrar  
VirtualCycleDetector  
(AkiraNetwork.VirtualStorageLibrary.VirtualCycleDetector.html#AkiraNetwork\_VirtualStorageLibrary\_VirtualCyc  
eDetector)  
VirtualDirectory  
(AkiraNetwork.VirtualStorageLibrary.VirtualDirectory.html#AkiraNetwork\_VirtualStorageLibrary\_VirtualDir  
ectory)  
VirtualDirectoryAdapter<T>  
(AkiraNetwork.VirtualStorageLibrary.VirtualDirectoryAdapter.html#AkiraNetwork\_VirtualStorageLibrary\_Virt  
ualDirectoryAdapter\_1.html)  
VirtualException  
(AkiraNetwork.VirtualStorageLibrary.VirtualException.html#AkiraNetwork\_VirtualStorageLibrary\_Virt  
ualException)  
VirtualGroupCondition<T, TKey>  
(AkiraNetwork.VirtualStorageLibrary.VirtualGroupCondition.html#AkiraNetwork\_VirtualStorageLibrary\_Virt  
ualGroupCondition\_2.html)  
VirtualID  
(AkiraNetwork.VirtualStorageLibrary.VirtualID.html#AkiraNetwork\_VirtualStorageLibrary\_Virt  
ualID)  
VirtualItem  
(AkiraNetwork.VirtualStorageLibrary.VirtualItem.html#AkiraNetwork\_VirtualStorageLibrary\_Virt  
ualItem)  
VirtualItemAdapter<T>  
(AkiraNetwork.VirtualStorageLibrary.VirtualItemAdapter.html#AkiraNetwork\_VirtualStorageLibrary\_Virt  
ualItemAdapter\_1.html)  
VirtualItem<T>  
(AkiraNetwork.VirtualStorageLibrary.VirtualItemT.html#AkiraNetwork\_VirtualStorageLibrary\_Virt  
ualItemT\_1.html)  
VirtualNode  
(AkiraNetwork.VirtualStorageLibrary.VirtualNode.html#AkiraNetwork\_VirtualStorageLibrary\_Virt  
ualNode)  
VirtualNodeContext  
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeContext.html#AkiraNetwork\_VirtualStorageLibrary\_Virt  
ualNodeContext)  
VirtualNodeExtensions  
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeExtensions.html#AkiraNetwork\_VirtualStorageLibrary\_Virt  
ualNodeExtensions)  
VirtualNodeListConditions  
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeListConditions.html#AkiraNetwork\_VirtualStorageLibrary\_V  
irtualNodeListConditions)  
VirtualnodeName  
(AkiraNetwork.VirtualStorageLibrary.VirtualnodeName.html#AkiraNetwork\_VirtualStorageLibrary\_Virt  
ualnodeName)  
VirtualNodeNotFoundException  
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeNotFoundException.html#AkiraNetwork\_VirtualStorageLibrary\_V  
irtualNodeNotFoundException)  
VirtualNodeType  
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeType.html#AkiraNetwork\_VirtualStorageLibrary\_Virt  
ualNodeType)  
VirtualNodeTypeFilter

# Class VirtualItemAdapter<T>

## Inheritance

↳ [Object](#) (<https://learn.microsoft.com/dotnet/api/system.object>)  
↳ [VirtualItemAdapter<T>](#)  
    (AkiraNetwork.VirtualStorageLibrary)

## Inherited Members

NotifyNodeDelegate

object.Equals(object) ([https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals\(system-object-object\)](https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals(system-object-object)))  
PatternMatch  
(AkiraNetwork.VirtualStorageLibrary)  
object.Equals(object) ([https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals\(system-object-object\)](https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals(system-object-object)))  
VirtualCycleDetector  
(AkiraNetwork.VirtualStorageLibrary)  
object.GetHashCode() (<https://learn.microsoft.com/dotnet/api/system.object.gethashcode>)  
object.GetType() (<https://learn.microsoft.com/dotnet/api/system.object.gettype>)  
object.MemberwiseClone()  
(AkiraNetwork.VirtualStorageLibrary)  
VirtualDirectoryAdapter<T>  
object.ReferenceEquals(object1, object2) (<https://learn.microsoft.com/dotnet/api/system.object.referenceequals>)  
object.ToString() (<https://learn.microsoft.com/dotnet/api/system.object.tostring>)  
1.html)

**Namespace:** AkiraNetwork (AkiraNetwork.html).VirtualStorageLibrary (AkiraNetwork.VirtualStorageLibrary.html)  
VirtualException

**Assembly:** VirtualStorageLibrary.dll

**Syntax:** VirtualGroupCondition<T, TKey>

(AkiraNetwork.VirtualStorageLibrary)

public class VirtualItemAdapter<T>

    VirtualID

(AkiraNetwork.VirtualStorageLibrary)

**Type Parameters**

VirtualItem

Name	Description
VirtualItemAdapter<T>	(AkiraNetwork.VirtualStorageLibrary)
T	(AkiraNetwork.VirtualStorageLibrary)

1.html)

VirtualItem<T>

## Constructors

(AkiraNetwork.VirtualStorageLibrary)

1.html)

VirtualNode

## VirtualItemAdapter(VirtualStorage<T>)

### Declaration

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeExtensions

public VirtualItemAdapter(VirtualStorage<T> storage)  
(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeListConditions

(AkiraNetwork.VirtualStorageLibrary)

Type	Name	Description
VirtualNodeName (AkiraNetwork.VirtualStorageLibrary)		
VirtualNodeNameFilter (AkiraNetwork.VirtualStorageLibrary.VirtualStorage-1.html)<T> (AkiraNetwork.VirtualStorageLibrary)	storage	

VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeTypeFilter

# Properties

## this[VirtualPath, bool]

### Declaration

```
public VirtualItem<T> this[VirtualPath path, bool followLinks = true] { get; set; }
```

(AkiraNetwork.VirtualStorageLibrary)

NotifyNodeDelegate

### Parameters

(AkiraNetwork.VirtualStorageLibrary)

#### Type

PatternMatch

(AkiraNetwork.VirtualStorageLibrary)

#### VirtualPath

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

#### VirtualCycleDetector

(AkiraNetwork.VirtualStorageLibrary)

#### bool

(https://learn.microsoft.com/dotnet/api/system.boolean)

VirtualDirectory

(AkiraNetwork.VirtualStorageLibrary)

#### Property Value

VirtualDirectoryAdapter<T>

#### Type

AkiraNetwork.VirtualStorageLibrary

1.html

#### VirtualItem

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

VirtualException

(AkiraNetwork.VirtualStorageLibrary)

#### VirtualGroupCondition<T, TKey>

(AkiraNetwork.VirtualStorageLibrary)

#### VirtualID

(AkiraNetwork.VirtualStorageLibrary.Utilities.VirtualTextFormatter.html#AkiraNetwork\_VirtualStorageLibrary\_Utils

#### s\_VirtualTextFormatter\_GenerateSingleTableDebugText\_1\_\_0\_

VirtualItem

(AkiraNetwork.VirtualStorageLibrary)

VirtualItemAdapter<T>

(AkiraNetwork.VirtualStorageLibrary)

1.html

VirtualItem<T>

(AkiraNetwork.VirtualStorageLibrary)

1.html

VirtualNode

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeContext

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeExtensions

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeListConditions

(AkiraNetwork.VirtualStorageLibrary)

VirtualnodeName

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeNotFoundException

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeTypeFilter

## Extension Methods

# Class VirtualItem<T>

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.

(AkiraNetwork.VirtualStorageLibrary)  
Virtual items ensure that the actual data is distinctly present. If the user-defined type T implements the IVirtualNotifyNodeDelegate  
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.  
(AkiraNetwork.VirtualStorageLibrary)

VirtualCycleDetector

Inheritance (AkiraNetwork.VirtualStorageLibrary)

↳ object (<https://learn.microsoft.com/dotnet/api/system.object>)  
↳ VirtualNode (AkiraNetwork.VirtualStorageLibrary.VirtualNode.html)  
↳ VirtualDirectory (AkiraNetwork.VirtualStorageLibrary.VirtualItem.html)  
(AkiraNetwork.VirtualStorageLibrary)  
↳ VirtualItem<T>  
1.html)

Implements

VirtualException  
IVirtualDeepCloneable (AkiraNetwork.VirtualStorageLibrary.IVirtualDeepCloneable-1.html) <VirtualNode  
(AkiraNetwork.VirtualStorageLibrary.VirtualNode.html)>  
IDisposable (<https://learn.microsoft.com/dotnet/api/system.idisposable>)  
2.html)

Inherited Members

VirtualID  
VirtualNode.Name  
(AkiraNetwork.VirtualStorageLibrary)  
(AkiraNetwork.VirtualStorageLibrary.VirtualNode.html#AkiraNetwork\_VirtualStorageLibrary\_VirtualNode\_Name)  
VirtualItem  
VirtualNode.CreatedDate  
(AkiraNetwork.VirtualStorageLibrary)  
(AkiraNetwork.VirtualStorageLibrary.VirtualNode.html#AkiraNetwork\_VirtualStorageLibrary\_VirtualNode\_CreatedD  
ate)  
VirtualNode.UpdatedDate  
VirtualItem<T>  
(AkiraNetwork.VirtualStorageLibrary.VirtualNode.html#AkiraNetwork\_VirtualStorageLibrary\_VirtualNode\_Updated  
Date)  
VirtualNode.VID  
VirtualNode  
VirtualNode.IsReferencedInStorage  
VirtualNodeContext  
(AkiraNetwork.VirtualStorageLibrary)  
object.Equals(object) ([https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals\(system-object\)](https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals(system-object)))  
(AkiraNetwork.VirtualStorageLibrary)  
object.Equals(object, object) ([https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals\(system-object-system-object\)](https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals(system-object-system-object)))  
(AkiraNetwork.VirtualStorageLibrary)  
object.GetHashCode() (<https://learn.microsoft.com/dotnet/api/system.object.gethashcode>)  
VirtualNodeName  
VirtualNode  
object.MemberwiseClone() (<https://learn.microsoft.com/dotnet/api/system.object.memberwiseclone>)  
VirtualNodeNotFoundException  
object.ReferenceEquals(object, object) (<https://learn.microsoft.com/dotnet/api/system.object.referenceequals>)  
(AkiraNetwork.VirtualStorageLibrary)

Namespace AkiraNetwork (AkiraNetwork.html).VirtualStorageLibrary (AkiraNetwork.VirtualStorageLibrary.html)

Assembly AkiraNetwork.VirtualStorageLibrary

VirtualNodeTypeFilter

## Syntax

```
public class VirtualItem<T> : VirtualItem, IVirtualDeepCloneable<VirtualNode>, IDisposable
```

### Type Parameters

Name	Description
T	(AkiraNetwork.VirtualStorageLibrary) The user-defined type library can be freely specified by the user of this library.

NotifyNodeDelegate

(AkiraNetwork.VirtualStorageLibrary)

PatternMatch

(AkiraNetwork.VirtualStorageLibrary)

VirtualCycleDetector

(AkiraNetwork.VirtualStorageLibrary)

VirtualDirectory

Initializes a new instance of the VirtualItem<T> (AkiraNetwork.VirtualStorageLibrary.VirtualItem-1.html) class. The node name is automatically generated by the GenerateNodeName(string)

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html#AkiraNetwork\_VirtualStorageLibrary\_VirtualNodeName\_GenerateNodeName\_System\_String\_) method.

VirtualException

Declaration

(AkiraNetwork.VirtualStorageLibrary)

VirtualGroupCondition<T, TKey>

public VirtualItem()

(AkiraNetwork.VirtualStorageLibrary)

2.html)

VirtualID

(AkiraNetwork.VirtualStorageLibrary)

### VirtualItem(VirtualNodeName)

VirtualItem

Initializes a new instance of the VirtualItem<T> (AkiraNetwork.VirtualStorageLibrary.VirtualItem-1.html) class.

VirtualItemAdapter<T>

Declaration

(AkiraNetwork.VirtualStorageLibrary)

1.html)

public VirtualItem(VirtualNodeName name)

(AkiraNetwork.VirtualStorageLibrary)

1.html)

### Parameters

VirtualNode

Type

(AkiraNetwork.VirtualStorageLibrary)

Name

Description

VirtualNodeContext

VirtualNodeName (AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)

name

The name of the item.

VirtualNodeExtensions

(AkiraNetwork.VirtualStorageLibrary)

### VirtualItem(VirtualNodeName, T?)

(AkiraNetwork.VirtualStorageLibrary)

Initializes a new instance of the VirtualItem<T> (AkiraNetwork.VirtualStorageLibrary.VirtualItem-1.html) class.

VirtualNodeName

Declaration

(AkiraNetwork.VirtualStorageLibrary)

VirtualNotFoundException

(AkiraNetwork.VirtualStorageLibrary)

public VirtualItem(VirtualNodeName name, T? itemData)

VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeTypeFilter

Type	Name	Description
VirtualNodeName (AkiraNetwork.VirtualStorageLibrary.VirtualNo deName.html)	<i>name</i>	The name of the item.
T <small>▼</small>	<i>itemData</i>	The data associated with this item. It is user-defined and can hold any type of data.

## (AkiraNetwork.VirtualStorageLibrary)

## NotifyNodeDelegate

(AkiraNetwork VirtualStorage) library

PatternMatch

**VirtualItem(VirtualNodeName, T?, DateTime)**

Initial ([AkiraNetworkVirtualStorageLibrary](#)) item <T> ([AkiraNetwork.VirtualStorage.library.VirtualItem-1.html](#)) class.

## VirtualCycleDetector

Virtual Declaration

AkiraNetwork.VirtualStorageLibrary

## VirtualDirectory

punkt & network virtual storage node name, T? itemData, DateTime createdDate)

## VirtualDirectoryAdapter<T>

By [AkiraN](#)

ameters  
1.html)

Type	Name	Description
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName<T, TKey>)(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)	<i>name</i>	The name of the item.
T VirtualID (AkiraNetwork.VirtualStorageLibrary)	<i>itemData</i>	The data associated with this item. It is user-defined and can hold any type of data.
VirtualItem DateTime (https://learn.microsoft.com/dotnet/api/system.datetime) VirtualItemAdapter<T> (AkiraNetwork.VirtualStorageLibrary)	<i>createdDate</i>	The creation date of the item.

1.html)

# Virtuall

**VirtualItem(VirtualNodeName, T?, DateTime, DateTime)**

Initializes a

## VirtualNode instance

# (AkiraNetwork.VirtualStorageLibrary) Declaration

```
VirtualNodeContext  
(AkiraNetwork.VirtualStorageLibrar  
public VirtualItem(VirtualNodeName name, T? itemData, DateTime createdDate, DateTime updated  
_VirtualNodeExtensions
```

(AkiraNetwork VirtualStorage Library)

### **VirtualNodeListConditions**

Virtual

(ALTER) Network.VirtualStorageLibrary

Type	VirtualNodeName (AkiraNetwork.VirtualStorageLibrary)	Name	Description
Virtu	VirtualNodeNotFoundException (AkiraNetwork.VirtualStorageLibrary)	name	The name of the item.
odeName	VirtualNodeType (AkiraNetwork.VirtualStorageLibrary)		
	VirtualNodeTypeFilter		



# Methods

## DeepClone(bool)

Creates a deep clone of this item.

### Declaration

```
(AkiraNetwork.VirtualStorageLibrary.  
VirtualNodeDelegate VirtualNode DeepClone(bool recursive = false))
```

### Parameters

Type	Name	Description
virtualCycleDetector		
bool	recursive	If true, perform a deep clone recursively.

VirtualDirectoryAdapter<T>

Returns  
(AkiraNetwork.VirtualStorageLibrary.

Type	Description
VirtualException	A deep clone of the current item as a VirtualNode (AkiraNetwork.VirtualStorageLibrary.VirtualNode.html).

### Overrides

VirtualDeepClone(bool)  
(AkiraNetwork.VirtualStorageLibrary.VirtualItem.html#AkiraNetwork\_VirtualStorageLibrary\_VirtualItem\_DeepClone\_1.html)

VirtualItemAdapter<T>

(AkiraNetwork.VirtualStorageLibrary.

## Dispose()

VirtualItem<T>  
Releases the resources used by the VirtualItem<T> (AkiraNetwork.VirtualStorageLibrary.VirtualItem-1.html).  
(AkiraNetwork.VirtualStorageLibrary.

### Declaration

```
VirtualNode  
public void Dispose()  
VirtualNodeContext  
(AkiraNetwork.VirtualStorageLibrary.  
VirtualNodeExtensions
```

## Dispose(bool)

VirtualNodeListConditions  
Releases the unmanaged resources used by the VirtualItem<T> (AkiraNetwork.VirtualStorageLibrary.VirtualItem-1.html) and optionally releases the managed resources.

(AkiraNetwork.VirtualStorageLibrary.  
VirtualNodeNotFoundException  
(AkiraNetwork.VirtualStorageLibrary.

```
protected virtual void Dispose(bool disposing)  
VirtualNodeType
```

(AkiraNetwork.VirtualStorageLibrary.  
VirtualNodeTypeFilter

### Parameters

Type	Name	Description
bool ( <a href="https://learn.microsoft.com/dotnet/api/system.boolean">https://learn.microsoft.com/dotnet/api/system.boolean</a> )	<i>disposing</i>	true to release both managed and unmanaged resources; false to release only unmanaged resources.

▼

## ~VirtualItem()

NotifyNodeDelegate  
Finalizer for the VirtualItem<T> ([AkiraNetwork.VirtualStorageLibrary.VirtualItem-1.html](#)) class.  
([AkiraNetwork.VirtualStorageLibrary](#))

### Declaration

```
(AkiraNetwork.VirtualStorageLibrary)
VirtualCycleDetector
protected VirtualItem()
(AkiraNetwork.VirtualStorageLibrary)
VirtualDirectory
(AkiraNetwork.VirtualStorageLibrary)
VirtualDirectoryAdapter<T>
(AkiraNetwork.VirtualStorageLibrary)
```

## Tostring()

VirtualException  
Declaration  
([AkiraNetwork.VirtualStorageLibrary](#))

VirtualGroupCondition<T, TKey>
public override string ToString()
(AkiraNetwork.VirtualStorageLibrary)
2.html)

### Returns

VirtualID  
([AkiraNetwork.VirtualStorageLibrary](#))
Type  
VirtualItem  
([AkiraNetwork.VirtualStorageLibrary](#))

### Description

A string that represents the current object.

Overrides  
1.html)
object.ToString() ([https://learn.microsoft.com/dotnet/api/system.object.tostring](#))

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

## Update(VirtualNode)

VirtualNode

Updates the specified node's data.

VirtualNodeContext  
Declaration  
([AkiraNetwork.VirtualStorageLibrary](#))

```
VirtualNodeExtensions
public override void Update(VirtualNode node)
(AkiraNetwork.VirtualStorageLibrary)
VirtualNodeListConditions
```

Parameters  
VirtualNodeName

Type  
([AkiraNetwork.VirtualStorageLibrary](#))

VirtualNodeNotFoundException  
([AkiraNetwork.VirtualStorageLibrary](#))

VirtualNodeType  
([AkiraNetwork.VirtualStorageLibrary](#))

Overrides  
VirtualNodeTypeFilter

### Name

### Description

The node containing the data to update.

`VirtualNode.Update(VirtualNode)  
(AkiraNetwork.VirtualStorageLibrary.VirtualNode.html#AkiraNetwork_VirtualStorageLibrary_VirtualNode_Update_AkiraNetwork_VirtualStorageLibrary_VirtualNode_)`

# Exceptions

Type		Condition
ArgumentException ( <a href="https://learn.microsoft.com/dotnet/api/system.argumentexception">https://learn.microsoft.com/dotnet/api/system.argumentexception</a> )	AkiraNetwork.VirtualStorageLibrary.NotifyNodeDelegate	Thrown if the specified node is not of the same type as the current item.

# Operators

## VirtualCycleDetector

(AkiraNetwork.VirtualStorageLibrary)

**implicit operator VirtualItem<T>(VirtualNodeName)**

(AkiraNetwork.VirtualStorageLibrary)

Implicitly converts the specified node name to a VirtualItem<T> (AkiraNetwork.VirtualStorageLibrary.VirtualItem-1.html) : Network\VirtualStorageLib

(AkiraNetwork.VirtualStorageLibrary)

## Declaration

## VirtualException

```
(AkiraNetwork.VirtualStorageLibrary) public static implicit operator VirtualItem<T>(VirtualNodeName name)  
VirtualGroupCondition<T, TKey>
```

(AkiraNetwork.VirtualStorageLibrary)

## Parameters

Type	VirtualID (AkiraNetwork.VirtualStorageLibrary)	Name	Description
VirtualNodeName	(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)	<i>name</i>	The name of the node.

(AkiraNetwork.virtual)

Type	Description
(AkiraNetwork.VirtualStorageLibrary.VirtualItem<T>.html)	A
(AkiraNetwork.VirtualStorageLibrary.VirtualItem<T>.VirtualNode)	VirtualItem<T> (AkiraNetwork.VirtualStorageLibrary.VirtualItem-1.html)
<T>VirtualNode	initialized with the specified node name.

**implicit operator VirtualItem<T>(VirtualNodeName nodeName, T? itemData)**

Converts a tuple (nodeName, itemData) to a VirtualItem<T> (AkiraNetwork.VirtualStorageLibrary.VirtualItem-1.htm)

Declaration NodeName

## Declaration of Node Name

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeNotFoundException) - AkiraNetwork.VirtualStorageLibrary

**VirtualNodeType**  
(AkiraNetwork VirtualStorageLibrary)

### Parameters

#### **virtualNodeTypeFilter**

Type	Name	Description
(VirtualNodeName (AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html) nodeName (https://learn.microsoft.com/dotnet/api/system.valuetuple-akiranetwork.virtualstoragelibrary.virtualnodename,-0-.nodename) , T itemData (https://learn.microsoft.com/dotnet/api/system.valuetuple-akiranetwork.virtualstoragelibrary.virtualnodename,-0-.itemdata) ) (AkiraNetwork.VirtualStorageLibrary.NotifyNodeDelegate)	tuple	A tuple containing the node name and item data.

Returns  
PatternMatch

Type	Description
AkiraNetwork.VirtualStorageLibrary.VirtualCycleDetector VirtualItem (AkiraNetwork.VirtualStorageLibrary.VirtualItem-1.html) (AkiraNetwork.VirtualStorageLibrary.VirtualDirectory Item-1.html) <T> (AkiraNetwork.VirtualStorageLibrary.VirtualDirectoryAdapter<T> (AkiraNetwork.VirtualStorageLibrary.VirtualItem-1.html)	A VirtualItem<T> (AkiraNetwork.VirtualStorageLibrary.VirtualItem-1.html) initialized with the specified values.

## implicit operator VirtualItem<T>(T?)

(AkiraNetwork.VirtualStorageLibrary.VirtualGroupCondition<T, TKey>  
Implicitly converts the specified data of type T to a VirtualItem<T>  
(AkiraNetwork.VirtualStorageLibrary.VirtualItem-1.html). The node name is automatically generated by the  
GenerateNodeName(string  
2.html)  
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html#AkiraNetwork\_VirtualStorageLibrary\_VirtualNodeName\_GenerateNodeName\_System\_String\_) method.  
(AkiraNetwork.VirtualStorageLibrary.VirtualItem-1.html)

Declaration  
(AkiraNetwork.VirtualStorageLibrary.VirtualItem  
VirtualItemAdapter<T>

```
public static implicit operator VirtualItem<T>(T? itemData)
(AkiraNetwork.VirtualStorageLibrary.VirtualItem-1.html)
```

Parameters  
item<T>

Type	Name	Description
T	VirtualNode	The data associated with this item. It is user-defined and can hold any type of data. (AkiraNetwork.VirtualStorageLibrary.VirtualItem-1.html)

Returns  
VirtualNodeContext

Type	Description
VirtualItem (AkiraNetwork.VirtualStorageLibrary.VirtualNodeContext<T> (AkiraNetwork.VirtualStorageLibrary.VirtualItem-1.html) <T>VirtualNodeName (AkiraNetwork.VirtualStorageLibrary.VirtualNodeName-1.html)	A VirtualItem<T> (AkiraNetwork.VirtualStorageLibrary.VirtualItem-1.html) initialized with the specified data.

Implements  
VirtualNodeMetadatable  
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeType  
VirtualNodeNotNotFoundException  
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeType-1.html)

IVirtualDeepCloneable (AkiraNetwork.VirtualStorageLibrary.IVirtualDeepCloneable-1.html)  
IDisposable (https://learn.microsoft.com/dotnet/api/system.idisposable)

# Extension Methods

VirtualTextFormatter.GenerateSingleTableDebugText<T>(T)  
(AkiraNetwork.VirtualStorageLibrary.Utilities.VirtualTextFormatter.html#AkiraNetwork\_VirtualStorageLibrary\_Utilitie  
s\_VirtualTextFormatter\_GenerateSingleTableDebugText\_1\_\_0\_)  
VirtualNodeExtensions.ResolveNodeType(VirtualNode)  
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeExtensions.html#AkiraNetwork\_VirtualStorageLibrary\_VirtualNode  
Extensions\_ResolveNodeType\_AkiraNetwork\_VirtualStorageLibrary\_VirtualNode\_)  
(AkiraNetwork.VirtualStorageLibrary.  
NotifyNodeDelegate  
(AkiraNetwork.VirtualStorageLibrary.  
PatternMatch  
(AkiraNetwork.VirtualStorageLibrary.  
VirtualCycleDetector  
(AkiraNetwork.VirtualStorageLibrary.  
VirtualDirectory  
(AkiraNetwork.VirtualStorageLibrary.  
VirtualDirectoryAdapter<T>  
(AkiraNetwork.VirtualStorageLibrary.  
1.html)  
VirtualException  
(AkiraNetwork.VirtualStorageLibrary.  
VirtualGroupCondition<T, TKey>  
(AkiraNetwork.VirtualStorageLibrary.  
2.html)  
VirtualID  
(AkiraNetwork.VirtualStorageLibrary.  
VirtualItem  
(AkiraNetwork.VirtualStorageLibrary.  
VirtualItemAdapter<T>  
(AkiraNetwork.VirtualStorageLibrary.  
1.html)  
VirtualItem<T>  
(AkiraNetwork.VirtualStorageLibrary.  
1.html)  
VirtualNode  
(AkiraNetwork.VirtualStorageLibrary.  
VirtualNodeContext  
(AkiraNetwork.VirtualStorageLibrary.  
VirtualNodeExtensions  
(AkiraNetwork.VirtualStorageLibrary.  
VirtualNodeListConditions  
(AkiraNetwork.VirtualStorageLibrary.  
VirtualnodeName  
(AkiraNetwork.VirtualStorageLibrary.  
VirtualNodeNotFoundException  
(AkiraNetwork.VirtualStorageLibrary.  
VirtualNodeType  
(AkiraNetwork.VirtualStorageLibrary.  
VirtualNodeTypeFilter

# Class VirtualNode

Represents an abstract class for nodes.

Filter by title

## Inheritance

(AkiraNetwork.VirtualStorageLibrary)  
↳ object (<https://learn.microsoft.com/dotnet/api/system.object>)  
    ↳ NotifyNodeDelegate  
        ↳ VirtualNode  
            ↳ AkiraNetwork.VirtualStorageLibrary  
                ↳ VirtualDirectory (AkiraNetwork.VirtualStorageLibrary.VirtualDirectory.html)  
                ↳ VirtualItem (AkiraNetwork.VirtualStorageLibrary.VirtualItem.html)  
                ↳ VirtualSymbolicLink (AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink.html)  
                ↳ VirtualCycleDetector

## Implements

IVirtualDeepCloneable ([AkiraNetwork.VirtualStorageLibrary.IVirtualDeepCloneable-1.html](#)) <VirtualNode>  
(AkiraNetwork.VirtualStorageLibrary.VirtualNode.html)  
    ↳ VirtualDirectoryAdapter<T>

## Inherited Members

(AkiraNetwork.VirtualStorageLibrary)  
object.Equals(object) ([https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals\(system-object\)](https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals(system-object)))  
VirtualException  
object.Equals(object) ([https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals\(system-object\)](https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals(system-object)))  
VirtualGroupCondition<TKey>  
object.GetHashCode () (<https://learn.microsoft.com/dotnet/api/system.object.gethashcode>)  
object.GetType () (<https://learn.microsoft.com/dotnet/api/system.object.gettype>)  
object.MemberwiseClone () (<https://learn.microsoft.com/dotnet/api/system.object.memberwiseclone>)  
object.ReferenceEquals(object, object) (<https://learn.microsoft.com/dotnet/api/system.object.referenceequals>)  
object.ToString () (<https://learn.microsoft.com/dotnet/api/system.object.tostring>)

**Namespace:** AkiraNetwork ([AkiraNetwork.html](#)).VirtualStorageLibrary ([AkiraNetwork.VirtualStorageLibrary.html](#))  
VirtualItemAdapter<T>

**Assembly:** VirtualStorageLibrary.dll  
(AkiraNetwork.VirtualStorageLibrary)

## Syntax

[.html](#)  
VirtualItem<T>  
public class VirtualNode : IVirtualDeepCloneable<VirtualNode>  
1.html

VirtualNode  
(AkiraNetwork.VirtualStorageLibrary)  
VirtualNodeContext

(AkiraNetwork.VirtualStorageLibrary)  
VirtualNodeExtensions  
VirtualNodeName

Initializes a new instance of the VirtualNode ([AkiraNetwork.VirtualStorageLibrary.VirtualNode.html](#)) class.  
(AkiraNetwork.VirtualStorageLibrary)

## Declaration

VirtualNodeName  
(AkiraNetwork.VirtualStorageLibrary)  
protected NodeNotFoundException VirtualNodeName(name)

(AkiraNetwork.VirtualStorageLibrary)  
VirtualNodeType  
(AkiraNetwork.VirtualStorageLibrary)  
VirtualNodeTypeFilter

Type	Name	Description
VirtualNodeName (AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)	name	The name of node.

## VirtualNode(VirtualNodeName, DateTime)

Initializes a new instance of the VirtualNode (AkiraNetwork.VirtualStorageLibrary.VirtualNode.html) class.  
 (AkiraNetwork.VirtualStorageLibrary)

### Declaration

```
(AkiraNetwork.VirtualStorageLibrary)
protected VirtualNode(VirtualNodeName name, DateTime createdDate)
(AkiraNetwork.VirtualStorageLibrary)
  VirtualCycleDetector
```

### Parameters

(AkiraNetwork.VirtualStorageLibrary)

Type	Name	Description
VirtualDirectoryName (AkiraNetwork.VirtualStorageLibrary)	name	The name of node.
DateTime (https://learn.microsoft.com/dotnet/api/system.datetime) VirtualException (AkiraNetwork.VirtualStorageLibrary)	createdDate	The created date of node.

## VirtualNode(VirtualNodeName, DateTime, DateTime)

VirtualID  
 Initializes a new instance of the VirtualNode (AkiraNetwork.VirtualStorageLibrary.VirtualNode.html) class.  
 (AkiraNetwork.VirtualStorageLibrary)

### Declaration

```
(AkiraNetwork.VirtualStorageLibrary)
VirtualItemAdapter<T>
protected VirtualNode(VirtualNodeName name, DateTime createdDate, DateTime updatedDate)
(AkiraNetwork.VirtualStorageLibrary)
  1.html)
```

### Parameters

VirtualItem<T>  
 (AkiraNetwork.VirtualStorageLibrary)

Type	Name	Description
VirtualNodeName (AkiraNetwork.VirtualStorageLibrary)	name	The name of node.
VirtualNodeContext DateTime (https://learn.microsoft.com/dotnet/api/system.datetime) (AkiraNetwork.VirtualStorageLibrary)	createdDate	The created date of node.
VirtualNodeExtensions DateTime (https://learn.microsoft.com/dotnet/api/system.datetime) VirtualNodeListConditions (AkiraNetwork.VirtualStorageLibrary)	updatedDate	The updated date of node.

## Properties

### CreatedDate

VirtualNodeTypeFilter  
 (AkiraNetwork.VirtualStorageLibrary)

Gets the created date of the node. This date represents when the node was first created.

## Declaration

```
public DateTime CreatedDate { get; }
```



## Property Value

Type	Description
NotifyNodeDelegate<DateTime> ( <a href="https://learn.microsoft.com/dotnet/api/system.datetime">https://learn.microsoft.com/dotnet/api/system.datetime</a> )	

PatternMatch

(AkiraNetwork.VirtualStorageLibrary)

VirtualCycleDetector

(AkiraNetwork.VirtualStorageLibrary)

## IsReferencedInStorage

(AkiraNetwork.VirtualStorageLibrary)

Gets [VirtualDirectory](#) indicating whether referenced in storage. If this property is true, the node is referenced from storage.

Other (AkiraNetwork.VirtualStorageLibrary)

VirtualDirectoryAdapter<T>

(AkiraNetwork.VirtualStorageLibrary)

1.html

```
public bool IsReferencedInStorage { get; }
```

VirtualException

(AkiraNetwork.VirtualStorageLibrary)

VirtualGroupCondition<T, TKey>

(AkiraNetwork.VirtualStorageLibrary)

## Property Value

Type	Description
VirtualID <a href="https://learn.microsoft.com/dotnet/api/system.boolean">https://learn.microsoft.com/dotnet/api/system.boolean</a>	

bool

(<https://learn.microsoft.com/dotnet/api/system.boolean>)

(AkiraNetwork.VirtualStorageLibrary)

VirtualItem

(AkiraNetwork.VirtualStorageLibrary)

## Name

VirtualItemAdapter<T>

(AkiraNetwork.VirtualStorageLibrary)

Gets the name of node.  
1.html

VirtualItem<T>

(AkiraNetwork.VirtualStorageLibrary)

1.html

```
public VirtualNodeName Name { get; }
```

VirtualNode

(AkiraNetwork.VirtualStorageLibrary)

## Property Value

Type	Description
VirtualNodeContext	

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeExtensions

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeName ([VirtualStorageLibrary.VirtualNodeName.html](#))

VirtualNodeListConditions

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeName

(AkiraNetwork.VirtualStorageLibrary)

## NodeType

(AkiraNetwork.VirtualStorageLibrary)

Gets [VirtualNodeNotFoundException](#).

(AkiraNetwork.VirtualStorageLibrary)

## Declaration

VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeTypeFilter

```
public abstract VirtualNodeType NodeType { get; }
```

## Property Value

Type	Description
VirtualNodeType (AkiraNetwork.VirtualStorageLibrary.VirtualNodeType.html) (AkiraNetwork.VirtualStorageLibrary NotifyNodeDelegate (AkiraNetwork.VirtualStorageLibrary PatternMatch (AkiraNetwork.VirtualStorageLibrary Gets the updated date of the node. This date represents the last time the node was modified. It is set to the current date and time at the moment of instantiation or cloning. (AkiraNetwork.VirtualStorageLibrary Declaration	Directory

## UpdatedDate

Gets the updated date of the node. This date represents the last time the node was modified. It is set to the current date and time at the moment of instantiation or cloning.  
(AkiraNetwork.VirtualStorageLibrary  
1.html)

## Property Value

Type	Description
AkiraNetwork.VirtualStorageLibrary VirtualGroupCondition<T, TKey> DateTime ( <a href="https://learn.microsoft.com/dotnet/api/system.datetime">https://learn.microsoft.com/dotnet/api/system.datetime</a> ) (AkiraNetwork.VirtualStorageLibrary 2.html)	

## VirtualID

(AkiraNetwork.VirtualStorageLibrary  
VirtualItem  
Gets the VID of node  
(AkiraNetwork.VirtualStorageLibrary  
VirtualItemAdapter<T>

Declaration  
(AkiraNetwork.VirtualStorageLibrary  
1.html)  
public VirtualID VID { get; }  
VirtualItem<T>  
(AkiraNetwork.VirtualStorageLibrary  
1.html)

## Property Value

Type	Description
AkiraNetwork.VirtualStorageLibrary VirtualNode VirtualData (AkiraNetwork.VirtualStorageLibrary.VirtualID.html) (AkiraNetwork.VirtualStorageLibrary VirtualNodeExtensions (AkiraNetwork.VirtualStorageLibrary VirtualNodeListConditions (AkiraNetwork.VirtualStorageLibrary VirtualNodeName (AkiraNetwork.VirtualStorageLibrary DeepClone(bool)	

## Methods

VirtualNodeNotNotFoundException  
(AkiraNetwork.VirtualStorageLibrary  
Creates a deep clone of the storage library. However, the CreatedDate and UpdatedDate should not be cloned as they are set to VirtualNode date and time at the time of cloning.

(AkiraNetwork.VirtualStorageLibrary  
Declaration  
VirtualNodeTypeFilter

```
public abstract VirtualNode DeepClone(bool recursive = false)
```

## Parameters

Type	Name	Description
bool	recursive	When true, all child nodes are also cloned, creating a deep copy of the entire system. The default is false. The CreatedDate and UpdatedDate properties are not preserved. They are set to the current date and time at the moment of instantiation or cloning.

PatternMatch

Returns [AkiraNetwork.VirtualStorageLibrary](#)

Type	Description
<a href="#">VirtualCycleDetector</a> (AkiraNetwork.VirtualStorageLibrary)	
<a href="#">VirtualNode</a> (AkiraNetwork.VirtualStorageLibrary)	Cloned
<a href="#">VirtualDirectory</a> (AkiraNetwork.VirtualStorageLibrary)	<a href="#">VirtualNode</a> (AkiraNetwork.VirtualStorageLibrary.VirtualNode.html)

.htm

[VirtualDirectoryAdapter<T>](#)

(AkiraNetwork.VirtualStorageLibrary)

1.html)

[VirtualException](#)

(AkiraNetwork.VirtualStorageLibrary)

## Update(VirtualNode)

VirtualGroupCondition<T, TKey>

(AkiraNetwork.VirtualStorageLibrary)

Updates the [VirtualNode](#) ([AkiraNetwork.VirtualStorageLibrary.VirtualNode.html](#)).

2.html)

## Declaration

(AkiraNetwork.VirtualStorageLibrary)

[VirtualItem](#)

public abstract void Update(VirtualNode node)

(AkiraNetwork.VirtualStorageLibrary)

[VirtualItemAdapter<T>](#)

Parameters

(AkiraNetwork.VirtualStorageLibrary)

Type	Name	Description
<a href="#">VirtualItem</a> (AkiraNetwork.VirtualStorageLibrary)	<a href="#">node</a>	Value to update

1.html)

[VirtualNode](#)

## Implements

(AkiraNetwork.VirtualStorageLibrary)

[VirtualNodeContext](#)

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

(AkiraNetwork.VirtualStorageLibrary)

[VirtualNodeExtensions](#)

(AkiraNetwork.VirtualStorageLibrary)

[VirtualNodeListConditions](#)

[VirtualTextFormatter](#).GenerateSingleTableDebugText<T>(T)

(AkiraNetwork.VirtualStorageLibrary)

(AkiraNetwork.VirtualStorageLibrary.Utilities.VirtualTextFormatter.html#AkiraNetwork\_VirtualStorageLibrary\_Utilitie

s\_VirtualTextFormatter

.GenerateSingleTableDebugText\_1\_\_0\_)

(AkiraNetwork.VirtualStorageLibrary)

[VirtualNodeExtensions](#).ResolveNodeType(VirtualNode)

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeExtensions.html#AkiraNetwork\_VirtualStorageLibrary\_VirtualNode

Extensions.ResolveNodeType\_AkiraNetwork\_VirtualStorageLibrary\_VirtualNode\_)

[VirtualNodeType](#)

(AkiraNetwork.VirtualStorageLibrary)

[VirtualNodeTypeFilter](#)



(AkiraNetwork.VirtualStorageLibrary)  
NotifyNodeDelegate  
(AkiraNetwork.VirtualStorageLibrary)  
PatternMatch  
(AkiraNetwork.VirtualStorageLibrary)  
VirtualCycleDetector  
(AkiraNetwork.VirtualStorageLibrary)  
VirtualDirectory  
(AkiraNetwork.VirtualStorageLibrary)  
VirtualDirectoryAdapter<T>  
(AkiraNetwork.VirtualStorageLibrary)  
1.html)  
VirtualException  
(AkiraNetwork.VirtualStorageLibrary)  
VirtualGroupCondition<T, TKey>  
(AkiraNetwork.VirtualStorageLibrary)  
2.html)  
VirtualID  
(AkiraNetwork.VirtualStorageLibrary)  
VirtualItem  
(AkiraNetwork.VirtualStorageLibrary)  
VirtualItemAdapter<T>  
(AkiraNetwork.VirtualStorageLibrary)  
1.html)  
VirtualItem<T>  
(AkiraNetwork.VirtualStorageLibrary)  
1.html)  
VirtualNode  
(AkiraNetwork.VirtualStorageLibrary)  
VirtualNodeContext  
(AkiraNetwork.VirtualStorageLibrary)  
VirtualNodeExtensions  
(AkiraNetwork.VirtualStorageLibrary)  
VirtualNodeListConditions  
(AkiraNetwork.VirtualStorageLibrary)  
VirtualnodeName  
(AkiraNetwork.VirtualStorageLibrary)  
VirtualNodeNotFoundException  
(AkiraNetwork.VirtualStorageLibrary)  
VirtualNodeType  
(AkiraNetwork.VirtualStorageLibrary)  
VirtualNodeTypeFilter

# Class VirtualNodeContext

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.

## Inheritance

NotifyNodeDelegate

↳ object (<https://learn.microsoft.com/dotnet/api/system.object>)

↳ VirtualNodeContext

## Inherited Members

VirtualCycleDetector

object.Equals(object) ([https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals\(system-object\)](https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals(system-object)))

VirtualDirectory

object.Equals(object, object) ([https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals\(system-object-system-object\)](https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals(system-object-system-object)))

VirtualDirectoryAdapter<T>

object.GetHashCode() (<https://learn.microsoft.com/dotnet/api/system.object.gethashcode>)

object.GetType() (<https://learn.microsoft.com/dotnet/api/system.object.gettype>)

object.MemberwiseClone() (<https://learn.microsoft.com/dotnet/api/system.object.memberwiseclone>)

object.ReferenceEquals(object, object) (<https://learn.microsoft.com/dotnet/api/system.object.referenceequals>)

NameSpace AkiraNetwork(AkiraNetwork.html).VirtualStorageLibrary (AkiraNetwork.VirtualStorageLibrary.html)

## Assembly

(AkiraNetwork.VirtualStorageLibrary.dll)

## Syntax

VirtualID

(AkiraNetwork.VirtualStorageLibrary)  
public class VirtualNodeContext

VirtualItem

(AkiraNetwork.VirtualStorageLibrary)

VirtualItemAdapter<T>

(AkiraNetwork.VirtualStorageLibrary)

1.html)

VirtualItem<T>

## VirtualNodeContext(VirtualNode?, VirtualPath, VirtualDirectory?, int, int,

## VirtualPath?, bool, VirtualSymbolicLink?)

VirtualNode

Initializes a new instance of the VirtualNodeContext (AkiraNetwork.VirtualStorageLibrary.VirtualNodeContext.html)

(AkiraNetwork.VirtualStorageLibrary)

class VirtualNodeContext

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeExtensions

(AkiraNetwork.VirtualStorageLibrary)

public VirtualNodeContext(VirtualNode? node, VirtualPath traversalPath, VirtualDirectory? pa  
rentNode = null, int depth = 0, int index = 0, VirtualPath? resolvedPath = null, bool resolv  
ed = false, VirtualSymbolicLink? resolvedLink = null)

VirtualnodeName

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeNotFoundException

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeTypeFilter

Type	Name	Description
VirtualNode (AkiraNetwork.VirtualStorageLibrary.VirtualNo de.html)	<i>node</i>	The node being traversed.
VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPat h.html) (AkiraNetwork.VirtualStorageLibrary.VirtualDir ectory.html) NotifyNodeDelegate (AkiraNetwork.VirtualStorageLibrary.VirtualDir ectoryPatternMatch PatternMatch (AkiraNetwork.VirtualStorageLibrary.VirtualCycleDetector (https://learn.microsoft.com/dotnet/api/system.int32) VirtualDirectory VirtualDirectoryAdapter<T> (https://learn.microsoft.com/dotnet/api/system.int32) 1.html)	<i>traversalPath</i> <i>parentNode</i> <i>depth</i> <i>index</i>	The path used for node traversal. The parent directory of the node. The depth of the node. The root is considered 0. The index of the node. Corresponds to the enumeration order within the directory.
VirtualPathException (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) VirtualGroupCondition<T, TKey> VirtualLink VirtualLinkAdapter<T> VirtualLinkItem VirtualLinkItemAdapter<T>	<i>resolvedPath</i> <i>resolved</i> <i>resolvedLink</i>	The result of resolving a symbolic link. Indicates whether the symbolic link has been resolved. The resolved symbolic link.

(AkiraNetwork  
Properties

Dentist  
(Aki)

## VirtualNodeContext

## Gets the ~~the~~ Network Virtual

## VirtualNodeExtensions

## Declaration (AkiraNetwork.VirtualStorageLibrary)

## VirtualNodeListCondition

public int DepthVirtualStorageLibrary

## VirtualNodeName

- (AkiraNetwork Virt

**Property Value** VirtualNodeNotFoundException

Type: AkiraNetwork.VirtualStorageLibrary

#### **VirtualNodeType**

```
int (https://learn.microsoft.com/dotnet/  
    (AlgriaNetwork.VirtualStorage).library
```

(AkiraNetwork.VirtualStorageLibrary)

## VirtualNode typeFilter

Type	Description
AkiraNetwork.VirtualStorageLibrary.VirtualNodeType int ( <a href="https://learn.microsoft.com/dotnet/api/system.int32">https://learn.microsoft.com/dotnet/api/system.int32</a> ) (AkiraNetwork.VirtualStorageLibrary)	The depth of the node.
VirtualNodeTypeFilter	

Index

Gets the index of the node and can only be set within the assembly. This is the index corresponding to the enumeration order within the directory.

## Declaration

```
    public int Index { get; }  
    (AkiraNetwork.VirtualStorageLibrary.  
     NotifyNodeDelegate  
     property Value  
     (AkiraNetwork.VirtualStorageLibrary.  
      Type patternMatch  
      (AkiraNetwork.VirtualStorageLibrary.  
       nt (https://learn.microsoft.com/dotnet/  
        VirtualCycleDetector
```

Type	Description
(AkiraNetwork.VirtualStorageLibrary.PatternMatch) int ( <a href="https://learn.microsoft.com/dotnet/api/system.int32">https://learn.microsoft.com/dotnet/api/system.int32</a> ) VirtualCycleDetector	The index of the node.

Node

**VirtualDirectoryAdapter<T>**  
Gets the node and can only be set within the assembly.  
(AkiraNetwork.VirtualStorageLibrary)

## Declaration<sup>1.htmh</sup>

```
VirtualException  
(AkiraNetwork.VirtualStorageLibrary  
public VirtualNode? Node { get;  
VirtualGroupCondition<T, TKey>  
(AkiraNetwork.VirtualStorageLibrary
```

Property	Type	Description
Value	VirtualID (AkiraNetwork.VirtualStorageLibrary)	
VirtualNode	VirtualNode (AkiraNetwork.VirtualStorageLibrary.VirtualNode.html)	The current node.

# ParentDirectory

Gets the parent directory of the node and can only be set within the assembly.  
(AkiraNetwork.VirtualStorage library)

(AKTA) Declaration

```
VirtualNode  
public NetworkVirtualServerClientDirectory { get; }
```

Property	Type	Description
Value	AkiraNetwork.VirtualStorageLibrary.VirtualNodeContext	The instance of the parent directory.
Type	AkiraNetwork.VirtualStorageLibrary.VirtualNodeListConditions	The instance of the parent directory.

## **Perspektiv**

(AkiraNetwork.VirtualStorageLibrary)  
Gets a value indicating whether the sym-  
VirtualNode.Voe

(AkiraNetwork VirtualStorageLibrary)

## Declaration VirtualN

## **VirtualNode type filter**

```
public bool Resolved { get; }
```

#### Property Value

Type	Description
bool ( <a href="https://learn.microsoft.com/dotnet/api/system.boolean">https://learn.microsoft.com/dotnet/api/system.boolean</a> )	True if the link is resolved; otherwise, false.

### ResolvedLink

PatternMatch

Gets the resolved symbolic link and can only be set within the assembly.

VirtualCycleDetector

Declaration

(AkiraNetwork.VirtualStorageLibrary.ResolvedLink)

VirtualDirectory

```
public VirtualSymbolicLink? ResolvedLink { get; }
```

VirtualDirectoryAdapter<T>

(AkiraNetwork.VirtualStorageLibrary.VirtualDirectory)

#### Property Value

Type	Description
VirtualException	
(AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink)	
VirtualGroupCondition<T, TKey>	
(AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink.html)	The resolved symbolic link, or null.
(AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink)	
2.html)	
VirtualID	

### ResolvedPath

VirtualItem

Gets the resolved path and can only be set within the assembly. This represents the result of resolving a symbolic link. VirtualItemAdapter<T>

Declaration

(AkiraNetwork.VirtualStorageLibrary.ResolvedPath)

1.html)

VirtualItem<T>

```
public VirtualPath? ResolvedPath { get; }
```

(AkiraNetwork.VirtualStorageLibrary.ResolvedPath)

1.html)

#### Property Value

Type	Description
VirtualNodeContext	
VirtualPath ( <a href="#">AkiraNetwork.VirtualStorageLibrary.VirtualPath.html</a> )	The resolved path, or null.

### TraversalPath

VirtualNodeName

Gets the traversal path and can only be set within the assembly.

(AkiraNetwork.VirtualStorageLibrary.TraversalPath)

Declaration

(AkiraNetwork.VirtualStorageLibrary.TraversalPath)

```
public VirtualNodePath TraversalPath { get; }
```

(AkiraNetwork.VirtualStorageLibrary.TraversalPath)

VirtualNodeTypeFilter

## Property Value

Type	Description
VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	The path used for node traversal.



## Methods

(AkiraNetwork.VirtualStorageLibrary)

NotifyNodeDelegate

(AkiraNetwork.VirtualStorageLibrary)

### ToString()

PatternMatch

(AkiraNetwork.VirtualStorageLibrary)

Returns a string representation of this instance's information.

VirtualCycleDetector

(AkiraNetwork.VirtualStorageLibrary)

Declaration

AkiraNetwork.VirtualStorageLibrary

VirtualDirectory

public string ToString()

VirtualDirectoryAdapter<T>

(AkiraNetwork.VirtualStorageLibrary)

Returns

1.html

Type	Description
VirtualException	(AkiraNetwork.VirtualStorageLibrary)

string (<https://learn.microsoft.com/dotnet/api/system.string>)

VirtualGroupCondition<T, TKey>

A string representing this instance's information.

(AkiraNetwork.VirtualStorageLibrary)

Overrides

2.html

object.ToString() (<https://learn.microsoft.com/dotnet/api/system.object.tostring>)

(AkiraNetwork.VirtualStorageLibrary)

## Extension Methods

(AkiraNetwork.VirtualStorageLibrary)

VirtualTextFormatterAdapter.GenerateSingleTableDebugText<T>(T)

(AkiraNetwork.VirtualStorageLibrary)

VirtualTextFormatter\_GenerateSingleTableDebugText\_1\_\_0\_)

VirtualItem<T>

(AkiraNetwork.VirtualStorageLibrary)

1.html

VirtualNode

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeContext

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeExtensions

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeListConditions

(AkiraNetwork.VirtualStorageLibrary)

VirtualnodeName

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeNotFoundException

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeTypeFilter

# Class VirtualNodeExtensions

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

Filter by title

## Inheritance

(AkiraNetwork.VirtualStorageLibrary)  
↳ object (<https://learn.microsoft.com/dotnet/api/system.object>)  
    ↳ NotifyNodeDelegate  
        ↳ VirtualNodeExtensions  
            ↳ AkiraNetwork.VirtualStorageLibrary

## Inherited Members

(AkiraNetwork.VirtualStorageLibrary)  
object.Equals(object) ([https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals\(system-object\)](https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals(system-object)))  
VirtualCycleDetector  
(AkiraNetwork.VirtualStorageLibrary)  
object.Equals(object, object) ([https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals\(system-object-system-object\)](https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals(system-object-system-object)))  
VirtualDirectory  
(AkiraNetwork.VirtualStorageLibrary)  
object.GetHashCode() (<https://learn.microsoft.com/dotnet/api/system.object.gethashcode>)  
VirtualDirectoryAdapter<T>  
object.GetType() (<https://learn.microsoft.com/dotnet/api/system.object.gettype>)  
(AkiraNetwork.VirtualStorageLibrary)  
object.MemberwiseClone() (<https://learn.microsoft.com/dotnet/api/system.object.memberwiseclone>)  
VirtualException  
object.ReferenceEquals(object, object) (<https://learn.microsoft.com/dotnet/api/system.object.referenceequals>)  
object.ToString() (<https://learn.microsoft.com/dotnet/api/system.object.tostring>)  
(AkiraNetwork.VirtualStorageLibrary)

NameSpace [AkiraNetwork](#) ([AkiraNetwork.html](#)).VirtualStorageLibrary ([AkiraNetwork.VirtualStorageLibrary.html](#))

Assembly [AkiraNetwork.VirtualStorageLibrary.dll](#) ([AkiraNetwork.VirtualStorageLibrary.dll.html](#))

## Syntax

VirtualID  
(AkiraNetwork.VirtualStorageLibrary)  
public static class VirtualNodeExtensions

VirtualItem  
(AkiraNetwork.VirtualStorageLibrary)

VirtualItemAdapter<T>

(AkiraNetwork.VirtualStorageLibrary)

## Methods

VirtualItem<T>

(AkiraNetwork.VirtualStorageLibrary)

### ResolveNodeType(VirtualNode)

VirtualNode

Resolves the type of the specified virtual node. If the node is a symbolic link, returns the type of the target node.

(AkiraNetwork.VirtualStorageLibrary)

Otherwise, returns the type of the node itself.

(AkiraNetwork.VirtualStorageLibrary)

#### Declaration

VirtualNodeExtensions

(AkiraNetwork.VirtualStorageLibrary)

public static VirtualNodeType ResolveNodeType(this VirtualNode node)

VirtualNodeListConditions

(AkiraNetwork.VirtualStorageLibrary)

#### Parameters

VirtualNodeName  
(AkiraNetwork.VirtualStorageLibrary)

Type  
VirtualNodeNotFoundException

Name Description

Virt  
(AkiraNetwork.VirtualStorageLibrary)

node

The virtual node to resolve the type for.

(AkiraNetwork.VirtualStorageLibrary.VirtualNode.html)

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeTypeFilter

## Returns

Type	Description
VirtualNodeType (AkiraNetwork.VirtualStorageLibrary.VirtualNodeType.html) ▼	The type of the node or, if the node is a symbolic link, the type of the target node.

(AkiraNetwork.VirtualStorageLibrary.NotifyNodeDelegate  
(AkiraNetwork.VirtualStorageLibrary.PatternMatch  
(AkiraNetwork.VirtualStorageLibrary.VirtualCycleDetector  
(AkiraNetwork.VirtualStorageLibrary.VirtualDirectory  
(AkiraNetwork.VirtualStorageLibrary.VirtualDirectoryAdapter<T>  
(AkiraNetwork.VirtualStorageLibrary.VirtualException  
(AkiraNetwork.VirtualStorageLibrary.VirtualGroupCondition<T, TKey>  
(AkiraNetwork.VirtualStorageLibrary.VirtualID  
(AkiraNetwork.VirtualStorageLibrary.VirtualItem  
(AkiraNetwork.VirtualStorageLibrary.VirtualItemAdapter<T>  
(AkiraNetwork.VirtualStorageLibrary.VirtualItem<T>  
(AkiraNetwork.VirtualStorageLibrary.VirtualNode  
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeContext  
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeExtensions  
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeListConditions  
(AkiraNetwork.VirtualStorageLibrary.VirtualnodeName  
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeNotFoundException  
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeType  
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeTypeFilter

# Struct VirtualNodeListConditions

Represents the conditions for creating a list of virtual nodes.

Filter by title

## Inherited Members

(AkiraNetwork.VirtualStorageLibrary)  
Value Type Equals(object) (https://learn.microsoft.com/dotnet/api/system.valuetype.equals)  
Value Type GetHashCode() (https://learn.microsoft.com/dotnet/api/system.valuetype.gethashcode)  
Value Type ToString() (https://learn.microsoft.com/dotnet/api/system.valuetype.tostring)  
object Equals(object, object) (https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals(system-object-system-object))  
object GetType() (https://learn.microsoft.com/dotnet/api/system.object.gettype)  
object ReferenceEquals(object, object) (https://learn.microsoft.com/dotnet/api/system.object.referenceequals)  
(AkiraNetwork.VirtualStorageLibrary)  
Namespace AkiraNetwork.VirtualStorageLibrary (AkiraNetwork.VirtualStorageLibrary.html)

## Assembly! VirtualStorageLibrary.dll >

(AkiraNetwork.VirtualStorageLibrary)  
Syntax 1.html

VirtualException  
public struct VirtualNodeListConditions  
(AkiraNetwork.VirtualStorageLibrary)  
VirtualGroupCondition<T, TKey>  
(AkiraNetwork.VirtualStorageLibrary)

## Constructors

VirtualID  
VirtualID  
(AkiraNetwork.VirtualStorageLibrary)

### VirtualNodeListConditions()

(AkiraNetwork.VirtualStorageLibrary)  
Initializes a new instance of the VirtualNodeListConditions  
(AkiraNetwork.VirtualStorageLibrary.VIRTUALNODELISTCONDITIONS)

1.html  
Declaration  
VirtualItem<T>

(AkiraNetwork.VirtualStorageLibrary)  
public VirtualNodeListConditions()  
1.html

VirtualNode  
(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeContext  
(AkiraNetwork.VirtualStorageLibrary)  
VirtualGroupCondition<VirtualNode, object>?,  
List<VirtualSortCondition<VirtualNode>>?)  
(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeListConditions  
Initializes a New instance of the VirtualNodeListConditions  
(AkiraNetwork.VirtualStorageLibrary.VIRTUALNODELISTCONDITIONS.html) class with the specified conditions.  
VirtualNodeName

Declaration  
(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeNotFoundException  
public VirtualNodeListConditions(VirtualNodeTypeFilter filter, VirtualGroupCondition<VirtualNode, object>? groupCondition, List<VirtualSortCondition<VirtualNode>>? sortConditions)  
(AkiraNetwork.VirtualStorageLibrary)  
VirtualNodeTypeFilter



## Property Value

Type	Description
VirtualGroupCondition (AkiraNetwork.VirtualStorageLibrary.VirtualGroupCondition-2.html) < VirtualNode (AkiraNetwork.VirtualStorageLibrary.VirtualNode.html) , object ( <a href="https://learn.microsoft.com/dotnet/api/system.object">https://learn.microsoft.com/dotnet/api/system.object</a> )> (AkiraNetwork.VirtualStorageLibrary. PatternMatch (AkiraNetwork.VirtualStorageLibrary. FileDetector	The condition for grouping virtual nodes. The default value is <code>null</code> . You can specify <code>null</code> if no grouping condition is needed.

## SortConditions

(AkiraNetwork.VirtualStorageLibrary.VirtualDirectory)  
Specifies the sorting conditions, determining the order in which the virtual nodes are arranged.

## Declaration

```
VirtualDirectoryAdapter<T>
(AkiraNetwork.VirtualStorageLibrary.VirtualSortCondition<VirtualNode>>? SortConditions { readonly get; set; }  

1.html)
```

VirtualException

## Property Value

Type	Description
VirtualGroupCondition<T, TKey> (AkiraNetwork.VirtualStorageLibrary.List 2.html) ( <a href="https://learn.microsoft.com/dotnet/api/system.collections.generic.list-1">https://learn.microsoft.com/dotnet/api/system.collections.generic.list-1</a> ) list-1(AkiraNetwork.VirtualStorageLibrary.< VirtualItem VirtualSortCondition (AkiraNetwork.VirtualStorageLibrary.VirtualSortCondition-1.html) < (AkiraNetwork.VirtualStorageLibrary.VirtualNode VirtualNode (AkiraNetwork.VirtualStorageLibrary.VirtualNode.html) >> (AkiraNetwork.VirtualStorageLibrary. 1.html)	A list of conditions for sorting the virtual nodes. The default value is <code>null</code> . You can specify <code>null</code> if no sorting condition is needed.

## Extension Methods

VirtualNodeContext  
VirtualTextFormatter.GenerateSingleTableDebugText<T>(T)  
(AkiraNetwork.VirtualStorageLibrary.Utilities.VirtualTextFormatter.html#AkiraNetwork\_VirtualStorageLibrary\_Utilities\_VirtualTextFormatter\_GenerateSingleTableDebugText\_1\_\_0\_)

VirtualNodeListConditions

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeNotFoundException

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeTypeFilter

# Class VirtualNodeName

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

Inheritance  
(AkiraNetwork.VirtualStorageLibrary)

NotifyNodeDelegate

↳ object (<https://learn.microsoft.com/dotnet/api/system.object>)

↳ VirtualNodeName

PatternMatch

Implementation  
(AkiraNetwork.VirtualStorageLibrary)

VirtualCycleDetector

IEquatable (<https://learn.microsoft.com/dotnet/api/system.iequatable-1>)<VirtualNodeName

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html) >

VirtualDirectory

IComparable (<https://learn.microsoft.com/dotnet/api/system.icomparable-1>)<VirtualNodeName

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html) >

VirtualDirectoryAdapter<T>

IComparable (<https://learn.microsoft.com/dotnet/api/system.icomparable>)

(AkiraNetwork.VirtualStorageLibrary)

Inherited Members  
[1.html](#)

object.Equals(object, object) (<https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals>)

(AkiraNetwork.VirtualStorageLibrary)

object.GetHashCode() (<https://learn.microsoft.com/dotnet/api/system.object.gethashcode>)

(AkiraNetwork.VirtualStorageLibrary)

object.MemberwiseClone() (<https://learn.microsoft.com/dotnet/api/system.object.memberwiseclone>)

object.ReferenceEquals(object, object) (<https://learn.microsoft.com/dotnet/api/system.object.referenceequals>)

VirtualID

Namespace: AkiraNetwork ([AkiraNetwork.html](#)).VirtualStorageLibrary ([AkiraNetwork.VirtualStorageLibrary.html](#))

(AkiraNetwork.VirtualStorageLibrary)

Assembly: VirtualStorageLibrary.dll

VirtualItem

Syntax  
(AkiraNetwork.VirtualStorageLibrary)

VirtualItemAdapter<T>

public class VirtualNodeName : IComparable<VirtualNodeName>, IComparable<VirtualNodeName>, ICmpable

VirtualItem<T>

(AkiraNetwork.VirtualStorageLibrary)

1.html)

VirtualNode

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeContext

(AkiraNetwork.VirtualStorageLibrary)

Initializes a new instance of the VirtualNodeName ([AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html](#))

class  
(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeListConditions

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeName

public class VirtualNodeName (string name)

VirtualNodeNotFoundException

(AkiraNetwork.VirtualStorageLibrary)

Parameters

VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeTypeFilter



```
public int CompareTo(VirtualNodeName? other)
```

## Parameters

Type	Name	Description
VirtualNodeName (AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html) Delegate (AkiraNetwork.VirtualStorageLibrary.PatternMatch PatternMatch (AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)	other	The VirtualNodeName (AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html) to compare with.

## Returns

Type	Description
VirtualDirectory (AkiraNetwork.VirtualStorageLibrary.VirtualDirectory.html)	A value indicating the relative order of the instances being compared.

## CompareTo(object?)

Compares this instance with a specified object.  
(AkiraNetwork.VirtualStorageLibrary.VirtualGroupCondition<T, TKey>  
2.html)

## Declaration

```
VirtualItem CompareTo(object? obj)
(AkiraNetwork.VirtualStorageLibrary.VirtualItemAdapter<T>
VirtualItemAdapter<T>
```

## Parameters

Type	Name	Description
VirtualItem<T> object (https://learn.microsoft.com/dotnet/api/system.object) (AkiraNetwork.VirtualStorageLibrary.VirtualItem.html)	obj	The object to compare with.

## Returns

Type	Description
VirtualNodeContext int (AkiraNetwork.VirtualStorageLibrary.VirtualNodeContext.html) VirtualNodeListConditions (https://learn.microsoft.com/dotnet/api/system.int32) (AkiraNetwork.VirtualStorageLibrary.VirtualNodeListConditions.html)	A value indicating the relative order of the instances being compared.

## Exceptions

Type	Condition
VirtualNodeName (AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html) VirtualNodeNotFoundException (http://AkiraNetwork.VirtualStorageLibrary/api/system.argumentOutOfRangeException VirtualNodeTypeFilter (AkiraNetwork.VirtualStorageLibrary.VirtualNodeTypeFilter.html) VirtualNodeTypeFilter	Thrown when the object is not a VirtualNodeName (AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html).

## Equals(VirtualNodeName?)

Determines whether the specified VirtualNodeName (AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html) is equal to this instance.

### Declaration

```
public bool Equals(VirtualNodeName? other)  
(AkiraNetwork.VirtualStorageLibrary.  
NotifyNodeDelegate
```

### Parameters

Type	Name	Description
atternMatch (AkiraNetwork.VirtualStorageLibrary. VirtualNodeName (AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html) VirtualCycleDetector (AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html) VirtualDirectoryName (AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html) VirtualDirectory (AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html) VirtualDirectoryAdapter<T> (AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)	other	The VirtualNodeName (AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html) to compare with this instance.

### Returns

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)

Type	Description
VirtualException bool (AkiraNetwork.VirtualStorageLibrary. (https://learn.microsoft.com/dotnet/api/system.boolean) VirtualGroupCondition<T, TKey> net/api/system.boolean) (AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html) 2.html)	true if the specified VirtualNodeName (AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html) is equal to this instance; otherwise, false .

VirtualID

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)

## Equals(object?)

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)

Determines whether the specified object is equal to this instance.

### Declaration

```
1.html)  
VirtualItem<T>  
public override bool Equals(object? obj)  
(AkiraNetwork.VirtualStorageLibrary.  
1.html)
```

### Parameters

Type	Name	Description
VirtualNodeContext (AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)	obj (AkiraNetwork.VirtualStorageLibrary. (https://learn.microsoft.com/dotnet/api/system.object) VirtualNodeExtensions (AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)	obj The object to compare with this instance.

### Returns

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)

Type	Description
VirtualNodeName (AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html) VirtualNodeNotFoundException (AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)	true if the specified object is equal to this instance; otherwise, false .

### Overrides

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)  
VirtualNodeType  
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)  
VirtualNodeTypeFilter

object.Equals(object) ([https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals\(system-object\)](https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals(system-object)))

## GenerateNodeName(string)

Generates a new node name using the specified prefix.

### Declaration

```
(AkiraNetwork.VirtualStorageLibrary)
    NotifyNodeDelegate
public static VirtualNodeName GenerateNodeName(string prefix)
(AkiraNetwork.VirtualStorageLibrary)
    PatternMatch
```

### Parameters

Type	Name	Description
VirtualCycleDetector (AkiraNetwork.VirtualStorageLibrary)		

string ( <a href="https://learn.microsoft.com/dotnet/api/system.string">https://learn.microsoft.com/dotnet/api/system.string</a> )	prefix	The prefix for the node name.
--	--------	-------------------------------

### Returns

VirtualDirectoryName  
(AkiraNetwork.VirtualStorageLibrary)

Type	Description
VirtualException VirtualNodeName (AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)	The generated node name.

### Remarks

(AkiraNetwork.VirtualStorageLibrary)  
The current implementation may be improved for better usability, including a possible re-implementation of the node name generation. Users are advised to check for updates in the documentation regularly.

### Exceptions

Type	Condition
ArgumentException (AkiraNetwork.VirtualStorageLibrary.Arguments.ArgumentException)	Thrown when the prefix is an empty string.

### 1.html

VirtualItem<T>

(AkiraNetwork.VirtualStorageLibrary)
1.html

## GetHashCode()

VirtualNode  
Returns a hash code for this instance.  
(AkiraNetwork.VirtualStorageLibrary)

### Declaration

```
(AkiraNetwork.VirtualStorageLibrary)
    VirtualNodeExtension
public int GetHashCode()
(AkiraNetwork.VirtualStorageLibrary)
```

### Returns

VirtualNodeName  
(AkiraNetwork.VirtualStorageLibrary)

Type	Description
VirtualNodeName (AkiraNetwork.VirtualStorageLibrary)	A hash code for this instance.

### Overrides

VirtualNodeType  
object.GetHashCode() (<https://learn.microsoft.com/dotnet/api/system.object.gethashcode>)

VirtualNodeTypeFilter

## IsValidNodeName(VirtualNodeName)

Checks if a node name is valid.

### Declaration

```
public static bool IsValidNodeName(VirtualNodeName nodeName)
```

#### Parameters

NotifyNodeDelegate

Type AkiraNetwork.VirtualStorageLibrary

VirtualNodeName

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)

VirtualCycleDetector

#### Returns

VirtualDirectory

Type AkiraNetwork.VirtualStorageLibrary

VirtualDirectoryAdapter<T>

bool (AkiraNetwork.VirtualStorageLibrary)

(https://learn.microsoft.com/dotnet/api/system.boolean.1.html)

VirtualException

(AkiraNetwork.VirtualStorageLibrary)

## ResetCounter()

(AkiraNetwork.VirtualStorageLibrary)

Resets the counter used for node name generation.

### Declaration

(AkiraNetwork.VirtualStorageLibrary)

VirtualItem

public static void ResetCounter()

(AkiraNetwork.VirtualStorageLibrary)

VirtualItemAdapter<T>

(AkiraNetwork.VirtualStorageLibrary)

1.html)

## ToString()

VirtualItem<T>

Returns the name of the node as a string.

1.html)

### Declaration

VirtualNode

(AkiraNetwork.VirtualStorageLibrary)

virtualNode.ToString() string ToString()

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeExtensions

Returns

(AkiraNetwork.VirtualStorageLibrary)

Type VirtualNodeListConditions

(AkiraNetwork.VirtualStorageLibrary)

string (https://learn.microsoft.com/dotnet/api/system.string)

VirtualnodeName

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeNotFoundException

Overrides object.ToString() (https://learn.microsoft.com/dotnet/api/system.object.tostring)

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeTypeFilter

## Operators

### Type

### Name

### Description

nodeName

The node name to check.

### Type

### Description

true if the node name is valid; otherwise, false .

**operator ==(VirtualNodeName?, VirtualNodeName?)**

Determines whether two VirtualNodeName (AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html) instances are equal.

## Declaration

1

```
public static bool operator ==(VirtualNodeName? left, VirtualNodeName? right)  
    (AkiraNetwork.VirtualStorageLibrary
```

## Notify Parameters

Type	Name	Description
(AkiraNetwork.VirtualStorageLibrary.VirtualDirectoryName.html)	left	The left VirtualDirectoryName (AkiraNetwork.VirtualStorageLibrary.VirtualDirectoryName.html) VirtualDirectory (AkiraNetwork.VirtualStorageLibrary.VirtualDirectoryName.html)
VirtualDirectoryNameAdapter<T>		to compare.
(AkiraNetwork.VirtualStorageLibrary.VirtualDirectoryName.html)	right	The right VirtualDirectoryName (AkiraNetwork.VirtualStorageLibrary.VirtualDirectoryName.html) VirtualException (AkiraNetwork.VirtualStorageLibrary.VirtualDirectoryName.html)
VirtualGroupCondition<T, TKey>		to compare.

(AKTU)

Type	Description
VirtualID (AkiraNetwork.VirtualStorageLibrary)	
bool VirtualItem (AkiraNetwork.VirtualStorageLibrary) n) VirtualItemAdapter<T>	true if the specified instances are equal; otherwise, false .

## **implicit operator string(VirtualNodeName)**

(AkiraNetwork.virtualStorage.getLibrary)

Implicitly converts a VirtualNodeName (AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html) to a string.

## **Declaration**

	<p>VirtualNodeContext</p> <p>(AkiraNetwork.VirtualStorageLibrary)</p> <pre>public static VirtualNodeContext string(VirtualNodeName nodeName)</pre> <p>VirtualNodeExtensions</p> <p>(AkiraNetwork.VirtualStorageLibrary)</p>						
<b>Parameters</b>	<p>VirtualNodeListConditions</p> <table border="1"> <thead> <tr> <th>Type</th><th>Name</th><th>Description</th></tr> </thead> <tbody> <tr> <td> <p>VirtualNodeName</p> <p>(AkiraNetwork.VirtualStorageLibrary)</p> <p>VirtualNodeName</p> <p>(AkiraNetwork.VirtualStorageLibrary)</p> <p>(AkiraNetwork.VirtualStorageLibrary.VirtualNodeNameNotFoundException)</p> <p>(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)</p> <p>VirtualNodeType</p> <p>(AkiraNetwork.VirtualStorageLibrary)</p> </td><td> <p>nodeName</p> </td><td> <p>The</p> <p>VirtualNodeName</p> <p>(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)</p> <p>to convert.</p> </td></tr> </tbody> </table>	Type	Name	Description	<p>VirtualNodeName</p> <p>(AkiraNetwork.VirtualStorageLibrary)</p> <p>VirtualNodeName</p> <p>(AkiraNetwork.VirtualStorageLibrary)</p> <p>(AkiraNetwork.VirtualStorageLibrary.VirtualNodeNameNotFoundException)</p> <p>(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)</p> <p>VirtualNodeType</p> <p>(AkiraNetwork.VirtualStorageLibrary)</p>	<p>nodeName</p>	<p>The</p> <p>VirtualNodeName</p> <p>(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)</p> <p>to convert.</p>
Type	Name	Description					
<p>VirtualNodeName</p> <p>(AkiraNetwork.VirtualStorageLibrary)</p> <p>VirtualNodeName</p> <p>(AkiraNetwork.VirtualStorageLibrary)</p> <p>(AkiraNetwork.VirtualStorageLibrary.VirtualNodeNameNotFoundException)</p> <p>(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)</p> <p>VirtualNodeType</p> <p>(AkiraNetwork.VirtualStorageLibrary)</p>	<p>nodeName</p>	<p>The</p> <p>VirtualNodeName</p> <p>(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)</p> <p>to convert.</p>					

## Returns

Type	Description
string ( <a href="https://learn.microsoft.com/dotnet/api/system.string">https://learn.microsoft.com/dotnet/api/system.string</a> )	



## implicit operator VirtualNodeName(string)

(AkiraNetwork.VirtualStorageLibrary)  
Implicitly converts a string to a VirtualNodeName ([AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html](#)).  
NotifyNodeDelegate

### Declaration

```
PatternMatch  
(AkiraNetwork.VirtualStorageLibrary) public static implicit operator VirtualNodeName(string name)  
VirtualCycleDetector  
(AkiraNetwork.VirtualStorageLibrary)
```

### Parameters

Type	Name	Description
VirtualDirectoryName ( <a href="https://learn.microsoft.com/dotnet/api/system.string">https://learn.microsoft.com/dotnet/api/system.string</a> )	name	The string to convert.

### Returns

Type	Description
VirtualNodeName ( <a href="#">AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html</a> )	

(AkiraNetwork.VirtualStorageLibrary)  
VirtualGroupCondition<T, TKey>

(AkiraNetwork.VirtualStorageLibrary)  
VirtualItemAdapter<T>

### Parameters

Type	Description
VirtualItemAdapter<T>	

Determines whether two VirtualNodeName ([AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html](#)) instances are not equal.

(AkiraNetwork.VirtualStorageLibrary)

### Declaration

```
VirtualItem<T>  
(AkiraNetwork.VirtualStorageLibrary) public bool operator !=(VirtualNodeName? left, VirtualNodeName? right)  
1.html)
```

### Parameters

(AkiraNetwork.VirtualStorageLibrary)

Type	Name	Description
VirtualNodeContext (AkiraNetwork.VirtualStorageLibrary) VirtualNodeName VirtualNodeExtensions (AkiraNetwork.VirtualStorageLibrary) Virtual (AkiraNetwork.VirtualStorageLibrary) Virtual NodeName.html) VirtualNodeListConditions (AkiraNetwork.VirtualStorageLibrary) VirtualNodeName	left	The left VirtualNodeName (AkiraNetwork.VirtualStorageLibrary) VirtualNodeName.html) to compare.
VirtualNodeContext (AkiraNetwork.VirtualStorageLibrary) VirtualNodeName VirtualNodeExtensions (AkiraNetwork.VirtualStorageLibrary) Virtual (AkiraNetwork.VirtualStorageLibrary) Virtual NodeType (AkiraNetwork.VirtualStorageLibrary) Virtual NodeTypeFilter	right	The right VirtualNodeName (AkiraNetwork.VirtualStorageLibrary) VirtualNodeName.html) to compare.

## Returns

Type	Description
bool ( <a href="https://learn.microsoft.com/dotnet/api/system.boolean">https://learn.microsoft.com/dotnet/api/system.boolean</a> )	true if the specified instances are not equal; otherwise, false.

## Implements

NotifyNodeDelegate  
(AkiraNetwork.VirtualStorageLibrary  
IEquatable<<https://learn.microsoft.com/dotnet/api/system.iequatable-1>>)  
IComparable<<https://learn.microsoft.com/dotnet/api/system.icomparable-1>>  
IComparable(<https://learn.microsoft.com/dotnet/api/system.icomparable>)

VirtualCycleDetector

(AkiraNetwork.VirtualStorageLibrary  
VirtualDirectory)

VirtualTextFormatter.GenerateSingleTableDebugText<T>(T)

(AkiraNetwork.VirtualStorageLibrary.Utilities.VirtualTextFormatter.html#AkiraNetwork\_VirtualStorageLibrary\_Utilsili

s\_VirtualTextFormatter\_GenerateSingleTableDebugText\_1\_\_0\_)

1.html)

VirtualException

(AkiraNetwork.VirtualStorageLibrary  
VirtualGroupCondition<T, TKey>)

(AkiraNetwork.VirtualStorageLibrary  
2.html)

VirtualID

(AkiraNetwork.VirtualStorageLibrary  
VirtualItem)

(AkiraNetwork.VirtualStorageLibrary  
VirtualItemAdapter<T>)

(AkiraNetwork.VirtualStorageLibrary  
1.html)

VirtualItem<T>

(AkiraNetwork.VirtualStorageLibrary  
1.html)

VirtualNode

(AkiraNetwork.VirtualStorageLibrary  
VirtualNodeContext)

(AkiraNetwork.VirtualStorageLibrary  
VirtualNodeExtensions)

(AkiraNetwork.VirtualStorageLibrary  
VirtualNodeListConditions)

(AkiraNetwork.VirtualStorageLibrary  
VirtualnodeName)

(AkiraNetwork.VirtualStorageLibrary  
VirtualNodeNotFoundException)

(AkiraNetwork.VirtualStorageLibrary  
VirtualNodeType)

(AkiraNetwork.VirtualStorageLibrary  
VirtualNodeTypeFilter)

## Extension Methods



Initializes a new instance of the VirtualNodeNotFoundException  
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeNotFoundException.html) class.

## Declaration

```
public VirtualNodeNotFoundException()  
    ▾
```

(AkiraNetwork.VirtualStorageLibrary)

## VirtualNodeNotFoundException(string)

(AkiraNetwork.VirtualStorageLibrary)

Initializes a new instance of the VirtualNodeNotFoundException

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeNotFoundException.html) class with a specified error message.

### Declaration

(AkiraNetwork.VirtualStorageLibrary)

VirtualDirectory

```
public VirtualNodeNotFoundException(string message)  
(AkiraNetwork.VirtualStorageLibrary)
```

VirtualDirectoryAdapter<T>

### Parameters

(AkiraNetwork.VirtualStorageLibrary)

1.html)

Type VirtualException

Name

Description

string (AkiraNetwork.VirtualStorageLibrary) (https://learn.microsoft.com/dotnet/api/system.string)

message

The error message.

VirtualGroupCondition<T, TKey>

(AkiraNetwork.VirtualStorageLibrary)

2.html)

## VirtualNodeNotFoundException(string, Exception)

(AkiraNetwork.VirtualStorageLibrary)

Initializes a new instance of the VirtualNodeNotFoundException

VirtualItem

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeNotFoundException.html) class with a specified error message

and a reference to the inner exception that is the cause of this exception.

VirtualItemAdapter<T>

### Declaration

(AkiraNetwork.VirtualStorageLibrary)

1.html)

```
public VirtualNodeNotFoundException(string message, Exception innerException)  
(AkiraNetwork.VirtualStorageLibrary)
```

1.html)

### Parameters

VirtualNode

Type (AkiraNetwork.VirtualStorageLibrary)

Name

Description

VirtualNodeContext

message

The error message.

string (AkiraNetwork.VirtualStorageLibrary)

(https://learn.microsoft.com/dotnet/api/system.string)

VirtualNodeExtensions

(AkiraNetwork.VirtualStorageLibrary)

Exception (https://learn.microsoft.com/dotnet/api/system.exception)

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeCondition

innerException

The exception that is the cause of this exception.

VirtualNodeName

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeNotFoundException

(AkiraNetwork.VirtualStorageLibrary)

ISerializable (VirtualNodeType) (https://learn.microsoft.com/dotnet/api/system.runtime.serialization.iserializable)

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeTypeFilter

## Implements

# Extension Methods

VirtualTextFormatter.GenerateSingleTableDebugText<T>(T)  
(AkiraNetwork.VirtualStorageLibrary.Utilities.VirtualTextFormatter.html#AkiraNetwork\_VirtualStorageLibrary\_Utils  
s\_VirtualTextFormatter\_GenerateSingleTableDebugText\_1\_\_0\_)

▼

(AkiraNetwork.VirtualStorageLibrary.  
NotifyNodeDelegate  
(AkiraNetwork.VirtualStorageLibrary.  
PatternMatch  
(AkiraNetwork.VirtualStorageLibrary.  
VirtualCycleDetector  
(AkiraNetwork.VirtualStorageLibrary.  
VirtualDirectory  
(AkiraNetwork.VirtualStorageLibrary.  
VirtualDirectoryAdapter<T>  
(AkiraNetwork.VirtualStorageLibrary.  
1.html)  
VirtualException  
(AkiraNetwork.VirtualStorageLibrary.  
VirtualGroupCondition<T, TKey>  
(AkiraNetwork.VirtualStorageLibrary.  
2.html)  
VirtualID  
(AkiraNetwork.VirtualStorageLibrary.  
VirtualItem  
(AkiraNetwork.VirtualStorageLibrary.  
VirtualItemAdapter<T>  
(AkiraNetwork.VirtualStorageLibrary.  
1.html)  
VirtualItem<T>  
(AkiraNetwork.VirtualStorageLibrary.  
1.html)  
VirtualNode  
(AkiraNetwork.VirtualStorageLibrary.  
VirtualNodeContext  
(AkiraNetwork.VirtualStorageLibrary.  
VirtualNodeExtensions  
(AkiraNetwork.VirtualStorageLibrary.  
VirtualNodeListConditions  
(AkiraNetwork.VirtualStorageLibrary.  
VirtualnodeName  
(AkiraNetwork.VirtualStorageLibrary.  
VirtualNodeNotFoundException  
(AkiraNetwork.VirtualStorageLibrary.  
VirtualNodeType  
(AkiraNetwork.VirtualStorageLibrary.  
VirtualNodeTypeFilter

# Enum VirtualNodeType

Specifies the types of virtual nodes.

Filter by title

**Namespace:** AkiraNetwork (AkiraNetwork.html).VirtualStorageLibrary (AkiraNetwork.VirtualStorageLibrary.html)  
(AkiraNetwork.VirtualStorageLibrary)

**Assembly:** VirtualStorageLibrary.dll  
NotifyNodeDelegate

**Syntax**  
(AkiraNetwork.VirtualStorageLibrary)

PatternMatch

public enum VirtualNodeType

VirtualCycleDetector

(AkiraNetwork.VirtualStorageLibrary)

VirtualDirectory

(AkiraNetwork.VirtualStorageLibrary)

VirtualDirectoryAdapter<T>

## Fields

Name	Description
1.html	
Directory	Represents a directory node.
VirtualException	Represents an item node with a user-defined type T.
Item	Represents an item node with a user-defined type T.
VirtualGroupCondition<T, TKey>	
None	Represents an unspecified or unknown node type.
2.html	
SymbolicLink	Represents a symbolic link node.
VirtualID	

(AkiraNetwork.VirtualStorageLibrary)

VirtualItem

(AkiraNetwork.VirtualStorageLibrary)

VirtualItemAdapter<T>

VirtualTextFormatter.GenerateSingleTableDebugText<VirtualNodeType>(VirtualNodeType)

(AkiraNetwork.VirtualStorageLibrary)  
(AkiraNetwork.VirtualStorageLibrary.Utilities.VirtualTextFormatter.html#AkiraNetwork\_VirtualStorageLibrary\_Utilities\_VirtualTextFormatter\_GenerateSingleTableDebugText\_1\_\_0\_)

VirtualItem<T>

(AkiraNetwork.VirtualStorageLibrary)

1.html

VirtualNode

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeContext

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeExtensions

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeListConditions

(AkiraNetwork.VirtualStorageLibrary)

VirtualnodeName

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeNotFoundException

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeTypeFilter

## Extension Methods

(AkiraNetwork.VirtualStorageLibrary)

VirtualTextFormatter

GenerateSingleTableDebugText<VirtualNodeType>(VirtualNodeType)

(AkiraNetwork.VirtualStorageLibrary)  
(AkiraNetwork.VirtualStorageLibrary.Utilities.VirtualTextFormatter.html#AkiraNetwork\_VirtualStorageLibrary\_Utilities\_VirtualTextFormatter\_GenerateSingleTableDebugText\_1\_\_0\_)

VirtualItem<T>

(AkiraNetwork.VirtualStorageLibrary)

1.html

VirtualNode

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeContext

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeExtensions

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeListConditions

(AkiraNetwork.VirtualStorageLibrary)

VirtualnodeName

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeNotFoundException

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeTypeFilter

# Enum VirtualNodeTypeFilter

Specifies the filters for virtual node types.

Filter by title

**Namespace:** AkiraNetwork (AkiraNetwork.html).VirtualStorageLibrary (AkiraNetwork.VirtualStorageLibrary.html)

(AkiraNetwork.VirtualStorageLibrary)

**Assembly:** VirtualStorageLibrary.dll

NotifyNodeDelegate

**Syntax**

```
(AkiraNetwork.VirtualStorageLibrary)
  PatternMatch
public enum VirtualNodeTypeFilter
  VirtualCycleDetector
(AkiraNetwork.VirtualStorageLibrary)
  VirtualDirectory
(AkiraNetwork.VirtualStorageLibrary)
  VirtualDirectoryAdapter<T>
```

## Fields

Name	Description
1.html	
All	VirtualException Filter for all node types.
Directory	(AkiraNetwork.VirtualStorageLibrary) Filter for directory nodes.
VirtualGroupCondition<T, TKey>	
Item	(AkiraNetwork.VirtualStorageLibrary) Item nodes with a user-defined type T.
2.html	
None	VirtualID No specific node type is specified.
SymbolicLink	(AkiraNetwork.VirtualStorageLibrary) Filter for symbolic link nodes.
VirtualItem	

## Extension Methods

1.html

VirtualTextFormatter.GenerateSingleTableDebugText<VirtualNodeTypeFilter>(VirtualNodeTypeFilter)

(AkiraNetwork.VirtualStorageLibrary.Utilities.VirtualTextFormatter.html#AkiraNetwork\_VirtualStorageLibrary\_Utils

s\_VirtualTextFormatter\_GenerateSingleTableDebugText\_1\_\_0\_)

1.html

VirtualNode

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeContext

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeExtensions

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeListConditions

(AkiraNetwork.VirtualStorageLibrary)

VirtualnodeName

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeNotFoundException

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeTypeFilter



## Declaration

```
public VirtualPath(string path)
```

### Parameters

Type	Name	Description
string ( <a href="https://learn.microsoft.com/dotnet/api/system.string">https://learn.microsoft.com/dotnet/api/system.string</a> )	path	

## Properties

### BaseDepth

Declaration: `int BaseDepth { get; }`

(AkiraNetwork.VirtualStorageLibrary)

Property Value: `VirtualGroupCondition<T, TKey>`

Type	Description
AkiraNetwork.VirtualStorageLibrary	

int (<https://learn.microsoft.com/dotnet/api/system.int32>)

(AkiraNetwork.VirtualStorageLibrary)

VIRTUALITEM

(AkiraNetwork.VirtualStorageLibrary)

VIRTUALITEMADAPTER<T>

Declaration: `VirtualItemAdapter<T>`

(AkiraNetwork.VirtualStorageLibrary)

1.html

PROPERTYDEPTH

{ get; }

(AkiraNetwork.VirtualStorageLibrary)

1.html

PROPERTYVALUE

VirtualNode

Type	Description
AkiraNetwork.VirtualStorageLibrary	

int (<https://learn.microsoft.com/dotnet/api/system.int32>)

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeExtensions

(AkiraNetwork.VirtualStorageLibrary)

### DirectoryPath

Declaration: `VirtualPath DirectoryPath { get; }`

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeNotFoundException

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeTypeFilter

Type	Description
VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	

## Dot

### Declaration

```
(AkiraNetwork.VirtualStorageLibrary)
    NotifyNodeDelegate
public static string Dot { get; }
```

PatternMatch

### Property Value

Type	Description
VirtualCycleDetector	

```
(AkiraNetwork.VirtualStorageLibrary)
string VirtualDirectory (https://www.microsoft.com/dotnet/api/system.string)
```

(AkiraNetwork.VirtualStorageLibrary)

VirtualDirectoryAdapter<T>

(AkiraNetwork.VirtualStorageLibrary)

## DotDot

1.html

### Declaration

```
(AkiraNetwork.VirtualStorageLibrary)
    VirtualGroupCondition<T, TKey>
public static string DotDot { get; }
```

(AkiraNetwork.VirtualStorageLibrary)

2.html

### Property Value

Type	Description
VirtualItem	

```
string (https://www.microsoft.com/dotnet/api/system.string)
```

VirtualItemAdapter<T>

(AkiraNetwork.VirtualStorageLibrary)

1.html

## FixedPath

VirtualItem<T>

### Declaration

```
1.html
VirtualNode
public VirtualPath FixedPath { get; }
```

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeContext

### Property Value

Type	Description
VirtualNodeExtensions	

(AkiraNetwork.VirtualStorageLibrary)

```
VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)
```

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeName

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeNotFoundException

### IsAbsolute

VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeTypeFilter

```
public bool IsAbsolute { get; }
```

## Property Value

Type	Description
bool ( <a href="https://learn.microsoft.com/dotnet/api/system.boolean">https://learn.microsoft.com/dotnet/api/system.boolean</a> )	

(AkiraNetwork.VirtualStorageLibrary)  
NotifyNodeDelegate  
(AkiraNetwork.VirtualStorageLibrary)  
**IsDot**  
PatternMatch  
(AkiraNetwork.VirtualStorageLibrary)

Declaration  
VirtualCycleDetector  
(AkiraNetwork.VirtualStorageLibrary)  
public bool IsDot { get; }  
VirtualDirectory  
(AkiraNetwork.VirtualStorageLibrary)

Property Value  
VirtualDirectoryAdapter<T>  
(AkiraNetwork.VirtualStorageLibrary)

Type	Description
1.html)	

bool (<https://learn.microsoft.com/dotnet/api/system.boolean>)

(AkiraNetwork.VirtualStorageLibrary)  
VirtualException  
VirtualGroupCondition<T, TKey>  
(AkiraNetwork.VirtualStorageLibrary)

**IsDotDot**  
2.html)

Declaration  
VirtualHD  
(AkiraNetwork.VirtualStorageLibrary)  
VirtualItem  
public bool IsDotDot { get; }  
(AkiraNetwork.VirtualStorageLibrary)  
VirtualItemAdapter<T>

Property Value  
(AkiraNetwork.VirtualStorageLibrary)

Type	Description
1.html)	

VirtualItem<T>  
bool (<https://learn.microsoft.com/dotnet/api/system.boolean>)  
(AkiraNetwork.VirtualStorageLibrary)

1.html)  
VirtualNode  
(AkiraNetwork.VirtualStorageLibrary)  
VirtualNodeContext

Declaration  
(AkiraNetwork.VirtualStorageLibrary)  
VirtualNodeExtensions  
(AkiraNetwork.VirtualStorageLibrary)  
VirtualNodeListConditions  
(AkiraNetwork.VirtualStorageLibrary)

Property Value  
VirtualNodeName  
(AkiraNetwork.VirtualStorageLibrary)

Type	Description
VirtualNodeNotFoundException bool ( <a href="https://learn.microsoft.com/dotnet/api/system.boolean">https://learn.microsoft.com/dotnet/api/system.boolean</a> ) (AkiraNetwork.VirtualStorageLibrary)	

VirtualNodeType  
(AkiraNetwork.VirtualStorageLibrary)  
VirtualNodeTypeFilter

## IsEndsWithSlash

### Declaration

```
public bool IsEndsWithSlash { get; }
```



### Property Value

Type	Description
NotifyNodeDelegate<VirtualStorageLibrary> bool ( <a href="https://learn.microsoft.com/dotnet/api/system.boolean">https://learn.microsoft.com/dotnet/api/system.boolean</a> ) (AkiraNetwork.VirtualStorageLibrary)	

PatternMatch

(AkiraNetwork.VirtualStorageLibrary)

### IsRoot

(AkiraNetwork.VirtualStorageLibrary)

### Declaration

Directory  
(AkiraNetwork.VirtualStorageLibrary)

~~VirtualDirectoryNameAdapter~~ { get; }

(AkiraNetwork.VirtualStorageLibrary)

1.html)

Type	Description
VirtualException	

Type (AkiraNetwork.VirtualStorageLibrary)

VirtualGroupCondition<T, TKey>	
bool ( <a href="https://learn.microsoft.com/dotnet/api/system.boolean">https://learn.microsoft.com/dotnet/api/system.boolean</a> ) (AkiraNetwork.VirtualStorageLibrary)	

2.html)

VirtualID

### NodeName

VirtualItem

### Declaration

(AkiraNetwork.VirtualStorageLibrary)

VirtualItemAdapter<T>

~~VirtualDirectoryNameAdapter~~ { get; }

1.html)

Type	Description
VirtualItem<T>	

Type (AkiraNetwork.VirtualStorageLibrary)

VirtualNode	
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html) (AkiraNetwork.VirtualStorageLibrary)	

VirtualNodeContext

(AkiraNetwork.VirtualStorageLibrary)

### PartsList

NodeExtensions  
(AkiraNetwork.VirtualStorageLibrary)

### Declaration

VirtualNodeListConditions

(AkiraNetwork.VirtualStorageLibrary)

public List<VirtualNodeName> PartsList { get; }

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeNotFoundException

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeTypeFilter

Type	Description
List ( <a href="https://learn.microsoft.com/dotnet/api/system.collections.generic.list-1">https://learn.microsoft.com/dotnet/api/system.collections.generic.list-1</a> )< VirtualNodeName (AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)>	

## Path

(AkiraNetwork.VirtualStorageLibrary)  
**Declaration**  
 NotifyNodeDelegate  
 (AkiraNetwork.VirtualStorageLibrary)  
 public string Path { get; }  
 PatternMatch  
 (AkiraNetwork.VirtualStorageLibrary)

**Property Value**

Type	Description
VirtualDirectory	

## Root

VirtualException  
 (AkiraNetwork.VirtualStorageLibrary)  
**Declaration**  
 VirtualGroupCondition<T, TKey>  
 (AkiraNetwork.VirtualStorageLibrary)  
 public static string Root { get; }  
 2.html)

VirtualID

**Property Value**

Type	Description
VirtualItem	

## Separator

(AkiraNetwork.VirtualStorageLibrary)  
**Declaration**  
 VirtualNode  
 (AkiraNetwork.VirtualStorageLibrary)  
 public static char Separator { get; }  
 VirtualNodeContext  
 (AkiraNetwork.VirtualStorageLibrary)

**Property Value**

Type	Description
VirtualNodeListConditions	

## Methods

VirtualNodeType  
 (AkiraNetwork.VirtualStorageLibrary)  
 VirtualNodeTypeFilter

## AddEndSlash()

### Declaration

```
public VirtualPath AddEndSlash()
```



### Returns

Type	Description
NotifyNodeDelegate<VirtualPath> (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	
(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	

PatternMatch

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

VirtualCodeDetector

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

## AddStartSlash()

### Declaration

Directory (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

```
public VirtualDirectoryAdapter<VirtualPath> AddStartSlash()
```

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

1.html)

### Returns

VirtualException

Type	Description
VirtualGroupCondition<T, TKey>	
VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

2.html)

VirtualID

## ArePathsSubdirectories(VirtualPath, VirtualPath)

### Declaration

VirtualItem (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

VirtualItemAdapter<T>

```
public static bool ArePathsSubdirectories(VirtualPath path1, VirtualPath path2)
```

1.html)

VirtualItem<T>

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

### Type

.html)

### Parameters

Type	Name	Description
VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	path1	
(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	path2	

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

### Returns

VirtualNodeExtensions

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

### Type

VirtualNodeListConditions

Type	Description
bool (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	

bool (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

VirtualnodeName

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

VirtualNodeNotFoundException

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

## Combine(params VirtualPath[])

### Declaration

VirtualNodeType (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

VirtualNodeTypeFilter

```
public VirtualPath Combine(params VirtualPath[] paths)
```

## Parameters

Type		Name	Description
VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)[]		paths	

#### **Put-NotifyNodeDelegate**

Returns	<code>VirtualPath</code>	Description
<b>Type</b>	(AkiraNetwork.VirtualStorageLibrary.PatternMatch)	

## VirtualCycleDetector

**Combine(params string[])**  
(AkiraNetwork.VirtualStorageLibrary)

**Combine(params string[])**  
(AktronNetwork.VirtualStorage)

#### -VirtualDirectoryAdapter<T>

## Declaration (Alien)

```
(AkiraNetwork.VirtualStorageLibrary  
 1.html)  
public static string Combine(params string[] paths)  
    VirtualException
```

Parametric

Type	(AkiraNetwork.VirtualStorageLibrary 2.html)	Name	Description
string	[get] <a href="https://learn.microsoft.com/dotnet/api/system.string">https://learn.microsoft.com/dotnet/api/system.string</a> []	paths	

(AkiraNetw  
Visited on

Returns	VirtualItem
Type	(AkiraNetwork.VirtualStorageLibrary.VirtualItemAdapter<T>)
string	(AkiraNetwork.VirtualStorageLibrary.VirtualItemAdapter<T>.GetNetPath(string))

string.(https://learn.freecodecamp.org/html-and-css/1.html)

**CombineFromIndex(VirtualPath, int)**

## VirtualNode Deployment

## Declaration (AkiraNetwork)

```
VirtualNodeContext  
public VirtualPath CombineFromIndex(VirtualPath path, int index)  
(AkiraNetworkVirtualStorage library)
```

VIAVANTWORK.VIRTUALIZATION

## VirtualNodeExtensions

Parameters		Name	Description
Type	Value		
VirtualNodeListConditions (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)		<i>path</i>	
int (https://learn.microsoft.com/dotnet/api/system.int32) VirtualNodeNotFoundException		<i>index</i>	

`VirtuanodeNetBindException`  
(AkiraNetwork VirtualStorage library)

**Returns** VirtualNodeType  
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeTypeFilter)

Type	Description
VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	

## CompareTo(VirtualPath?)

Compares the current instance with another object of the same type and returns an integer that indicates whether the current instance precedes, follows, or occurs in the same position in the sort order as the other object.

(AkiraNetwork.VirtualStorageLibrary.NotifyNodeDelegate)

Declaration:

```
PatternMatch
public int CompareTo(VirtualPath? other)
VirtualCycleDetector
(AkiraNetwork.VirtualStorageLibrary)
```

Parameters:

Type	Name	Description
VirtualDirectory (AkiraNetwork.VirtualStorageLibrary.VirtualDirectoryAdapter<T>)	other	An object to compare with this instance.

VirtualException

Return Value:

Type	Description								
VirtualGroupCondition<T, TKey> (AkiraNetwork.VirtualStorageLibrary)	A value that indicates the relative order of the objects being compared.								
int (2.html) (http://srftable.microsoft.com/dotnet/)	The return value has these meanings:								
api/AkiraNetwork/VirtualStorageLibrary/VirtualItem (AkiraNetwork.VirtualStorageLibrary.VirtualItemAdapter<T>)	<table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>Less than zero</td> <td>This instance precedes <code>other</code> in the sort order.</td> </tr> <tr> <td>Zero</td> <td>This instance occurs in the same position in the sort order as <code>other</code>.</td> </tr> <tr> <td>Greater than zero</td> <td>This instance follows <code>other</code> in the sort order.</td> </tr> </tbody> </table>	Value	Meaning	Less than zero	This instance precedes <code>other</code> in the sort order.	Zero	This instance occurs in the same position in the sort order as <code>other</code> .	Greater than zero	This instance follows <code>other</code> in the sort order.
Value	Meaning								
Less than zero	This instance precedes <code>other</code> in the sort order.								
Zero	This instance occurs in the same position in the sort order as <code>other</code> .								
Greater than zero	This instance follows <code>other</code> in the sort order.								

VirtualNode

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeContext

(AkiraNetwork.VirtualStorageLibrary)

Compares the current instance with another object of the same type and returns an integer that indicates whether the current instance precedes, follows, or occurs in the same position in the sort order as the other object.

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeListConditions)

Declaration:

```
VirtualNodeName
public int CompareTo(object? obj)
VirtualNodeNotFoundException
(AkiraNetwork.VirtualStorageLibrary)
```

Parameters:

```
VirtualNodeType
(AkiraNetwork.VirtualStorageLibrary)
VirtualNodeTypeFilter
```

Type	Name	Description
object ( <a href="https://learn.microsoft.com/dotnet/api/system.object">https://learn.microsoft.com/dotnet/api/system.object</a> )	<i>obj</i>	An object to compare with this instance.

## Returns

Type	Description
int (AkiraNetwork.VirtualStorageLibrary (https://learn.microsoft.com/dotnet/ NotifyNodeDelegate api/system.int32) (AkiraNetwork.VirtualStorageLibrary PatternMatch (AkiraNetwork.VirtualStorageLibrary VirtualCycleDetector (AkiraNetwork.VirtualStorageLibrary VirtualDirectory (AkiraNetwork.VirtualStorageLibrary VirtualDirectoryAdapter<T>	A value that indicates the relative order of the objects being compared. The return value has these meanings:
Value	Meaning
Less than zero	This instance precedes <code>obj</code> in the sort order.
Zero	This instance occurs in the same position in the sort order as <code>obj</code> .
Greater than zero	This instance follows <code>obj</code> in the sort order.

**Exceptions**<sup>1.html</sup>)

**VirtualException**

Type	Condition
<code>VirtualGroupCondition&lt;T, TKey&gt;</code> <code>ArgumentException</code> ( <a href="https://learn.microsoft.com/dotnet/api/system.argumentexception">https://learn.microsoft.com/dotnet/api/system.argumentexception</a> )	<code>obj</code> is not the same type as this instance.
<code>VirtualException</code> <code>AkiraNetwork.VirtualStorageLibrary</code> ( <a href="https://learn.microsoft.com/dotnet/api/system.virtualexception">https://learn.microsoft.com/dotnet/api/system.virtualexception</a> )	<code>obj</code> is not the same type as this instance.

**Equals(VirtualPath?)**

(AkiraNetwork VirtualStorage) library

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

VirtualItemAdapter<T>

## Declaration (Akira N)

[imulation  
1.html](#))

```
public bool Equals(VirtualPath? other)
```

1.html

## Parameter Nodes

Type	(AkiraNetwork.VirtualStorageLibrary.VirtualNodeContext)	Name	Description
VirtualPath	(AkiraNetwork.VirtualStorageLibrary.VIRTUALPATH)	other	An object to compare with this object.

## **VirtualNetwork.virtualStora**

Returns	
Type	(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName)
Description	VirtualNodeName
bool	(AkiraNetwork.VirtualStorageLibrary.VirtualStorageLibrary) <a href="http://microsoft.com/dotnet/csharp/language-reference/builtin-types/bool">microsoft.com/dotnet/csharp/language-reference/builtin-types/bool</a> ) if the <a href="http://microsoft.com/dotnet/api/AkiraNetwork.VirtualStorageLibrary.VirtualStorageLibrary.VirtualNodeName">VirtualNodeName</a> is equal to the other parameter; otherwise, <a href="http://microsoft.com/dotnet/api/AkiraNetwork.VirtualStorageLibrary.VirtualStorageLibrary.VirtualNodeName">VirtualNodeName</a> is not found.
crosses	(AkiraNetwork.VirtualStorageLibrary.VirtualStorageLibrary) <a href="http://microsoft.com/dotnet/csharp/language-reference/builtin-types/bool">microsoft.com/dotnet/csharp/language-reference/builtin-types/bool</a> ).
net	AkiraNetwork.VirtualStorageLibrary.VirtualNodeType
bool	(AkiraNetwork.VirtualStorageLibrary.VirtualStorageLibrary) <a href="http://microsoft.com/dotnet/csharp/language-reference/builtin-types/bool">microsoft.com/dotnet/csharp/language-reference/builtin-types/bool</a> )
	VirtualNodeTypeFilter

## Equals(object?)

Determines whether the specified object is equal to the current object.

### Declaration

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

#### Parameters

NotifyNodeDelegate

Type	Name	Description
AkiraNetwork.VirtualStorageLibrary.NotifyNodeDelegate	object	The object to compare with the current object.

#### Returns

VirtualDirectory

Type	Description
AkiraNetwork.VirtualStorageLibrary.VirtualGroupCondition<T, TKey>	VirtualDirectoryAdapter<T> true (https://learn.microsoft.com/dotnet/csharp/language-reference/builtin-types/bool) if the specified object is equal to the current object; otherwise, false (https://learn.microsoft.com/dotnet/csharp/language-reference/builtin-types/bool).

#### Overrides

2.html

object.Equals(object) (https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals(system-object))	AkiraNetwork.VirtualStorageLibrary.VirtualGroupCondition<T, TKey>
---	---

## GetHashCode()

VirtualItemAdapter<T>

(AkiraNetwork.VirtualStorageLibrary.VirtualItemAdapter<T>)	Serves as the default hash function. 1.html
--	---

### Declaration

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

VirtualNode

(AkiraNetwork.VirtualStorageLibrary.VirtualNode)	VirtualNodeContext
--	--------------------

Type	Description
AkiraNetwork.VirtualStorageLibrary.VirtualNodeExtensions	A hash code for the current object.

VirtualNodeListConditions

Overrides	(AkiraNetwork.VirtualStorageLibrary.VirtualNodeListConditions)
object.GetHashCode() (https://learn.microsoft.com/dotnet/api/system.object.gethashcode)	(AkiraNetwork.VirtualStorageLibrary.VirtualNodeListConditions)

VirtualNodeNotFoundException

GetnodeName()	(AkiraNetwork.VirtualStorageLibrary.VirtualNodeNotFoundException)
---------------	---

### Declaration

```
public override string GetnodeName()
```

VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeType)	VirtualNodeTypeFilter
--	-----------------------

```
public VirtualNodeName GetNodeName()
```

## Returns

Type	Description
VirtualNodeName (AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)	(AkiraNetwork.VirtualStorageLibrary.NotifyNodeDelegate (AkiraNetwork.VirtualStorageLibrary.PatternMatch))

## GetParentPath()

### Declaration

```
public VirtualPath GetParentPath()
{
    VirtualDirectory
}
```

### Returns

```
VirtualDirectoryAdapter<T>
{
    VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)
    VirtualException (AkiraNetwork.VirtualStorageLibrary.VirtualException.html)
    VirtualGroupCondition<T, TKey>
}
```

## GetPartsLinkedList()

### Declaration

```
public VirtualItem<T> GetPartsLinkedList<VirtualNodeName>()
{
    VirtualItemAdapter<T>
}
```

### Returns

```
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)
{
    VirtualItem<T>
    LinkedList<(https://learn.microsoft.com/dotnet/api/system.collections.generic.linkedlist-1)<VirtualNodeName> (AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)
}
```

## GetPartsList()

### Declaration

```
public VirtualNodeList<VirtualNodeName> GetPartsList()
{
    VirtualNodeList<VirtualNodeName>
}
```

### Returns

```
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)
{
    VirtualNodeNotFoundException (AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)
    VirtualNodeTypeFilter (AkiraNetwork.VirtualStorageLibrary.VirtualNodeType.html)
}
```

## GetRelativePath(VirtualPath)

### Declaration

```
public VirtualPath GetRelativePath(VirtualPath basePath)
```



### Parameters

Type	Name	Description
(AkiraNetwork.VirtualStorageLibrary.VirtualPath) NotifyNodeDelegate (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) (AkiraNetwork.VirtualStorageLibrary.VirtualPath)	basePath	

PatternMatch

### Returns

Type	Description
(VirtualCycleDetector (AkiraNetwork.VirtualStorageLibrary.VirtualPath) VirtualPath(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) (AkiraNetwork.VirtualStorageLibrary.VirtualPath) VirtualDirectoryAdapter<T> (AkiraNetwork.VirtualStorageLibrary.VirtualPath) 1.html)	

## IsSubdirectory(VirtualPath)

### Declaration

```
virtual void IsSubdirectory(TKey virtualGroupCondition, TKey virtualPath parentPath)
    (AkiraNetwork.VirtualStorageLibrary.VirtualItemAdapter<T>
        (AkiraNetwork.VirtualStorageLibrary.VirtualPath parentPath)
        2.html)
```

### Parameters

Type	Name	Description
(AkiraNetwork.VirtualStorageLibrary.VirtualItem VirtualPath(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) (AkiraNetwork.VirtualStorageLibrary.VirtualPath)	parentPath	

VirtualItemAdapter<T>

### Returns

Type	Description
VirtualItem<T> bool(AkiraNetwork.VirtualStorageLibrary.VirtualPath) 1.html)	

VirtualNode

## NormalizePath()

### Parameters

```
VirtualNodeExtensions
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeListConditions()
    VirtualNodeContext)
```

### Returns

Type	Description
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName VirtualNodeNotFoundException VirtualPath(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) (AkiraNetwork.VirtualStorageLibrary.VirtualPath) VirtualNodeType (AkiraNetwork.VirtualStorageLibrary.VirtualNodeTypeFilter)	

## NormalizePath(string)

### Declaration

```
public static string NormalizePath(string path)
```



### Parameters

Type	Name	Description
(AkiraNetwork.VirtualStorageLibrary.NotifyNodeDelegate<T> (https://learn.microsoft.com/dotnet/api/system.string))	<i>path</i>	

PatternMatch

Return (AkiraNetwork.VirtualStorageLibrary.

Type	Description
(VirtualCycleDetector (AkiraNetwork.VirtualStorageLibrary. string (https://learn.microsoft.com/dotnet/api/system.string))	

(AkiraNetwork.VirtualStorageLibrary.

VirtualDirectoryAdapter<T>

(AkiraNetwork.VirtualStorageLibrary.  
1.html)

### StartsWith(VirtualPath)

#### Declaration

(AkiraNetwork.VirtualStorageLibrary.

VirtualGroupCondition<T> VirtualPath path)

(AkiraNetwork.VirtualStorageLibrary.

2.html)

#### Parameters

Type	Name	Description
(AkiraNetwork.VirtualStorageLibrary.VirtualItem VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) (AkiraNetwork.VirtualStorageLibrary.	<i>path</i>	

VirtualItemAdapter<T>

Return (AkiraNetwork.VirtualStorageLibrary.

1.html)

Type	Description
(VirtualItem<T> bool (https://learn.microsoft.com/dotnet/api/system.boolean))	

1.html)

VirtualNode

(AkiraNetwork.VirtualStorageLibrary.

VirtualNodeContext

Returns a string that represents the current object.

VirtualNodeExtensions

Declaration (AkiraNetwork.VirtualStorageLibrary.

VirtualNodeListConditions

public NetworkVirtualStorageFiltering()

VirtualNodeName

(AkiraNetwork.VirtualStorageLibrary.

Returns VirtualNodeNotFoundException

Type	Description
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeType string (https://learn.microsoft.com/dotnet/api/system.string))	A string that represents the current object.

(AkiraNetwork.VirtualStorageLibrary.

VirtualNodeTypeFilter

## Overrides

object.ToString() (<https://learn.microsoft.com/dotnet/api/system.object.tostring>)

### **TrimEndSlash()**

## **Declaration**

Type	Description
AkiraNetwork.VirtualStorageLibrary.VirtualCycleDetector AkiraNetwork.VirtualStorageLibrary.VirtualPath AkiraNetwork.VirtualStorageLibrary.VirtualPath.html	

# (AkiraNetwo Operators

**operator + (VirtualPath, VirtualNodeName)**

## Declaration (Akira N)

```
VirtualGroupCondition<T, TKey>
(AkiraNetwork.VirtualStorageLibrary)
public static VirtualPath operator +(VirtualPath path, VirtualNodeName nodeName)
? html)
```

## VirtualID

Parameters			
Type	Description	Name	Description
VirtualItem (AkiraNetwork.VirtualStorageLibrary.VirtualItem.html)			
VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)		<i>path</i>	
VirtualNodeName (AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html) 1.html		<i>nodeName</i>	

**VirtualItem<T>**

Type.html)	Description
VirtualNode VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) AkiraNetwork.VirtualStorageLibrary	

## **operator + (VirtualPath, VirtualPath)**

(AkiraNetwork.VirtualStorageLibrary)

**Declaration** `Virtual NodeList Conditions`

(AkiraNetwork.VirtualStorageLibrary)

~~VirtualNodeName virtualPath~~ opera

(AkiraNetwork.VirtualStorageLibrary

## VirtualNodeNotFoundException

Parameters	(AkiraNetwork.VirtualStorageLibrary.VirtualPath.AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	Name	Description
Type	virtualNodeType (AkiraNetwork.VirtualStorageLibrary.VirtualPath.AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) VirtualPath.AkiraNetwork.VirtualStorageLibrary.VirtualPath.html VirtualNodeTypeFilter	path1	





Type	Name	Description
VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	left	
VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	right	

## Returns

Type	Description
(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	

## implicit operator string(VirtualPath?)

Declaration

```
public static implicit operator string(VirtualPath? virtualPath)
    VirtualDirectoryAdapter<T>
(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)
```

Parameters

Type	Name	Description
(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	virtualPath	

Returns

Type	Description
(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	

## implicit operator VirtualPath(string)

Declaration

```
public static implicit operator VirtualPath(string path)
    VirtualNode
(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)
```

Parameters

Type	Name	Description
(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	path	

Returns

Type	Description
(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	

## operator !=(VirtualPath?, VirtualPath?)



# Class VirtualSortCondition<T>

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

## Inheritance

NotifyNodeDelegate

↳ object (<https://learn.microsoft.com/dotnet/api/system.object>)  
↳ [\(AkiraNetwork.VirtualStorageLibrary\)](#)  
↳ VirtualSortCondition<T>  
↳ PatternMatch

## Inherited Members

VirtualCycleDetector  
object.Equals(object) ([https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals\(system-object\)](https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals(system-object)))  
VirtualDirectory  
object.Equals(object, object) ([https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals\(system-object-system-object\)](https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals(system-object-system-object)))  
VirtualDirectoryAdapter<T>  
object.GetHashCode() (<https://learn.microsoft.com/dotnet/api/system.object.gethashcode>)  
object.GetType() (<https://learn.microsoft.com/dotnet/api/system.object.gettype>)  
1.html  
object.MemberwiseClone() (<https://learn.microsoft.com/dotnet/api/system.object.memberwiseclone>)  
VirtualException  
object.ReferenceEquals(object, object) (<https://learn.microsoft.com/dotnet/api/system.object.referenceequals>)  
object.ToString() (<https://learn.microsoft.com/dotnet/api/system.object.tostring>)  
VirtualGroupCondition<T, TKey>

**Name** [\(AkiraNetwork.VirtualStorageLibrary\)](#) [\(AkiraNetwork.VirtualStorageLibrary.html\)](#) [\(AkiraNetwork.VirtualStorageLibrary\)](#)

## Assembly

VirtualStorageLibrary.dll

## Syntax

(AkiraNetwork.VirtualStorageLibrary)

VirtualItem

public class VirtualSortCondition<T>  
(AkiraNetwork.VirtualStorageLibrary)

VirtualItemAdapter<T>

**Type** [\(AkiraNetwork.VirtualStorageLibrary\)](#)

Name	Description
T	(AkiraNetwork.VirtualStorageLibrary) The type of the entity to be sorted.

1.html

VirtualNode

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeContext

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeFilter

## VirtualSortCondition(Expression<Func<T, object>>, bool)

(AkiraNetwork.VirtualStorageLibrary)

Initializes a new instance of the VirtualSortCondition<T>

(AkiraNetwork.VirtualStorageLibrary) class with the specified sorting property and order.

VirtualNodeName

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeNotFoundException

(AkiraNetwork.VirtualStorageLibrary)

public VirtualSortCondition(Expression<Func<T, object>> sortBy, bool ascending = true)

VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeTypeFilter

Type	Name	Description
Expression ( <a href="https://learn.microsoft.com/dotnet/api/system.linq.expressions.expression-1">https://learn.microsoft.com/dotnet/api/system.linq.expressions.expression-1</a> ) <Func ( <a href="https://learn.microsoft.com/dotnet/api/system.func-2">https://learn.microsoft.com/dotnet/api/system.func-2</a> )<T, object ( <a href="https://learn.microsoft.com/dotnet/api/system.object">https://learn.microsoft.com/dotnet/api/system.object</a> )>>	sortBy	The property to sort by.
bool ( <a href="https://learn.microsoft.com/dotnet/api/system.boolean">https://learn.microsoft.com/dotnet/api/system.boolean</a> ) (AkiraNetwork.VirtualStorageLibrary) NotifyNodeDelegate (AkiraNetwork.VirtualStorageLibrary) PatternMatch (AkiraNetwork.VirtualStorageLibrary) VirtualCycleDetector (AkiraNetwork.VirtualStorageLibrary) VirtualDirectory (AkiraNetwork.VirtualStorageLibrary) VirtualDirectoryAdapter<T> (AkiraNetwork.VirtualStorageLibrary)	ascending	A value indicating whether the sorting order is ascending.

## Properties

### Ascending

Gets or sets a value indicating whether the sorting order is ascending. True if the order is ascending; otherwise, false.

[VirtualException](#)

### Declaration

```
VirtualGroupCondition<T, TKey>
(AkiraNetwork.VirtualStorageLibrary)
public bool Ascending { get; set; }
```

[VirtualID](#)

### PropertyValue

Type	Description
VirtualItem (AkiraNetwork.VirtualStorageLibrary)	A boolean value indicating whether the sorting order is ascending.

### SortBy

Gets or sets the property used for sorting.

[VirtualNodeContext](#)

### Declaration

```
VirtualNode
(AkiraNetwork.VirtualStorageLibrary)
VirtualNodeExtensions
(AkiraNetwork.VirtualStorageLibrary)
public VirtualNode<T>> SortBy { get; set; }
```

[VirtualNodeListConditions](#)

### PropertyValue

[VirtualNodeName](#)

Type	Description
VirtualNodeNotFoundException (AkiraNetwork.VirtualStorageLibrary) <Func ( <a href="https://learn.microsoft.com/dotnet/api/system.func-2">https://learn.microsoft.com/dotnet/api/system.func-2</a> )<T, object ( <a href="https://learn.microsoft.com/dotnet/api/system.object">https://learn.microsoft.com/dotnet/api/system.object</a> )>>	An expression that specifies the property to use for sorting.

# Extension Methods

VirtualTextFormatter.GenerateSingleTableDebugText<T>(T)  
(AkiraNetwork.VirtualStorageLibrary.Utilities.VirtualTextFormatter.html#AkiraNetwork\_VirtualStorageLibrary\_Utilitie  
s\_VirtualTextFormatter\_GenerateSingleTableDebugText\_1\_\_0\_)

▼

(AkiraNetwork.VirtualStorageLibrary.  
NotifyNodeDelegate  
(AkiraNetwork.VirtualStorageLibrary.  
PatternMatch  
(AkiraNetwork.VirtualStorageLibrary.  
VirtualCycleDetector  
(AkiraNetwork.VirtualStorageLibrary.  
VirtualDirectory  
(AkiraNetwork.VirtualStorageLibrary.  
VirtualDirectoryAdapter<T>  
(AkiraNetwork.VirtualStorageLibrary.  
1.html)  
VirtualException  
(AkiraNetwork.VirtualStorageLibrary.  
VirtualGroupCondition<T, TKey>  
(AkiraNetwork.VirtualStorageLibrary.  
2.html)  
VirtualID  
(AkiraNetwork.VirtualStorageLibrary.  
VirtualItem  
(AkiraNetwork.VirtualStorageLibrary.  
VirtualItemAdapter<T>  
(AkiraNetwork.VirtualStorageLibrary.  
1.html)  
VirtualItem<T>  
(AkiraNetwork.VirtualStorageLibrary.  
1.html)  
VirtualNode  
(AkiraNetwork.VirtualStorageLibrary.  
VirtualNodeContext  
(AkiraNetwork.VirtualStorageLibrary.  
VirtualNodeExtensions  
(AkiraNetwork.VirtualStorageLibrary.  
VirtualNodeListConditions  
(AkiraNetwork.VirtualStorageLibrary.  
VirtualnodeName  
(AkiraNetwork.VirtualStorageLibrary.  
VirtualNodeNotFoundException  
(AkiraNetwork.VirtualStorageLibrary.  
VirtualNodeType  
(AkiraNetwork.VirtualStorageLibrary.  
VirtualNodeTypeFilter

# Class VirtualStorageExtensions

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

Filter by title

## Inheritance

(AkiraNetwork.VirtualStorageLibrary)  
↳ object (https://learn.microsoft.com/dotnet/api/system.object)  
    ↳ NotifyNodeDelegate  
        ↳ VirtualStorageExtensions  
            ↳ AkiraNetwork.VirtualStorageLibrary

## Inherited Members

(AkiraNetwork.VirtualStorageLibrary)  
object.Equals(object) (https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals(system-object))  
VirtualCycleDetector  
(AkiraNetwork.VirtualStorageLibrary)  
object.Equals(object, object) (https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals(system-object-system-object))  
VirtualDirectory  
(AkiraNetwork.VirtualStorageLibrary)  
object.GetHashCode() (https://learn.microsoft.com/dotnet/api/system.object.gethashcode)  
VirtualDirectoryAdapter<T>  
object.GetType() (https://learn.microsoft.com/dotnet/api/system.object.gettype)  
(AkiraNetwork.VirtualStorageLibrary)  
object.MemberwiseClone() (https://learn.microsoft.com/dotnet/api/system.object.memberwiseclone)  
VirtualException  
object.ReferenceEquals(object, object) (https://learn.microsoft.com/dotnet/api/system.object.referenceequals)  
object.ToString() (https://learn.microsoft.com/dotnet/api/system.object.tostring)  
(AkiraNetwork.VirtualStorageLibrary)

Name space AkiraNetwork (AkiraNetwork.html).VirtualStorageLibrary (AkiraNetwork.VirtualStorageLibrary.html)

Assembly AkiraNetwork.VirtualStorageLibrary (AkiraNetwork.VirtualStorageLibrary.dll)

## Syntax

VirtualID  
(AkiraNetwork.VirtualStorageLibrary)  
public static class VirtualStorageExtensions

VirtualItem  
(AkiraNetwork.VirtualStorageLibrary)

VirtualItemAdapter<T>  
(AkiraNetwork.VirtualStorageLibrary)

## Methods

VirtualItem<T>  
(AkiraNetwork.VirtualStorageLibrary)

**ApplySortConditions<T>(IEnumerable<T>, List<VirtualSortCondition<T>>?)**  
VirtualNode  
Applies the specified sort conditions to the elements of the source sequence.  
(AkiraNetwork.VirtualStorageLibrary)

## Declaration

(AkiraNetwork.VirtualStorageLibrary)  
VirtualNodeExtensions  
public static IEnumerable<T> ApplySortConditions<T>(this IEnumerable<T> source, List<VirtualSortCondition<T>>? sortConditions = null)  
VirtualNodeListConditions  
(AkiraNetwork.VirtualStorageLibrary)

## Parameters

VirtualNodeName  
(AkiraNetwork.VirtualStorageLibrary)  
VirtualNodeNotFoundException  
(AkiraNetwork.VirtualStorageLibrary)  
VirtualNodeType  
(AkiraNetwork.VirtualStorageLibrary)  
VirtualNodeTypeFilter

Type	Name	Description
IEnumerable ( <a href="https://learn.microsoft.com/dotnet/api/system.collections.generic.ienumerable-1">https://learn.microsoft.com/dotnet/api/system.collections.generic.ienumerable-1</a> )  <T> List ( <a href="https://learn.microsoft.com/dotnet/api/system.collections.generic.list-1">https://learn.microsoft.com/dotnet/api/system.collections.generic.list-1</a> ) NotifyNodeDelegate < (AkiraNetwork.VirtualStorageLibrary.VirtualSortCondition VirtualSortCondition PatternMatch (AkiraNetwork.VirtualStorageLibrary.VirtualSortCondition- 1.html) VirtualCycleDetector <T>(AkiraNetwork.VirtualStorageLibrary.VirtualDirectory VirtualDirectory Return(AkiraNetwork.VirtualStorageLibrary.VirtualDirectoryAdapter<T> (AkiraNetwork.VirtualStorageLibrary.VirtualDirectoryAdapter<T> IEnumerable ( <a href="https://learn.microsoft.com/dotnet/api/system.collections.generic.ienumerable-1">https://learn.microsoft.com/dotnet/api/system.collections.generic.ienumerable-1</a> ) <T>VirtualGroupCondition<T, TKey> (AkiraNetwork.VirtualStorageLibrary.VirtualGroupCondition<T, TKey> 2.html)	source	The sequence of elements to sort.

Type	Description
(AkiraNetwork.VirtualStorageLibrary.VirtualDirectoryAdapter<T> IEnumerable ( <a href="https://learn.microsoft.com/dotnet/api/system.collections.generic.ienumerable-1">https://learn.microsoft.com/dotnet/api/system.collections.generic.ienumerable-1</a> ) <T>VirtualGroupCondition<T, TKey> (AkiraNetwork.VirtualStorageLibrary.VirtualGroupCondition<T, TKey> 2.html)	An IEnumerable<T> ( <a href="https://learn.microsoft.com/dotnet/api/system.collections.generic.ienumerable-1">https://learn.microsoft.com/dotnet/api/system.collections.generic.ienumerable-1</a> ) containing the sorted elements.

Type Parameters	
Name	Description
T (AkiraNetwork.VirtualStorageLibrary.VirtualItem)	The type of elements in the source sequence.

VirtualItem
(AkiraNetwork.VirtualStorageLibrary.VirtualItem)

## GroupAndSort<T>(IEnumerable<T>, VirtualGroupCondition<T, object>?, List<VirtualSortCondition<T>>?)

Groups and sorts the elements of the source sequence based on the specified conditions.  
VirtualNode

Declaration  
AkiraNetwork.VirtualStorageLibrary.VirtualNodeContext  
public static VirtualNodeGroupAndSort<T>(this IEnumerable<T> source, VirtualGroupCondition<T, object>? groupCondition = null, List<VirtualSortCondition<T>>? sortConditions = null)  
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeContext)

VirtualNodeListConditions  
Parameters  
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeListConditions)

Type	Name	Description
VirtualNodeName (AkiraNetwork.VirtualStorageLibrary.VirtualNodeName) IEnumerable VirtualNodeNotFoundException ( <a href="https://learn.microsoft.com/dotnet/api/system.collections.generic.ienumerable-1">https://learn.microsoft.com/dotnet/api/system.collections.generic.ienumerable-1</a> ) <T> (AkiraNetwork.VirtualStorageLibrary.VirtualNodeType) VirtualNodeTypeFilter	source	The sequence of elements to group and sort.

Type	Name	Description
VirtualGroupCondition (AkiraNetwork.VirtualStorageLibrary.VirtualGroupCondition-2.html) <T, object ( <a href="https://learn.microsoft.com/dotnet/api/system.object">https://learn.microsoft.com/dotnet/api/system.object</a> ) > (AkiraNetwork.VirtualStorageLibrary. NotifyNodeDelegate List (AkiraNetwork.VirtualStorageLibrary. ( <a href="https://learn.microsoft.com/dotnet/api/system.collection.generic.list-1">https://learn.microsoft.com/dotnet/api/system.collection.generic.list-1</a> ) AkiraNetwork.VirtualStorageLibrary. < VirtualCycleDetector VirtualSortCondition (AkiraNetwork.VirtualStorageLibrary. (AkiraNetwork.VirtualStorageLibrary.VirtualSortCondition-1.html) 1.html) (AkiraNetwork.VirtualStorageLibrary. <T>VirtualDirectoryAdapter<T> (AkiraNetwork.VirtualStorageLibrary	<i>groupCondition</i>	The condition that defines how to group the elements. If null, no grouping is applied.

Returns [IList<T>](#)

Type	Description
VirtualException (AkiraNetwork.VirtualStorageLibrary. IEnumerable<VirtualGroupCondition<T, TKey>> ( <a href="https://learn.microsoft.com/dotnet/api/system.collections.generic.ienumerable-1">https://learn.microsoft.com/dotnet/api/system.collections.generic.ienumerable-1</a> ) <T>VirtualID (AkiraNetwork.VirtualStorageLibrary. VirtualItem (AkiraNetwork.VirtualStorageLibrary.	An <a href="#">IEnumerable&lt;T&gt;</a> ( <a href="https://learn.microsoft.com/dotnet/api/system.collections.generic.ienumerable-1">https://learn.microsoft.com/dotnet/api/system.collections.generic.ienumerable-1</a> ) containing the grouped and sorted elements.

Type Parameters

Name	Description
T	1.html) The type of elements in the source sequence. VirtualItem<T> (AkiraNetwork.VirtualStorageLibrary. 1.html) VirtualNode (AkiraNetwork.VirtualStorageLibrary. VirtualNodeContext (AkiraNetwork.VirtualStorageLibrary. VirtualNodeExtensions (AkiraNetwork.VirtualStorageLibrary. VirtualNodeListConditions (AkiraNetwork.VirtualStorageLibrary. VirtualnodeName (AkiraNetwork.VirtualStorageLibrary. VirtualNodeNotFoundException (AkiraNetwork.VirtualStorageLibrary. VirtualNodeType (AkiraNetwork.VirtualStorageLibrary. VirtualNodeTypeFilter

# Class VirtualStorageSettings

Manages settings for the virtual storage. This class implements the singleton pattern and holds default settings and values by type.

## Inheritance

NotifyNodeDelegate

↳ object (<https://learn.microsoft.com/dotnet/api/system.object>)  
↳ (AkiraNetwork.VirtualStorageLibrary)  
↳ VirtualStorageSettings  
PatternMatch

## Inherited Members

VirtualCycleDetector

object.Equals(object) ([https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals\(system-object\)](https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals(system-object)))  
(AkiraNetwork.VirtualStorageLibrary)

VirtualDirectory

object.Equals(object, object) ([https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals\(system-object-system-object\)](https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals(system-object-system-object)))  
(AkiraNetwork.VirtualStorageLibrary)

VirtualDirectoryAdapter<T>

object.GetHashCode() (<https://learn.microsoft.com/dotnet/api/system.object.gethashcode>)  
(AkiraNetwork.VirtualStorageLibrary)

object.GetType() (<https://learn.microsoft.com/dotnet/api/system.object.gettype>)  
1.html

object.MemberwiseClone() (<https://learn.microsoft.com/dotnet/api/system.object.memberwiseclone>)

object.ReferenceEquals(object, object) (<https://learn.microsoft.com/dotnet/api/system.object.referenceequals>)  
(AkiraNetwork.VirtualStorageLibrary)

object.ToString() (<https://learn.microsoft.com/dotnet/api/system.object.tostring>)  
VirtualGroupCondition<T, TKey>

Namespace: AkiraNetwork.AkiraNetwork.html.VirtualStorageLibrary (AkiraNetwork.VirtualStorageLibrary.html)

## Assembly

VirtualStorageLibrary.dll

## Syntax

VirtualID

(AkiraNetwork.VirtualStorageLibrary)

VirtualItem

public class VirtualStorageSettings  
(AkiraNetwork.VirtualStorageLibrary)

VirtualItemAdapter<T>

(AkiraNetwork.VirtualStorageLibrary)  
1.html

VirtualList<T>

(AkiraNetwork.VirtualStorageLibrary)  
1.html

## Properties

InvalidNodeNameCharacters

(AkiraNetwork.VirtualStorageLibrary)

Gets or sets an array of characters that are invalid in node names.

VirtualNodeContext

(AkiraNetwork.VirtualStorageLibrary)

## Declaration

VirtualNodeExtensions

(AkiraNetwork.VirtualStorageLibrary)  
public char[] InvalidNodeNameCharacters { get; set; }

VirtualNodeListConditions

(AkiraNetwork.VirtualStorageLibrary)

## Property Value

(AkiraNetwork.VirtualStorageLibrary)

Type

VirtualNodeNotFoundException

char

(AkiraNetwork.VirtualStorageLibrary)

(<https://learn.microsoft.com/dotnet/api/system.char>)

[]

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeTypeFilter

Type	Description
VirtualNodeTypeFilter	An array of characters that cannot be used in node names.

## InvalidNodeNames

Gets or sets an array of invalid node names.

## Declaration

```
    public string[] InvalidNodeNames { get; set; }
```

Property	Type	Description
Value	(AkiraNetwork.VirtualStorageLibrary.NotifyNodeDelegate)	
Type	(AkiraNetwork.VirtualStorageLibrary.PatternMatch)	An array of node names that are not allowed in the virtual storage.
string	(https://learn.microsoft.com/dotnet/api/system.string)	
VirtualCycleDetector	(AkiraNetwork.VirtualStorageLibrary.VirtualCycleDetector)	
[]	(AkiraNetwork.VirtualStorageLibrary.VirtualDirectory)	

## **NodeListConditions**

1.html)

`VirtualEnvironment` Gets or sets the conditions for listing nodes.

Declaration (Akira N)

VirtualGroupCondition<T TKey>

```
(AkiraNetwork.VirtualStorageLibrary) public VirtualNodeListConditions NodeListConditions { get; set; }
```

2. HTML

### Properties

Type	VirtualItem (AkiraNetwork.VirtualStorageLibrary)	Description
VirtualNodeListConditions (AkiraNetwork.VirtualStorageLibrary.VirtualNodeListConditions.html)		The conditions used to filter, group, and sort nodes.

(Akira)  
PathDot  
1.htm

1.11(1)

**Gets or sets the path representing the current virtual node.**

(AkiraN)

## **VirtualNodeContext**

(AkiraNetwork.VirtualStorageLibrary)

```
public String ExtensionDot { get; set; }
```

(AkiraNetwork.VirtualStorage)

**VirtualNodeListConditions**  
**Property Value**

Type	Description
<code>virtualNodeName</code> (AkiraNetwork.VirtualStorageLibrary string ( <a href="https://learn.microsoft.com/dotnet/api/system.string">https://learn.microsoft.com/dotnet/api/system.string</a> ) <code>VirtualNodeNotFoundException</code> (AkiraNetwork.VirtualStorageLibrary VirtualNodeType VirtualNodeTypeFilter	The representation of the current directory.
<code>Path</code> DotDot VirtualStorageLibrary VirtualNodeTypeFilter	

Gets or sets the path representing the parent directory. The default value is "..".

## Declaration

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

### Property Value

Type	Description
(AkiraNetwork.VirtualStorageLibrary.NotifyNodeDelegate<T>)( <a href="https://learn.microsoft.com/dotnet/api/system.string">https://learn.microsoft.com/dotnet/api/system.string</a> )	The representation of the parent directory.

## PathRoot

Gets [VirtualDirectory](#) path. The default value is "/".

(AkiraNetwork.VirtualStorageLibrary.VirtualDirectoryAdapter<T>)(<https://learn.microsoft.com/dotnet/api/system.string>)

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

### VirtualException

### Property Value

Type	Description
(AkiraNetwork.VirtualStorageLibrary.VirtualGroupCondition<T, TKey>)( <a href="https://learn.microsoft.com/dotnet/api/system.string">https://learn.microsoft.com/dotnet/api/system.string</a> )	The root directory path in the virtual storage.

## PathSeparator

Gets or sets the path separator character. The default value is '/'.  
(AkiraNetwork.VirtualStorageLibrary.VirtualItemAdapter<T>)

### Declaration

```
VirtualItem<T>(AkiraNetwork.VirtualStorageLibrary.VirtualItemAdapter<T>)(https://learn.microsoft.com/dotnet/api/system.string)
```

### VirtualNode

### Property Value

Type	Description
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeContext)( <a href="https://learn.microsoft.com/dotnet/api/system.char">https://learn.microsoft.com/dotnet/api/system.char</a> )	The character used to separate path segments.

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeListConditions)(<https://learn.microsoft.com/dotnet/api/system.char>)

Gets or sets the prefix of [VirtualStorageLibrary](#) names. The default value is "dir". Used for auto-generating node names.

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeNotFoundException)(<https://learn.microsoft.com/dotnet/api/system.char>)

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

### VirtualNodeTypeFilter

## Property Value

Type	Description
string ( <a href="https://learn.microsoft.com/dotnet/api/system.string">https://learn.microsoft.com/dotnet/api/system.string</a> )	The prefix added to auto-generated directory names.

▼

## PrefixItem

(AkiraNetwork.VirtualStorageLibrary)

NotifyNodeDelegate

Gets or sets the prefix used for item names. The default value is "item". Used for auto-generating node names.  
(AkiraNetwork.VirtualStorageLibrary)

### Declaration

(AkiraNetwork.VirtualStorageLibrary)

VirtualCycleDetector

public string PrefixItem { get; set; }

(AkiraNetwork.VirtualStorageLibrary)

VirtualDirectory

## Property Value

(AkiraNetwork.VirtualStorageLibrary)

VirtualDirectoryAdapter<T>

(AkiraNetwork.VirtualStorageLibrary)

string  
(<https://learn.microsoft.com/dotnet/api/system.string>)

Description

The prefix added to auto-generated item names.

VirtualException

(AkiraNetwork.VirtualStorageLibrary)

VirtualGroupCondition<T, TKey>

(AkiraNetwork.VirtualStorageLibrary)

Gets or sets the prefix used for symbolic link names. The default value is "link". Used for auto-generating node names.  
VirtualID

(AkiraNetwork.VirtualStorageLibrary)

### Declaration

VirtualItem

(AkiraNetwork.VirtualStorageLibrary)

public string PrefixSymbolicLink { get; set; }

(AkiraNetwork.VirtualStorageLibrary)

1.html

## Property Value

VirtualItem<T>

(AkiraNetwork.VirtualStorageLibrary)

Description

1.html

string

VirtualNode

(<https://learn.microsoft.com/dotnet/api/system.string>)

The prefix added to auto-generated symbolic link names.

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeContext

(AkiraNetwork.VirtualStorageLibrary)

## Settings

(AkiraNetwork.VirtualStorageLibrary)

Gets the current instance of the settings.

VirtualNodeListConditions

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeName

(AkiraNetwork.VirtualStorageLibrary)

public static VirtualStorageSettings Settings { get; }

VirtualNodeNotFoundException

(AkiraNetwork.VirtualStorageLibrary)

## Property Value

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeTypeFilter



# Class VirtualStorageState

Manages the state of the virtual storage. This class implements the singleton pattern and maintains the current setting of [FilterAndState](#).

## Inheritance

NotifyNodeDelegate

↳ object (<https://learn.microsoft.com/dotnet/api/system.object>)  
↳ [\(AkiraNetwork.VirtualStorageLibrary\)](#)  
↳ [VirtualStorageState](#)  
↳ [PatternMatch](#)

## Inherited Members

VirtualCycleDetector

object.Equals(object) ([https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals\(system-object\)](https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals(system-object)))  
↳ [\(AkiraNetwork.VirtualStorageLibrary\)](#)

VirtualDirectory

object.Equals(object, object) ([https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals\(system-object-system-object\)](https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals(system-object-system-object)))  
↳ [VirtualDirectoryAdapter<T>](#)

object.GetHashCode() (<https://learn.microsoft.com/dotnet/api/system.object.gethashcode>)

object.GetType() (<https://learn.microsoft.com/dotnet/api/system.object.gettype>)  
↳ [1.html](#)

object.MemberwiseClone() (<https://learn.microsoft.com/dotnet/api/system.object.memberwiseclone>)

object.ReferenceEquals(object, object) (<https://learn.microsoft.com/dotnet/api/system.object.referenceequals>)  
↳ [\(AkiraNetwork.VirtualStorageLibrary\)](#)

object.ToString() (<https://learn.microsoft.com/dotnet/api/system.object.tostring>)  
↳ [VirtualGroupCondition<T, TKey>](#)

**Namespace** [AkiraNetwork](#) [AkiraNetwork.html](#).VirtualStorageLibrary ([AkiraNetwork.VirtualStorageLibrary.html](#))

## Assembly

[VirtualStorageLibrary.dll](#)

## Syntax

VirtualID

(AkiraNetwork.VirtualStorageLibrary)

VirtualItem

public class VirtualStorageState  
↳ [\(AkiraNetwork.VirtualStorageLibrary\)](#)

VirtualItemAdapter<T>

(AkiraNetwork.VirtualStorageLibrary)

1.html)

VirtualItem<T>

(AkiraNetwork.VirtualStorageLibrary)

1.html)

## Properties

(AkiraNetwork.VirtualStorageLibrary)

## InvalidNodeNameCharacters

(AkiraNetwork.VirtualStorageLibrary)

Gets or sets an array of characters that are invalid in node names.

VIRTUALNODECONTEXT

(AkiraNetwork.VirtualStorageLibrary)

## Declaration

VirtualNodeExtensions

(AkiraNetwork.VirtualStorageLibrary)

public char[] InvalidNodeNameCharacters { get; set; }  
↳ [VirtualNodeListConditions](#)

(AkiraNetwork.VirtualStorageLibrary)

## Property Value

(AkiraNetwork.VirtualStorageLibrary)

### Type

VirtualNodeNotFoundException

### Description

char (<https://learn.microsoft.com/dotnet/api/system.char>)

[] VirtualNodeType

An array of characters that are invalid in node names.

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeTypeFilter

## InvalidNodeNames

Gets or sets an array of invalid node names.

### Declaration

```
public string[] InvalidNodeNames { get; set; }
```

**Property Value**  
(AkiraNetwork.VirtualStorageLibrary)

NotifyNodeDelegate

**Type** AkiraNetwork.VirtualStorageLibrary

PatternMatch  
string (<https://learn.microsoft.com/dotnet/api/system.string>)[]  
(AkiraNetwork.VirtualStorageLibrary)

VirtualCycleDetector

(AkiraNetwork.VirtualStorageLibrary)

## NodeListConditions

(AkiraNetwork.VirtualStorageLibrary)  
Gets or sets the conditions for listing nodes.

VirtualDirectoryAdapter<T>

**Declaration**

1.html

VirtualException

```
public VirtualNodeListConditions NodeListConditions { get; set; }
```

(AkiraNetwork.VirtualStorageLibrary)

VirtualGroupCondition<T, TKey>

**Property Value**  
(AkiraNetwork.VirtualStorageLibrary)

2.html

VirtualID

VirtualNodeListConditions

(AkiraNetwork.VirtualStorageLibrary)

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeListConditions.html)

**Description**

The conditions for listing nodes.

## PathDot

1.html

VirtualItem<T>

Gets or sets the path representing the current directory. The default value is ":".

(AkiraNetwork.VirtualStorageLibrary)

**Declaration**

1.html

VirtualNode

path (AkiraNetwork.VirtualStorageLibrary)

VirtualNodeContext

(AkiraNetwork.VirtualStorageLibrary)

**Property Value**  
(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeExtensions

**Type** AkiraNetwork.VirtualStorageLibrary

**Description**

VirtualNodeListConditions  
string (<https://learn.microsoft.com/dotnet/api/system.string>)

The path representing the current directory.

VirtualNodeName

(AkiraNetwork.VirtualStorageLibrary)

**PathDotDot** NotFoundException

(AkiraNetwork.VirtualStorageLibrary)

Gets or sets the path representing the parent directory. The default value is "..".

VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary)

**Declaration**

VirtualNodeTypeFilter

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

#### Property Value

Type	Description
string ( <a href="https://learn.microsoft.com/dotnet/api/system.string">https://learn.microsoft.com/dotnet/api/system.string</a> )	The path representing the parent directory.

#### PathRoot

Gets or sets the root path. The default value is "/".

VirtualCycleDetector

(AkiraNetwork.VirtualStorageLibrary)

Declaration

```
(AkiraNetwork.VirtualStorageLibrary)
```

VirtualDirectory

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

VirtualDirectoryAdapter<T>

(AkiraNetwork.VirtualStorageLibrary)

#### Property Value

1.html

Type	Description
virtualException	

(AkiraNetwork.VirtualStorageLibrary)

string (<https://learn.microsoft.com/dotnet/api/system.string>)

VirtualGroupCondition<T, TKey>

(AkiraNetwork.VirtualStorageLibrary)

2.html

#### PathSeparator

(AkiraNetwork.VirtualStorageLibrary)

Gets or sets the path separator character. The default value is '/'.

VirtualItem

(AkiraNetwork.VirtualStorageLibrary)

VirtualItemAdapter<T>

(AkiraNetwork.VirtualStorageLibrary)

```
public char PathSeparator { get; set; }
```

1.html

VirtualItem<T>

(AkiraNetwork.VirtualStorageLibrary)

#### Property Value

1.html

Type	Description
VirtualNode	

char (<https://learn.microsoft.com/dotnet/api/system.char>)

VirtualNodeContext

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeExtensions

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeListConditions

Gets or sets the prefix used for directory names. The default value is "dir". Used for auto-generating node names.

(AkiraNetwork.VirtualStorageLibrary)

Declaration

NodeName

(AkiraNetwork.VirtualStorageLibrary)

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

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeTypeFilter

Type	Description
string ( <a href="https://learn.microsoft.com/dotnet/api/system.string">https://learn.microsoft.com/dotnet/api/system.string</a> )	The prefix used for directory names.

## PrefixItem

Gets or sets the prefix used for item names. The default value is "item". Used for auto-generating node names.  
 (AkiraNetwork.VirtualStorageLibrary)

### Declaration

(AkiraNetwork.VirtualStorageLibrary)

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

(AkiraNetwork.VirtualStorageLibrary)

### Property Value

(AkiraNetwork.VirtualStorageLibrary)

#### Type

#### Description

VirtualDirectory  
 (AkiraNetwork.VirtualStorageLibrary)

The prefix used for item names.

string (<https://learn.microsoft.com/dotnet/api/system.string>)

(AkiraNetwork.VirtualStorageLibrary)

1.html)

## PrefixSymbolicLink

(AkiraNetwork.VirtualStorageLibrary)

Gets or sets the prefix used for symbolic link names. The default value is "link". Used for auto-generating node names.  
 (AkiraNetwork.VirtualStorageLibrary)

2.html)

### Declaration

VirtualID

(AkiraNetwork.VirtualStorageLibrary)

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

(AkiraNetwork.VirtualStorageLibrary)

### Property Value

(AkiraNetwork.VirtualStorageLibrary)

#### Type

#### Description

.html)

The prefix used for symbolic link names.

string (<https://learn.microsoft.com/dotnet/api/system.string>)

(AkiraNetwork.VirtualStorageLibrary)

1.html)

## State

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeContext

Gets the current instance of the state.  
 (AkiraNetwork.VirtualStorageLibrary)

### Declaration

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeListConditions

```
public static VirtualStorageState State { get; }
```

(AkiraNetwork.VirtualStorageLibrary)

### Property Value

(AkiraNetwork.VirtualStorageLibrary)

#### Type

#### Description

VirtualNodeNotFoundException  
 (AkiraNetwork.VirtualStorageLibrary)

VirtualStorageState ([AkiraNetwork.VirtualStorageLibrary.VirtualStorageState.html](#))

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeTypeFilter

## WildcardMatcher

Gets or sets the wildcard matcher.

### Declaration

```
public IVirtualWildcardMatcher? WildcardMatcher { get; set; }
```

Property	Value	Description
Type	(AkiraNetwork.VirtualStorageLibrary) NotifyNodeDelegate	
IVirtualWildcardMatcher	(AkiraNetwork.VirtualStorageLibrary) (AkiraNetwork.VirtualStorageLibrary.IVirtualWildcardMatcher.html)	The wildcard matcher.

## Methods

(AkiraNetwork.VirtualStorageLibrary) VirtualDirectoryAdapter<T>

(AkiraNetwork.VirtualStorageLibrary) VirtualDirectoryAdapter<T>(1.html)

### Set NodeListConditions(Virtual NodeListConditions)

Sets the conditions for listing nodes.

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

Declaration (AkiraNetwork.VirtualStorageLibrary) VirtualItem

(AkiraNetwork.VirtualStorageLibrary) VirtualItem(2.html)

```
public static void Set NodeListConditions(Virtual NodeListConditions conditions)
```

(AkiraNetwork.VirtualStorageLibrary) VirtualItem

(AkiraNetwork.VirtualStorageLibrary) VirtualItemAdapter<T>(VirtualItem.html)

Parameters (AkiraNetwork.VirtualStorageLibrary) VirtualItemAdapter<T>

Type	Name	Description
(AkiraNetwork.VirtualStorageLibrary) VirtualItemAdapter<T>(VirtualItem.html)	conditions	The conditions for listing nodes.

(AkiraNetwork.VirtualStorageLibrary) VirtualItem(1.html)

### Set NodeListConditions(Virtual NodeTypeFilter,

(AkiraNetwork.VirtualStorageLibrary) VirtualGroupCondition<VirtualNode, object>?,

List<VirtualSortCondition<VirtualNode>>?)

(AkiraNetwork.VirtualStorageLibrary) VirtualNodeContext

Sets the conditions for listing nodes.

(AkiraNetwork.VirtualStorageLibrary) VirtualNodeContext(1.html)

Declaration (AkiraNetwork.VirtualStorageLibrary) VirtualNodeListConditions

(AkiraNetwork.VirtualStorageLibrary) VirtualNodeListConditions

```
public static void Set NodeListConditions(Virtual NodeTypeFilter filter, VirtualGroupCondition<VirtualNode, object>? groupCondition = null, List<VirtualSortCondition<VirtualNode>>? sortC
```

```
onditions = null)
```

(AkiraNetwork.VirtualStorageLibrary) VirtualNotFoundException

(AkiraNetwork.VirtualStorageLibrary) VirtualNotFoundException(1.html)

Parameters (AkiraNetwork.VirtualStorageLibrary) VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary) VirtualNodeTypeFilter

Type	Name	Description
VirtualNodeTypeFilter (AkiraNetwork.VirtualStorageLibrary.VirtualNodeTypeFilter.html)	<i>filter</i>	The filter for node types.
VirtualGroupCondition (AkiraNetwork.VirtualStorageLibrary.VirtualGroupCondition-2.html) <VirtualNode (AkiraNetwork.VirtualStorageLibrary.VirtualNode.html), object (https://learn.microsoft.com/dotnet/api/system.object)> (AkiraNetwork.VirtualStorageLibrary)	<i>groupCondition</i>	The condition for grouping nodes.
NotifyNodeDelegate List (https://learn.microsoft.com/dotnet/api/system.collections.generic.list-1) (AkiraNetwork.VirtualStorageLibrary) < PatternMatch (AkiraNetwork.VirtualStorageLibrary) VirtualSortCondition VirtualCycleDetector (AkiraNetwork.VirtualStorageLibrary) <VirtualNode (AkiraNetwork.VirtualStorageLibrary.VirtualNode.html)>> VirtualDirectory (AkiraNetwork.VirtualStorageLibrary) VirtualDirectoryAdapter<T> (AkiraNetwork.VirtualStorageLibrary)	<i>sortConditions</i>	The conditions for sorting nodes.

## Extension Methods

VirtualTextFormatter.GenerateSingleTableDebugText<T>(T)  
(AkiraNetwork.VirtualStorageLibrary.Utilities.VirtualTextFormatter.html#AkiraNetwork\_VirtualStorageLibrary\_Utils  
s\_VirtualTextFormatter\_GenerateSingleTableDebugText\_1\_\_0\_)

VirtualGroupCondition<T, TKey>  
(AkiraNetwork.VirtualStorageLibrary)

VirtualItem  
(AkiraNetwork.VirtualStorageLibrary)

VirtualItemAdapter<T>  
(AkiraNetwork.VirtualStorageLibrary)

VirtualItem<T>  
(AkiraNetwork.VirtualStorageLibrary)

VirtualNode  
(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeContext  
(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeExtensions  
(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeListConditions  
(AkiraNetwork.VirtualStorageLibrary)

VirtualnodeName  
(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeNotFoundException  
(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeType  
(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeTypeFilter  
(AkiraNetwork.VirtualStorageLibrary)

# Class VirtualStorage<T>

## Inheritance

↳ [Object](#) (<https://learn.microsoft.com/dotnet/api/system.object>)  
↳ [VirtualStorage<T>](#)  
(AkiraNetwork.VirtualStorageLibrary)

## Inherited Members

NotifyNodeDelegate

object.Equals(object) ([https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals\(system-object-system-object\)](https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals(system-object-system-object)))  
PatternMatch  
(AkiraNetwork.VirtualStorageLibrary)  
object.Equals(object) ([https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals\(system-object-system-object\)](https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals(system-object-system-object)))  
VirtualCycleDetector  
(AkiraNetwork.VirtualStorageLibrary)  
object.GetHashCode() (<https://learn.microsoft.com/dotnet/api/system.object.gethashcode>)  
object.GetType() (<https://learn.microsoft.com/dotnet/api/system.object.gettype>)  
object.MemberwiseClone()  
(AkiraNetwork.VirtualStorageLibrary)  
object.ReferenceEquals(object1, object2) (<https://learn.microsoft.com/dotnet/api/system.object.referenceequals>)  
object.ToString() (<https://learn.microsoft.com/dotnet/api/system.object.tostring>)  
1.html)

**Namespace:** AkiraNetwork (AkiraNetwork.html).VirtualStorageLibrary (AkiraNetwork.VirtualStorageLibrary.html)  
VirtualException

**Assembly:** VirtualStorageLibrary.dll

**Syntax:** VirtualGroupCondition<T, TKey>

(AkiraNetwork.VirtualStorageLibrary)

public class VirtualStorage<T>

VirtualID

(AkiraNetwork.VirtualStorageLibrary)

**Type Parameters:**

VirtualItem

Name	Description
VirtualItemAdapter<T>	
T	(AkiraNetwork.VirtualStorageLibrary)

1.html)

VirtualItem<T>

## Constructors

(AkiraNetwork.VirtualStorageLibrary)

1.html)

VirtualNode

**VirtualStorage()**

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeContext

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeExtensions

public VirtualStorage() (<https://learn.microsoft.com/dotnet/api/a>)

VirtualNodeListConditions

(AkiraNetwork.VirtualStorageLibrary)

VirtualnodeName

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeNotFoundException

(AkiraNetwork.VirtualStorageLibrary)

## Properties

VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary)

Declaration

VirtualNodeTypeFilter

```
public VirtualPath CurrentPath { get; }
```

## Property Value

Type	Description
VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	(AkiraNetwork.VirtualStorageLibrary.NotifyNodeDelegate (AkiraNetwork.VirtualStorageLibrary.PatternMatch))

## CycleDetectorForTarget

Declaration  
AkiraNetwork.VirtualStorageLibrary.VirtualCycleDetector  
VirtualCycleDetector  
(AkiraNetwork.VirtualStorageLibrary.VirtualCycleDetector.CycleDetectorForTarget { get; })  
VirtualDirectory  
(AkiraNetwork.VirtualStorageLibrary.VirtualDirectoryAdapter<T>)

## Property Value

Type	Description
(AkiraNetwork.VirtualStorageLibrary.VirtualCycleDetector 1.html) VirtualCycleDetector (AkiraNetwork.VirtualStorageLibrary.VirtualCycleDetector.html)	(AkiraNetwork.VirtualStorageLibrary.VirtualGroupCondition<T, TKey> (AkiraNetwork.VirtualStorageLibrary.VirtualGroupCondition 2.html))

## CycleDetectorForTree

Declaration  
VirtualItem  
VirtualCycleDetector CycleDetectorForTree { get; }  
(AkiraNetwork.VirtualStorageLibrary.VirtualItemAdapter<T>)

## Property Value

Type	Description
VirtualItem<T> VirtualCycleDetector (AkiraNetwork.VirtualStorageLibrary.VirtualCycleDetector.html)	1.html VirtualNode

## Dir

Declaration  
VirtualNodeContext  
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeExtensions & VirtualNodeListConditions)

## Property Value

Type	Description
AkiraNetwork.VirtualStorageLibrary.VirtualNodeName VirtualNodeNotFoundException VirtualDirectoryAdapter (AkiraNetwork.VirtualStorageLibrary.VirtualDirectoryAdapter-1.html)<T> (AkiraNetwork.VirtualStorageLibrary.VirtualNodeType)	VirtualNodeContext Dir { get; } VirtualNodeListConditions (AkiraNetwork.VirtualStorageLibrary.VirtualNodeTypeFilter)

## this[VirtualPath, bool]

### Declaration

```
public VirtualNode this[VirtualPath path, bool followLinks = true] { get; set; }
```



### Parameters

Type	Name	Description
NotifyNodeDelegate (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	path	
bool ( <a href="https://learn.microsoft.com/dotnet/api/system.boolean">https://learn.microsoft.com/dotnet/api/system.boolean</a> ) (AkiraNetwork.VirtualStorageLibrary.VirtualCycleDetector)	followLinks	

### Property Value

(AkiraNetwork.VirtualStorageLibrary.VirtualDirectory)

Type	Description
virtualDirectory (AkiraNetwork.VirtualStorageLibrary.VirtualNode) VirtualNodeAdapter<T> (AkiraNetwork.VirtualStorageLibrary.VirtualDirectoryAdapter<T> 1.html)	

### Item

(AkiraNetwork.VirtualStorageLibrary.VirtualException)

### Declaration

```
VirtualGroupCondition<T, TKey>  
(AkiraNetwork.VirtualStorageLibrary.VirtualItemAdapter<T> Item { get; }  
2.html)
```

VirtualID

(AkiraNetwork.VirtualStorageLibrary.VirtualItem)

Type	Description
VirtualItemAdapter<T> VirtualItemAdapter<T>(AkiraNetwork.VirtualStorageLibrary.VirtualItemAdapter-1.html)<T> (AkiraNetwork.VirtualStorageLibrary.VirtualItemAdapter-1.html)	

1.html)

VirtualItem<T>

### Link

(AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink)

1.html)

### Declaration

```
VirtualNode  
(AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLinkAdapter<T> Link { get; }  
VirtualNodeContext)
```

(AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink)

### Property Value

(AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink)

Type	Description
VirtualNodeListConditions VirtualNodeName <T>(AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLinkAdapter-1.html)	

VirtualNodeNotFoundException

(AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink)

VirtualNodeType

### LinkDictionary

(AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink)

VirtualNodeTypeFilter

## Declaration

```
public Dictionary<VirtualPath, HashSet<VirtualPath>> LinkDictionary { get; }
```

### Property Value

Type	Description
Dictionary<AkiraNetwork.VirtualStorageLibrary.VirtualPath, HashSet<VirtualPath>> (https://learn.microsoft.com/dotnet/api/system.collections.generic.dictionary-2)< VirtualPath> (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html), HashSet<AkiraNetwork.VirtualStorageLibrary.VirtualPath> (https://learn.microsoft.com/dotnet/api/system.collections.generic.hashset-1)< VirtualPath> (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)>>	

### Root

VirtualDirectory

Declaration

```
VirtualDirectoryAdapter<T>
(AkiraNetwork.VirtualStorageLibrary.VirtualDirectoryRoot { get; } 1.html)
```

VirtualException

### Property Value

AkiraNetwork.VirtualStorageLibrary.VirtualDirectory

Type VirtualGroupCondition<T, TKey>

```
(AkiraNetwork.VirtualStorageLibrary.VirtualGroupCondition<T, TKey> 1.html)
```

VirtualDirectory (AkiraNetwork.VirtualStorageLibrary.VirtualDirectory.html)

VirtualID

(AkiraNetwork.VirtualStorageLibrary.VirtualID 1.html)

VirtualItem

AkiraNetwork.VirtualStorageLibrary.VirtualItem

VirtualItemAdapter<T>

(AkiraNetwork.VirtualStorageLibrary.VirtualItemAdapter<T> 1.html)

## Methods

### AddDirectory(VirtualPath, VirtualDirectory, bool)

VirtualItem<T>

Declaration

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

VirtualNode AddDirectory(VirtualPath directoryPath, VirtualDirectory directory, bool createsSubdirectories, bool filter) (AkiraNetwork.VirtualStorageLibrary.VirtualNode 1.html)

VirtualNodeContext

AkiraNetwork.VirtualStorageLibrary.VirtualNodeContext

VirtualNodeExtensions

Type AkiraNetwork.VirtualStorageLibrary.VirtualNodeExtensions

Name

Description

VirtualNodeListConditions

directoryPath

VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

(AkiraNetwork.VirtualStorageLibrary.VirtualPath 1.html)

VirtualDirectoryByName

directory

(AkiraNetwork.VirtualStorageLibrary.VirtualDirectoryByName 1.html)

bool (https://learn.microsoft.com/dotnet/api/system.boolean)

createSubdirectories

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeCreateSubdirectories 1.html)

VirtualNodeType

VirtualNodeTypeFilter

VirtualNodeNotFoundException

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeNotFoundException 1.html)

VirtualNodeNotTypeException

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeNotTypeException 1.html)

### AddDirectory(VirtualPath, bool)

VirtualNodeTypeFilter

## Declaration

```
public void AddDirectory(VirtualPath path, bool createSubdirectories = false)
```

### Parameters

Type	Name	Description
VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	path	
bool ( <a href="https://learn.microsoft.com/dotnet/api/system.boolean">https://learn.microsoft.com/dotnet/api/system.boolean</a> ) (AkiraNetwork.VirtualStorageLibrary)	createSubdirectories	

## AddItem(VirtualPath, VirtualItem<T>, bool)

(AkiraNetwork.VirtualStorageLibrary)

### Declaration

```
VirtualDirectory  
    (AkiraNetwork.VirtualStorageLibrary)  
VirtualDirectoryAdapter  
    (AkiraNetwork.VirtualStorageLibrary)  
VirtualPath AddItem(VirtualPath itemDirectoryPath, VirtualItem<T> item, bool overwrite = false)  
    (AkiraNetwork.VirtualStorageLibrary)  
        1.html)
```

### Parameters

Type	Name	Description
VirtualGroupCondition<T, TKey>		
VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) 2.html)	itemDirectoryPath	
VirtualItem<T> (AkiraNetwork.VirtualStorageLibrary.VirtualItem-1.html)<T> bool ( <a href="https://learn.microsoft.com/dotnet/api/system.boolean">https://learn.microsoft.com/dotnet/api/system.boolean</a> ) VirtualItem (AkiraNetwork.VirtualStorageLibrary) VirtualItemAdapter<T>	item	
	overwrite	

## AddItem(VirtualPath, T?, bool)

### Declaration

```
VirtualItem<T>  
    (AkiraNetwork.VirtualStorageLibrary)  
public void AddItem(VirtualPath itemPath, T? data = default, bool overwrite = false)  
    1.html)
```

### Parameters

Type	Name	Description
VirtualNode (AkiraNetwork.VirtualStorageLibrary)		
VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) (AkiraNetwork.VirtualStorageLibrary)	itemPath	
T VirtualNodeListConditions (AkiraNetwork.VirtualStorageLibrary) bool ( <a href="https://learn.microsoft.com/dotnet/api/system.boolean">https://learn.microsoft.com/dotnet/api/system.boolean</a> ) VirtualNodeName (AkiraNetwork.VirtualStorageLibrary) VirtualNodeNotFoundException	data	
	overwrite	

## AddLinkToDictionary(VirtualPath, VirtualPath)

### Declaration

```
VirtualNodeType  
    (AkiraNetwork.VirtualStorageLibrary)  
VirtualNodeTypeFilter
```

```
public void AddLinkToDictionary(VirtualPath targetPath, VirtualPath linkPath)
```

## Parameters

Type	Name	Description
VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	targetPath	
(AkiraNetwork.VirtualStorageLibrary.VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)) NotifyNodeDelegate	linkPath	

(AkiraNetwork.VirtualStorageLibrary.

PatternMatch

## AddNode(VirtualPath, VirtualNode, bool)

VirtualCycleDetector

Declaration  
(AkiraNetwork.VirtualStorageLibrary.

VirtualDirectory

```
public void AddNode(VirtualPath nodeDirectoryPath, VirtualNode node, bool overwrite = false)
```

VirtualDirectoryAdapter<T>

(AkiraNetwork.VirtualStorageLibrary.  
1.html)

Type VirtualException

Name

Description

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

nodeDirectoryPath

VirtualNode (AkiraNetwork.VirtualStorageLibrary.VirtualNode.html)

node

bool (https://learn.microsoft.com/dotnet/api/system.boolean)  
VirtualID

overwrite

(AkiraNetwork.VirtualStorageLibrary.

VirtualItem

## AddSymbolicLink(VirtualPath, VirtualPath?, bool)

VirtualItemAdapter<T>

Declaration  
(AkiraNetwork.VirtualStorageLibrary.

1.html)

```
public void AddSymbolicLink(VirtualPath linkPath, VirtualPath? targetPath = null, bool overwrite = false)
```

(AkiraNetwork.VirtualStorageLibrary.

1.html)

VirtualNode

(AkiraNetwork.VirtualStorageLibrary.

Type VirtualNodeContext

Name

Description

(AkiraNetwork.VirtualStorageLibrary.  
VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html))  
VirtualNodeExtensions

linkPath

VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

targetPath

bool (https://learn.microsoft.com/dotnet/api/system.boolean)  
(AkiraNetwork.VirtualStorageLibrary.

overwrite

VirtualnodeName

(AkiraNetwork.VirtualStorageLibrary.

## AddSymbolicLink(VirtualPath, VirtualSymbolicLink, bool)

(AkiraNetwork.VirtualStorageLibrary.

Declaration  
VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary.

VirtualNodeTypeFilter

```
public void AddSymbolicLink(VirtualPath linkDirectoryPath, VirtualSymbolicLink link, bool overwrite = false)
```

## Parameters

Type	Name	Description
VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	<i>linkDirectoryPath</i>	
NotifyNodeDelegate VirtualSymbolicLink (AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink.html) PatternMatch	<i>link</i>	
bool (AkiraNetwork.VirtualStorageLibrary.net/api/system.boolean)	<i>overwrite</i>	

## ChangeDirectory(VirtualPath)

### Declaration

```
VirtualDirectoryAdapter<T>
(AkiraNetwork.VirtualStorageLibrary)
1.html) public void ChangeDirectory(VirtualPath path)
VirtualException
(AkiraNetwork.VirtualStorageLibrary)
```

### Parameters

#### VirtualGroupCondition<T, TKey>

Type	Name	Description
VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	<i>path</i>	

## ConvertToAbsolutePath(VirtualPath?, VirtualPath?)

### Declaration

```
AkiraNetwork.VirtualStorageLibrary
1.html)
VirtualItem<T> ConvertToAbsolutePath(VirtualPath? relativePath, VirtualPath? basePath =
n(AkiraNetwork.VirtualStorageLibrary)
1.html)
```

### Parameters

#### VirtualNodeContext

Type	Name	Description
VirtualNodeContext (AkiraNetwork.VirtualStorageLibrary)	<i>relativePath</i>	

#### VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

Type	Name	Description
VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	<i>basePath</i>	

#### VirtualNodeListConditions

Type	Description
VirtualNodeNotFoundException VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	

#### VirtualNodeType

Type	Description
VirtualNodeType (AkiraNetwork.VirtualStorageLibrary)	

## CopyNode(VirtualPath, VirtualPath, bool, bool, List<VirtualNodeContext>?)

## Declaration

```
public void CopyNode(VirtualPath sourcePath, VirtualPath destinationPath, bool overwrite = false, bool recursive = false, bool followLinks = false, List<VirtualNodeContext>? destinationContextList = null)
```



## Parameters

Type	Name	Description
NotifyNodeDelegate		
VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	sourcePath	
PatternMatch	destinationPath	
bool (https://learn.microsoft.com/dotnet/api/system.boolean)	overwrite	
bool (https://learn.microsoft.com/dotnet/api/system.boolean)	recursive	
bool (https://learn.microsoft.com/dotnet/api/system.boolean)	followLinks	
VirtualDirectoryAdapter<T> List (https://learn.microsoft.com/dotnet/api/system.collections.generic.list-1) (AkiraNetwork.VirtualStorageLibrary.VirtualException < VirtualException VirtualNodeContext (AkiraNetwork.VirtualStorageLibrary.VirtualNodeContext.html) (AkiraNetwork.VirtualStorageLibrary.VirtualGroupCondition<T, TKey> > (AkiraNetwork.VirtualStorageLibrary.VirtualID 2.html) VirtualID (AkiraNetwork.VirtualStorageLibrary.VirtualItem VirtualItem VirtualItemAdapter<T> p(AkiraNetwork.VirtualStorageLibrary.VirtualPath path, bool followLinks = false) 1.html)	destinationContextList	

## DirectoryExists(VirtualPath, bool)

### Declaration

```
(AkiraNetwork.VirtualStorageLibrary.VirtualPath  
VirtualItem  
VirtualItemAdapter<T>  
p(AkiraNetwork.VirtualStorageLibrary.VirtualPath path, bool followLinks = false)  
1.html)
```

### Parameters

Type	Name	Description
.html)		
VirtualNode VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) (AkiraNetwork.VirtualStorageLibrary.VirtualException VirtualItem<T>	path	
bool (https://learn.microsoft.com/dotnet/api/system.boolean)	followLinks	

### Returns

Type	Description
VirtualNodeListConditions	

## ExpandPath(VirtualPath, VirtualNodeTypeFilter, bool, bool)

### Declaration

```
(AkiraNetwork.VirtualStorageLibrary.VirtualPath  
VirtualNodeType  
VirtualNodeTypeFilter  
(AkiraNetwork.VirtualStorageLibrary.VirtualPath  
VirtualNodeTypeFilter)
```

```
public IEnumerable<VirtualPath> ExpandPath(VirtualPath path, VirtualNodeTypeFilter filter =
VirtualNodeTypeFilter.All, bool followLinks = true, bool resolveLinks = true)
```

## Parameters

Type	Name	Description
VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	path	
NotifyNodeDelegate VirtualNodeTypeFilter (AkiraNetwork.VirtualStorageLibrary.VirtualNodeTypeFilter.html) PatternMatch	filter	
bool (AkiraNetwork.VirtualStorageLibrary (https://learn.microsoft.com/dotnet/api/system.boolean))	followLinks	
VirtualCycleDetector bool (https://learn.microsoft.com/dotnet/api/system.boolean) (AkiraNetwork.VirtualStorageLibrary)	resolveLinks	

## VirtualDirectory

### Returns

AkiraNetwork.VirtualStorageLibrary

Type	Description
VirtualDirectoryAdapter<T>	
(AkiraNetwork.VirtualStorageLibrary)	
IEnumerable (https://learn.microsoft.com/dotnet/api/system.collections.generic.ienumerable-1)< VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)>	
VirtualException	

## ExpandPathTree(VirtualPath, VirtualNodeTypeFilter, bool, bool)

### Declaration

```
virtual IEnumerable<VirtualNodeContext> ExpandPathTree(VirtualPath path, VirtualNodeTypeFilte
r (AkiraNetwork.VirtualStorageLibrary.VirtualNodeTypeFilter.All, bool followLinks = true, bool resolveLinks = true)
```

### Parameters

VirtualItem  
VirtualItemAdapter<T>  
(AkiraNetwork.VirtualStorageLibrary)

Type	Name	Description
VirtualItem<T>		
(AkiraNetwork.VirtualStorageLibrary)		
VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	path	
VirtualNodeTypeFilter	filter	
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeTypeFilter.html)		
VirtualNodeContext		
bool (https://learn.microsoft.com/dotnet/api/system.boolean)	followLinks	
VirtualNodeExtension		
bool (https://learn.microsoft.com/dotnet/api/system.boolean)	resolveLinks	

### Returns

VirtualNodeListConditions  
(AkiraNetwork.VirtualStorageLibrary)

Type	Description
VirtualNodeName	
(AkiraNetwork.VirtualStorageLibrary)	
IEnumerable (https://learn.microsoft.com/dotnet/api/system.collections.generic.ienumerable-1)< VirtualNodeNotFoundException	
VirtualNodeContext (AkiraNetwork.VirtualStorageLibrary.VirtualNodeContext.html)>	

(AkiraNetwork.VirtualStorageLibrary)

## GetDirectory(VirtualPath, bool)

## Declaration

```
public VirtualDirectory GetDirectory(VirtualPath path, bool followLinks = false)
```

## Parameters

Type		Name	Description
VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)		path	
NotifyNodeDelegate bool (https://learn.microsoft.com/dotnet/api/system.boolean) (AkiraNetwork.VirtualStorageLibrary)		followLinks	

## PatternMatch

**Returns** `(AkiraNetwork.VirtualStorageLibrary)`

Type	VirtualCycleDetector (AkiraNetwork.VirtualStorageLibrary.VirtualDirectory)	Description
VirtualDirectory	(AkiraNetwork.VirtualStorageLibrary.VirtualDirectory.html)	

# .GetItem(VirtualPath, bool)

## Declaration Virtual Exception

(AkiraNetwork.VirtualStorageLibrary)

```
public VirtualUri GetItem(VirtualPath path, bool followLinks = false)
```

(AkiraNetwork.VirtualStorageLibrary)

2.html)

## Parameters

Type	Name	Description
AkiraNetwork.VirtualStorageLibrary.VirtualItem	<i>path</i>	VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) (AkiraNetwork.VirtualStorageLibrary.VirtualItem)
bool	<i>followLinks</i>	VirtualItemAdapter To (https://learn.microsoft.com/dotnet/api/system.boolean) (AkiraNetwork.VirtualStorageLibrary.VirtualItemAdapter)

Part\_1.html

**VirtualItem<T>**

Type AkiraNetwork.VirtualStorageLibrary

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

Virtual

## (AkiraNetwork.VirtualStorageLibrary)

## VirtualNodeContext

# GetLinksFromDictionary(VirtualPath)

# VirtualNodeExtensions Narration

## **Declaration**

(AkiraNetwork.VirtualStorageLibrary)

## VirtualNodeListConditions

```
public HashSet<VirtualPath> GetLinksFromDictionary(VirtualPath targetPath)  
(AkiraNetwork.VirtualStorageLibrary)
```

## VirtualNodeName

## Parameters

Type		Name	Description
VirtualNodeNotFoundException (AkiraNetwork.VirtualStorageLibrary)			
VirtualNodeType (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) (AkiraNetwork.VirtualStorageLibrary)	<i>targetPath</i>		
VirtualNodeTypeFilter			

## Returns

Type	Description
HashSet ( <a href="https://learn.microsoft.com/dotnet/api/system.collections.generic.hashset-1">https://learn.microsoft.com/dotnet/api/system.collections.generic.hashset-1</a> )<VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)>	

▼

## GetNode(VirtualPath, bool)

Declaration  
(AkiraNetwork.VirtualStorageLibrary)

PatternMatch  
public VirtualNode GetNode(VirtualPath path, bool followLinks = false)  
(AkiraNetwork.VirtualStorageLibrary)

VirtualCycleDetector  
(AkiraNetwork.VirtualStorageLibrary)

Parameters  
Type  
VirtualDirectory  
(AkiraNetwork.VirtualStorageLibrary)

VirtualDirectoryAdapter  
VirtualPath (<https://learn.microsoft.com/dotnet/api/system.string>)  
(AkiraNetwork.VirtualStorageLibrary)

bool (<https://learn.microsoft.com/dotnet/api/system.boolean>)  
(<https://learn.microsoft.com/dotnet/api/system.boolean>)

VirtualException  
Returns  
(AkiraNetwork.VirtualStorageLibrary)

Type  
VirtualGroupCondition<T, TKey>  
(AkiraNetwork.VirtualStorageLibrary)

VirtualNode (<https://learn.microsoft.com/dotnet/api/system.object>)  
(AkiraNetwork.VirtualStorageLibrary)

VirtualID  
(AkiraNetwork.VirtualStorageLibrary)

VirtualItem  
(AkiraNetwork.VirtualStorageLibrary)

## GetNodeType(VirtualPath, bool)

Declaration  
VirtualItemAdapter<T>  
(AkiraNetwork.VirtualStorageLibrary)

1.html  
public VirtualNodeType GetNodeType(VirtualPath path, bool followLinks = false)  
VirtualItem<T>  
(AkiraNetwork.VirtualStorageLibrary)

Parameters  
Type  
VirtualNode  
(AkiraNetwork.VirtualStorageLibrary)

VirtualPath (<https://learn.microsoft.com/dotnet/api/system.string>)  
(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeContext  
VirtualPathContext  
(AkiraNetwork.VirtualStorageLibrary)

bool (<https://learn.microsoft.com/dotnet/api/system.boolean>)  
(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeExtensions  
(AkiraNetwork.VirtualStorageLibrary)

Returns  
VirtualNodeListConditions  
Type  
AkiraNetwork.VirtualStorageLibrary

VirtualNodeName  
VirtualNodeType (<https://learn.microsoft.com/dotnet/api/system.object>)  
(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeNotFoundException  
(AkiraNetwork.VirtualStorageLibrary)

## GetNodes(VirtualNodeTypeFilter, bool, bool)

(AkiraNetwork.VirtualStorageLibrary)

Declaration  
VirtualNodeTypeFilter

```
public IEnumerable<VirtualNode> GetNodes(VirtualNodeTypeFilter nodeType = VirtualNodeTypeFilter.All, bool recursive = false, bool followLinks = false)
```

## Parameters

Type	Name	Description
VirtualNodeTypeFilter (AkiraNetwork.VirtualStorageLibrary.VirtualNodeTypeFilter.html)	nodeType	
bool (https://learn.microsoft.com/dotnet/api/system.boolean) PatternMatch	recursive	
bool (AkiraNetwork.VirtualStorageLibrary.dotnet/api/system.boolean)	followLinks	

## Returns

Type	Description
VirtualDirectory (AkiraNetwork.VirtualStorageLibrary)	

VirtualDirectoryAdapter  
(https://learn.microsoft.com/dotnet/api/system.collections.generic.ienumerable-1)<

VirtualNode  
(AkiraNetwork.VirtualStorageLibrary.VirtualNode.html)>  
1.html)

VirtualException  
(AkiraNetwork.VirtualStorageLibrary)

## GetNodes(VirtualPath, VirtualNodeTypeFilter, bool, bool)

### Declaration

```
public IEnumerable<VirtualNode> GetNodes(VirtualPath basePath, VirtualNodeTypeFilter nodeTyp  
(AkiraNetwork.VirtualStorageLibrary)  
e = VirtualNodeTypeFilter.All, bool recursive = false, bool followLinks = false)  
VirtualItem  
(AkiraNetwork.VirtualStorageLibrary)
```

## Parameters

Type	Name	Description
VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	basePath	
VirtualNodeTypeFilter (AkiraNetwork.VirtualStorageLibrary) 1.html)	nodeType	
bool (https://learn.microsoft.com/dotnet/api/system.boolean)	recursive	
VirtualNodeContext (AkiraNetwork.VirtualStorageLibrary) VirtualNode	followLinks	

## Return

AkiraNetwork.VirtualStorageLibrary

Type	Description
VirtualNodeListConditions (AkiraNetwork.VirtualStorageLibrary)	

IEnumerable  
(https://learn.microsoft.com/dotnet/api/system.collections.generic.ienumerable-1)<

VirtualNode  
(AkiraNetwork.VirtualStorageLibrary.VirtualNode.html)>

VirtualNodeNotFoundException  
(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary)

## GetNodesWithPaths(VirtualNodeTypeFilter, bool, bool)

### Declaration

VirtualNodeTypeFilter  
(AkiraNetwork.VirtualStorageLibrary)

```
public IEnumerable<VirtualPath> GetNodesWithPaths(VirtualNodeTypeFilter nodeType = VirtualNodeTypeFilter.All, bool recursive = false, bool followLinks = false)
```

## Parameters

Type	Name	Description
VirtualNodeTypeFilter (AkiraNetwork.VirtualStorageLibrary.VirtualNodeTypeFilter.html)	nodeType	
bool (https://learn.microsoft.com/dotnet/api/system.boolean) PatternMatch	recursive	
bool (AkiraNetwork.VirtualStorageLibrary.dotnet/api/system.boolean)	followLinks	

## Returns

Type	Description
VirtualDirectory (AkiraNetwork.VirtualStorageLibrary)	

## GetNodesWithPaths(VirtualPath, VirtualNodeTypeFilter, bool, bool)

### Declaration

```
public IEnumerable<VirtualPath> GetNodesWithPaths(VirtualPath basePath, VirtualNodeTypeFilter nodeType = VirtualNodeTypeFilter.All, bool recursive = false, bool followLinks = false)
    VirtualItem
    (AkiraNetwork.VirtualStorageLibrary)
```

### Parameters

Type	Name	Description
VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	basePath	
VirtualNodeTypeFilter (AkiraNetwork.VirtualStorageLibrary.VirtualNodeTypeFilter.html)	nodeType	
bool (https://learn.microsoft.com/dotnet/api/system.boolean)	recursive	
bool (AkiraNetwork.VirtualStorageLibrary.dotnet/api/system.boolean)	followLinks	

### Returns

Type	Description
VirtualNodeListConditions (AkiraNetwork.VirtualStorageLibrary)	

## GetSymbolicLink(VirtualPath)

### Declaration

```
VirtualNodeFilter
```

```
public VirtualSymbolicLink GetSymbolicLink(VirtualPath path)
```

#### Parameters

Type	Name	Description
VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) (AkiraNetwork.VirtualStorageLibrary)	path	

#### Returns

Type	Description
VirtualSymbolicLink (AkiraNetwork.VirtualStorageLibrary.VirtualStorageLibrary.VirtualSymbolicLink.html)	

VirtualCycleDetector  
(AkiraNetwork.VirtualStorageLibrary)

VirtualDirectory  
(AkiraNetwork.VirtualStorageLibrary)

**ItemExists(VirtualPath, bool)**  
(AkiraNetwork.VirtualStorageLibrary)

VirtualDirectoryAdapter<T>  
(AkiraNetwork.VirtualStorageLibrary)

```
1.html)
public bool ItemExists(VirtualPath path, bool followLinks = false)
```

VirtualException  
(AkiraNetwork.VirtualStorageLibrary)

#### Parameters

Type	Name	Description
VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	path	
bool (https://learn.microsoft.com/dotnet/api/system.boolean) VirtualItem (AkiraNetwork.VirtualStorageLibrary)	followLinks	

VirtualNode  
(AkiraNetwork.VirtualStorageLibrary)

**MoveNode(VirtualPath, VirtualPath, bool, bool)**  
(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeContext  
(AkiraNetwork.VirtualStorageLibrary)

```
VirtualNodeExtensions(VirtualPath sourcePath, VirtualPath destinationPath, bool overwrite = false)
```

VirtualNodeListConditions  
(AkiraNetwork.VirtualStorageLibrary)

**Parameters**  
VirtualNodeName  
(AkiraNetwork.VirtualStorageLibrary)

Type	Name	Description
VirtualNodeNotFoundException VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) (AkiraNetwork.VirtualStorageLibrary)	sourcePath	
VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) (AkiraNetwork.VirtualStorageLibrary)	destinationPath	
bool (https://learn.microsoft.com/dotnet/api/system.boolean) VirtualNodeTypeFilter (AkiraNetwork.VirtualStorageLibrary)	overwrite	

Type	Name	Description
bool ( <a href="https://learn.microsoft.com/dotnet/api/system.boolean">https://learn.microsoft.com/dotnet/api/system.boolean</a> )	<i>resolveLinks</i>	

## NodeExists(VirtualPath, bool)

### Declaration

```
(AkiraNetwork.VirtualStorageLibrary)
    NotifyNodeDelegate
public bool NodeExists(VirtualPath path, bool followLinks = false)
(AkiraNetwork.VirtualStorageLibrary)
    PatternMatch
```

### Parameters

Type	Name	Description
VirtualCycleDetector (AkiraNetwork.VirtualStorageLibrary)		
VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) bool ( <a href="https://learn.microsoft.com/dotnet/api/system.boolean">https://learn.microsoft.com/dotnet/api/system.boolean</a> )	<i>path</i> <i>followLinks</i>	

### Returns

Type	Description
VirtualException (AkiraNetwork.VirtualStorageLibrary)	

## RemoveLinkByLinkPath(VirtualPath)

### Declaration

```
(AkiraNetwork.VirtualStorageLibrary)
VirtualItemAdapter<T>
public void RemoveLinkByLinkPath(VirtualPath linkPath)
(AkiraNetwork.VirtualStorageLibrary)
    1.html
```

### Parameters

Type	Name	Description
VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	<i>linkPath</i>	

## RemoveLinkFromDictionary(VirtualPath, VirtualPath)

### Declaration

```
(AkiraNetwork.VirtualStorageLibrary)
VirtualNodeListConditions
public void RemoveLinkFromDictionary(VirtualPath targetPath, VirtualPath linkPath)
    VirtualnodeName
```

### Parameters

Type	Name	Description
VirtualNodeType VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	<i>targetPath</i>	

VirtualNodeTypeFilter

Type	Name	Description
VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	linkPath	

## RemoveNode(VirtualPath, bool, bool, bool)

### Declaration

```
(AkiraNetwork.VirtualStorageLibrary)
    NotifyNodeDelegate
public void RemoveNode(VirtualPath nodePath, bool recursive = false, bool followLinks = false,
(AkiraNetwork.VirtualStorageLibrary)
    bool resolveLinks = true)
PatternMatch
(AkiraNetwork.VirtualStorageLibrary)
```

### Parameters

Type	Name	Description
VirtualDirectory		
VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	nodePath	
VirtualDirectoryAdapter<T>		
bool (https://learn.microsoft.com/dotnet/api/system.boolean)	recursive	
bool (https://learn.microsoft.com/dotnet/api/system.boolean)	followLinks	
VirtualException		
bool (https://learn.microsoft.com/dotnet/api/system.boolean)	resolveLinks	

## ResolveLinkTarget(VirtualPath)

### Declaration

```
(AkiraNetwork.VirtualStorageLibrary)
    VirtualItem
VirtualItemAdapter<T>
(AkiraNetwork.VirtualStorageLibrary)
```

### Parameters

Type	Name	Description
VirtualItem<T>		
VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	path	

### Returns

```
(AkiraNetwork.VirtualStorageLibrary)
    VirtualNodeContext
```

Type	Description
VirtualNodeContext	
VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	

## SetLinkTargetNodeType(HashSet<VirtualPath>, VirtualNodeType)

### Declaration

```
(AkiraNetwork.VirtualStorageLibrary)
    VirtualNotFoundException
VirtualNodeType(HashSet<VirtualPath> linkPathSet, VirtualNodeType nodeTyp
e)
VirtualNodeType
(AkiraNetwork.VirtualStorageLibrary)
    VirtualNodeTypeFilter
```

## Parameters

Type	Name	Description
HashSet ( <a href="https://learn.microsoft.com/dotnet/api/system.collections.generic.hashset-1">https://learn.microsoft.com/dotnet/api/system.collections.generic.hashset-1</a> ) <VirtualPath ( <a href="#">AkiraNetwork.VirtualStorageLibrary.VirtualPath.html</a> )> ▼	<i>linkPathSet</i>	
VirtualNodeType ( <a href="#">AkiraNetwork.VirtualStorageLibrary.VirtualNodeType.html</a> )	<i>nodeType</i>	

# **SetNode(VirtualPath, VirtualNode)**

# Node.js PatternMatch

DeclarationNetwork.VirtualStorageLibrary

## VirtualCycleDetector

(AkiraNetwork VirtualStorage library)

### VirtualDirectory

(AkiraNetwork VirtualStorage library)

## (AKRAN) Parameters

## VirtualDirectoryAdapter<I

Type	Description	Name
(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)		
(AkiraNetwork.VirtualStorageLibrary.VirtualException)		<i>destinationPath</i>
(AkiraNetwork.VirtualStorageLibrary.VirtualNode.html)		<i>node</i>

### **SetNodeName(VirtualPath, VirtualnodeName, bool)**

(AkiraNetwork.VirtualStorageLibrary)

**VirtualItem**

(AkiraNetwork VirtualStorage) library

```
public void SetNodeName(VirtualPath nodePath, VirtualNodeName newName, bool resolveLinks = true)
```

(AkiraNetwork.VirtualStorageLibrary)

1.html

1.htm(l)

## Parameters

(AkiraNetwork.VirtualStorageLibrary  
1.html)

Type	Description	Name
(AkiraNetwork.VirtualStorageLibrary.html)		
VirtualPathName (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)		<i>nodePath</i>
(AkiraNetwork.VirtualStorageLibrary.html)		
VirtualNodeName (AkiraNetwork.VirtualStorageLibrary.VirtualnodeName.html)		<i>newName</i>
VirtualNodeContext		
bool (AkiraNetwork.VirtualStorageLibrary.html)		<i>resolveLinks</i>
bool (https://akiranetwork.com/api/system.boolean)		

**SymbolicLinkExists(VirtualPath)**  
(MAXTRANSNODENAME) throws an exception

## ▶ VirtualNodeName

(AkiraNetwork VirtualStorage) library

#### **VirtualNodeNetFoundException**

```
public bool SymbolicLinkExists(  
    (AkiraNetwork)VirtualStorageLibrary
```

## VirtuaNandaTrunc

## VirtualNode type

Type	Name	Description
VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	path	

## Returns

Type	Description
bool ( <a href="https://learn.microsoft.com/dotnet/api/system.boolean">https://learn.microsoft.com/dotnet/api/system.boolean</a> ) (AkiraNetwork.VirtualStorageLibrary.NotifyNodeDelegate)	

## TryGetDirectory(VirtualPath, bool)

(AkiraNetwork.VirtualStorageLibrary.VirtualDirectory? TryGetDirectory(VirtualPath path, bool followLinks = false))

### Parameters

Type	Name	Description
VirtualPath ( <a href="https://learn.microsoft.com/dotnet/api/system.boolean">https://learn.microsoft.com/dotnet/api/system.boolean</a> )	path	
bool ( <a href="https://learn.microsoft.com/dotnet/api/system.boolean">https://learn.microsoft.com/dotnet/api/system.boolean</a> )	followLinks	

### Returns

Type	Description
VirtualDirectory ( <a href="https://learn.microsoft.com/dotnet/api/system.boolean">https://learn.microsoft.com/dotnet/api/system.boolean</a> )	

## TryGetItem(VirtualPath, bool)

(AkiraNetwork.VirtualStorageLibrary.VirtualItem<T>? TryGetItem(VirtualPath path, bool followLinks = false))

### Parameters

Type	Name	Description
VirtualPath ( <a href="https://learn.microsoft.com/dotnet/api/system.boolean">https://learn.microsoft.com/dotnet/api/system.boolean</a> )	path	
bool ( <a href="https://learn.microsoft.com/dotnet/api/system.boolean">https://learn.microsoft.com/dotnet/api/system.boolean</a> )	followLinks	

### Returns

Type	Description
VirtualItem<T> ( <a href="https://learn.microsoft.com/dotnet/api/system.boolean">https://learn.microsoft.com/dotnet/api/system.boolean</a> )	

## TryGetNode(VirtualPath, bool)

### Declaration

```
public VirtualNode? TryGetNode(VirtualPath path, bool followLinks = false)
```



### Parameters

Type	Name	Description
(AkiraNetwork.VirtualStorageLibrary.NotifyNodeDelegate)(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	<i>path</i>	
bool (https://learn.microsoft.com/dotnet/api/system.boolean)(AkiraNetwork.VirtualStorageLibrary.VirtualCycleDetector)	<i>followLinks</i>	

### Returns

(AkiraNetwork.VirtualStorageLibrary.VirtualDirectory)

Type	Description
(AkiraNetwork.VirtualStorageLibrary.VirtualNode)(AkiraNetwork.VirtualStorageLibrary.VirtualNode.html)	

(AkiraNetwork.VirtualStorageLibrary.VirtualDirectoryAdapter<T>)(1.html)

## TryGetSymbolicLink(VirtualPath)

### Declaration

```
VirtualGroupCondition<T, TKey>
public VirtualSymbolicLink? TryGetSymbolicLink(VirtualPath path)
2.html)
```

VirtualID

### Parameters

Type	Name	Description
(AkiraNetwork.VirtualStorageLibrary.VirtualItem)(VirtualItemAdapter<T>)(VirtualPath)(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	<i>path</i>	

1.html)

### Returns

VirtualItem<T>

Type	Description
VirtualSymbolicLink (AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink.html)	

VirtualNode

## TryResolveLinkTarget(VirtualPath)

### Declaration

```
VirtualNodeExtensions
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeListConditions)
public VirtualPath? TryResolveLinkTarget(VirtualPath path)
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName)
```

### Parameters

Type	Name	Description
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeNotFoundException)(VirtualNodeType)(VirtualPath)(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	<i>path</i>	
VirtualNodeTypeFilter		

## Returns

Type	Description
VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	

## **UpdateAllTargetNodeTypesInDictionary()**

```
    (AkiraNetwork.VirtualStorageLibrary)
Declaration NotifyNodeDelegate
    (AkiraNetwork.VirtualStorageLibrary)
public void UpdateAllTargetNodeTypesInDictionary()
PatternMatch
```

**UpdateDirectory(VirtualPath, VirtualDirectory)**  
(AkiraNetwork.VirtualStorageLibrary)

```
Declarations
    AkiraNetwork.VirtualStorageLibrary
        VirtualDirectoryAdapter<T>
    (AkiraNetwork.VirtualStorageLibrary
        public void UpdateDirectory(VirtualPath directoryPath, VirtualDirectory newDirectory)
            1.html)
```

**Parameters** `network VirtualStorageLibrary`

Type	VirtualGroupCondition<T, TKey> (AkiraNetwork.VirtualStorageLibrary)	Name	Description
VirtualPath	(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) [.html]	directoryPath	
VirtualDirectory	(AkiraNetwork.VirtualStorageLibrary.VirtualDirectory.html) (AkiraNetwork.VirtualStorageLibrary)	newDirectory	

**UpdateItem(VirtualPath, VirtualItem<T>)**

```
(AkiraNetwork.VirtualStorageLibrary  
Declaration  
1.html)  
  
VirtualItem<T>  
public void UpdateItem(VirtualPath itemPath, VirtualItem<T> newItem)  
(AkiraNetwork.VirtualStorageLibrary  
1.html)
```

## VirtualNode Parameters

Type	Name	Description
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeContext)		
(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	<i>itemPath</i>	
VirtualExtensions		
VirtualItem (AkiraNetwork.VirtualStorageLibrary.VirtualItem-1.html) <T>	<i>newItem</i>	

## **UpdateLinkInDictionary(VirtualPath, VirtualPath)**

```
(AkiraNetwork.VirtualStorageLibrary)
Declaration NodeNotFoundException
(AkiraNetwork.VirtualStorageLibrary)
VirtualNodeType
public void UpdateLinkInDictionary(VirtualPath linkPath, VirtualPath newTargetPath)
(AkiraNetwork.VirtualStorageLibrary)
VirtualNodeTypeFilter
```

## Parameters

Type	Name	Description
VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	<i>linkPath</i>	
VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	<i>newTargetPath</i>	

▼

## UpdateLinksToTarget(VirtualPath, VirtualPath)

(AkiraNetwork.VirtualStorageLibrary.NotifyNodeDelegate)

Declaration: AkiraNetwork.VirtualStorageLibrary.

PatternMatch

public void UpdateLinksToTarget(VirtualPath oldTargetPath, VirtualPath newTargetPath)  
    VirtualCycleDetector

(AkiraNetwork.VirtualStorageLibrary.

Parameters:

VirtualDirectory

Type	Name	Description
VirtualDirectoryAdapter<T>		
VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	<i>oldTargetPath</i>	

VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

VirtualException

(AkiraNetwork.VirtualStorageLibrary.

VirtualGroupCondition<T, TKey>

## UpdateSymbolicLink(VirtualPath, VirtualSymbolicLink)

(AkiraNetwork.VirtualStorageLibrary.NotifyNodeDelegate)

Declaration:

VirtualID

(AkiraNetwork.VirtualStorageLibrary.  
public void UpdateSymbolicLink(VirtualPath linkPath, VirtualSymbolicLink newLink)  
VirtualItem

(AkiraNetwork.VirtualStorageLibrary.

VirtualItemAdapter<T>

Type	Name	Description
VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	<i>linkPath</i>	
VirtualSymbolicLink (AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink.html)	<i>newLink</i>	

VirtualNode

(AkiraNetwork.VirtualStorageLibrary.

VirtualNodeContext

(AkiraNetwork.VirtualStorageLibrary.

VirtualNodeExtensions

(AkiraNetwork.VirtualStorageLibrary.

public void UpdateTargetNodeTypesInDictionary(VirtualPath targetPath)

(AkiraNetwork.VirtualStorageLibrary.

VirtualNodeName

(AkiraNetwork.VirtualStorageLibrary.

VirtualNodeNotFoundException

Type	Name	Description
VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	<i>targetPath</i>	

(AkiraNetwork.VirtualStorageLibrary.

VirtualNodeTypeFilter

**WalkPathToTarget(VirtualPath, NotifyNodeDelegate?, ActionNodeDelegate?, bool, bool)**

## Declaration

```
public VirtualNodeContext WalkPathToTarget(VirtualPath targetPath, NotifyNodeDelegate? notifyNode = null, ActionNodeDelegate? actionNode = null, bool followLinks = true, bool exceptionEnabled = true)
    (AkiraNetwork.VirtualStorageLibrary.NotifyNodeDelegate? notifyNode, AkiraNetwork.VirtualStorageLibrary.ActionNodeDelegate? actionNode)
parameters
    (AkiraNetwork.VirtualStorageLibrary.NotifyNodeDelegate? notifyNode, AkiraNetwork.VirtualStorageLibrary.ActionNodeDelegate? actionNode)
```

Type	Description
(AkiraNetwork.VirtualStorageLibrary.VirtualPath(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html))	<i>targetPath</i>
(AkiraNetwork.VirtualStorageLibrary.VirtualCycleDetector)	
NotifyNodeDelegate(AkiraNetwork.VirtualStorageLibrary.NotifyNodeDelegate.html)	<i>notifyNode</i>
(AkiraNetwork.VirtualStorageLibrary.ActionNodeDelegate.html)	
(AkiraNetwork.VirtualStorageLibrary.VirtualDirectoryAdapter<T>(AkiraNetwork.VirtualStorageLibrary.ActionNodeDelegate.html))	<i>actionNode</i>
bool (https://learn.microsoft.com/dotnet/api/system.boolean)	<i>followLinks</i>
VirtualException	
bool (https://learn.microsoft.com/dotnet/api/system.boolean)	<i>exceptionEnabled</i>
(AkiraNetwork.VirtualStorageLibrary.VirtualPath(AkiraNetwork.VirtualPath.html))	

**VirtualGroupCondition<T, TKey>**

Type	Description
VirtualID	
VirtualNodeContext(AkisNetworkVirtualStorageLibrary.VirtualNodeContext.html)	

**WalkPathTree(VirtualPath, VirtualNodeTypeFilter, bool, bool, bool)**  
(AkiraNetwork.VirtualStorageLibrary)

## Declaration

```
VirtualItem<T>
(AkiraNetwork.VirtualStorageLibrary)
public IEnumerable<VirtualNodeContext> WalkPathTree(VirtualPath basePath, VirtualNodeTypeFilter filter = VirtualNodeTypeFilter.All, bool recursive = true, bool followLinks = true, bool resolveNodes = true)
(AkiraNetwork.VirtualStorageLibrary)
```

Type	Name	Description
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeExtensions)		
VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	basePath	
VirtualNodeListConditions		
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeTypeFilter)	filter	
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName)		
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeTypeFilter.html)		
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeNotNotFoundException)	recursive	
bool (https://learn.microsoft.com/dotnet/api/system.boolean)		
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeType)	followLinks	
bool (https://learn.microsoft.com/dotnet/api/system.boolean)		
VirtualNodeType		
bool (https://learn.microsoft.com/dotnet/api/system.boolean)	resolveLinks	
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeTypeFilter)		

## Returns

Type	Description
IEnumerable ( <a href="https://learn.microsoft.com/dotnet/api/system.collections.generic.ienumerable-1">https://learn.microsoft.com/dotnet/api/system.collections.generic.ienumerable-1</a> )<VirtualNodeContext (AkiraNetwork.VirtualStorageLibrary.VirtualNodeContext.html)>	

▼

## Extension Methods

(AkiraNetwork.VirtualStorageLibrary)

NotifyNodeDelegate

VirtualTextFormatter.GenerateLinkTableDebugText<T>(VirtualStorage<T>)

(AkiraNetwork.VirtualStorageLibrary.Utilities.VirtualTextFormatter.html#AkiraNetwork\_VirtualStorageLibrary\_Utilsilites\_VirtualTextFormatter\_GenerateLinkTableDebugText\_1\_AkiraNetwork\_VirtualStorageLibrary\_VirtualStorage\_0\_)

PatternMatch

VirtualTextFormatter.GenerateLinkTableDebugText<T>(T)

(AkiraNetwork.VirtualStorageLibrary.Utilities.VirtualTextFormatter.html#AkiraNetwork\_VirtualStorageLibrary\_Utilsilites\_VirtualTextFormatter\_GenerateLinkTableDebugText\_1\_AkiraNetwork\_VirtualStorageLibrary\_VirtualStorage\_0\_)

VirtualCycleDetector

VirtualTextFormatter.GenerateLinkTableDebugText<T>(T)

(AkiraNetwork.VirtualStorageLibrary.Utilities.VirtualTextFormatter.html#AkiraNetwork\_VirtualStorageLibrary\_Utilsilites\_VirtualTextFormatter\_GenerateLinkTableDebugText\_1\_AkiraNetwork\_VirtualStorageLibrary\_VirtualStorage\_0\_)

VirtualDirectory

VirtualTextFormatter.GenerateTreeDebugText<T>(VirtualStorage<T>, VirtualPath, bool, bool)

(AkiraNetwork.VirtualStorageLibrary.Utilities.VirtualTextFormatter.html#AkiraNetwork\_VirtualStorageLibrary\_Utilsilites\_VirtualTextFormatter\_GenerateTreeDebugText\_1\_AkiraNetwork\_VirtualStorageLibrary\_VirtualStorage\_0\_Akira

Network\_VirtualStorageLibrary\_VirtualPath\_System\_Boolean\_System\_Boolean\_

1.html)

VirtualException

(AkiraNetwork.VirtualStorageLibrary)

VirtualGroupCondition<T, TKey>

(AkiraNetwork.VirtualStorageLibrary)

2.html)

VirtualID

(AkiraNetwork.VirtualStorageLibrary)

VirtualItem

(AkiraNetwork.VirtualStorageLibrary)

VirtualItemAdapter<T>

(AkiraNetwork.VirtualStorageLibrary)

1.html)

VirtualItem<T>

(AkiraNetwork.VirtualStorageLibrary)

1.html)

VirtualNode

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeContext

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeExtensions

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeListConditions

(AkiraNetwork.VirtualStorageLibrary)

VirtualnodeName

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeNotFoundException

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeTypeFilter

# Class VirtualSymbolicLink

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

(AkiraNetwork.VirtualStorageLibrary)

The 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.

PatternMatch

(AkiraNetwork.VirtualStorageLibrary)

## Inheritance

VirtualCycleDetector

↳ object (<https://learn.microsoft.com/dotnet/api/system.object>)

↳ VirtualNode ([AkiraNetwork.VirtualStorageLibrary.VirtualNode.html](#))

(AkiraNetwork.VirtualStorageLibrary)

VirtualSymbolicLink

VirtualDirectoryAdapter<T>

## Implements

(AkiraNetwork.VirtualStorageLibrary)

IVirtualDeepCloneable ([AkiraNetwork.VirtualStorageLibrary.IVirtualDeepCloneable-1.html](#)) <VirtualNode

([AkiraNetwork.VirtualStorageLibrary.VirtualNode.html](#))>

## Inherited Members

VirtualGroupCondition<T, TKey>

VirtualNode.Name

(AkiraNetwork.VirtualStorageLibrary)

([AkiraNetwork.VirtualStorageLibrary.VirtualNode.html#AkiraNetwork\\_VirtualStorageLibrary\\_VirtualNode\\_Name](#))

2.html

VirtualNode.CreatedDate

VirtualID

([AkiraNetwork.VirtualStorageLibrary.VirtualNode.html#AkiraNetwork\\_VirtualStorageLibrary\\_VirtualNode\\_CreatedD](#)

ate)

VirtualItem

VirtualNode.UpdatedDate

(AkiraNetwork.VirtualStorageLibrary)

([AkiraNetwork.VirtualStorageLibrary.VirtualNode.html#AkiraNetwork\\_VirtualStorageLibrary\\_VirtualNode\\_Updated](#)

Date)

(AkiraNetwork.VirtualStorageLibrary)

VirtualNode.VID

1.html

([AkiraNetwork.VirtualStorageLibrary.VirtualNode.html#AkiraNetwork\\_VirtualStorageLibrary\\_VirtualNode\\_VID](#))

VirtualNode.IsReferencedInStorage

(AkiraNetwork.VirtualStorageLibrary)

([AkiraNetwork.VirtualStorageLibrary.VirtualNode.html#AkiraNetwork\\_VirtualStorageLibrary\\_VirtualNode\\_IsReferen](#)

cedinStorage)

VirtualNode

object.Equals(object) ([https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals\(system-object-object\)](https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals(system-object-object)))

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeContext

object.Equals(object, object) ([https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-object-equals\(system-object-system-object\)](https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-object-equals(system-object-system-object)))

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeExtensions

object.GetHashCode() (<https://learn.microsoft.com/dotnet/api/system.object.gethashcode>)

(AkiraNetwork.VirtualStorageLibrary)

object.GetType() (<https://learn.microsoft.com/dotnet/api/system.object.gettype>)

(VirtualNodeListConditions)

object.MemberwiseClone() (<https://learn.microsoft.com/dotnet/api/system.object.memberwiseclone>)

(AkiraNetwork.VirtualStorageLibrary)

object.ReferenceEquals(object, object) (<https://learn.microsoft.com/dotnet/api/system.object.referenceequals>)

VirtualNodeName

Name

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeFaultedException

(AkiraNetwork.VirtualStorageLibrary)

## Syntax

VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeTypeFilter

```
public class VirtualSymbolicLink : VirtualNode, IVirtualDeepCloneable<VirtualNode>
```

## Constructors

### **VirtualSymbolicLink()**

**Declaration**  
Notifies the NotifyNodeDelegate of the VirtualSymbolicLink (AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink.html) class. The node name is automatically generated using the VirtualNodeName.GenerateNodeName method.

**Declaration**  
(AkiraNetwork.VirtualStorageLibrary)

VirtualCycleDetector

(AkiraNetwork.VirtualStorageLibrary)  
public VirtualSymbolicLink()

VirtualDirectory

(AkiraNetwork.VirtualStorageLibrary)

VirtualDirectoryAdapter<T>

(AkiraNetwork.VirtualStorageLibrary)

### **VirtualSymbolicLink(VirtualNodeName)**

1.html)

**InitialValue**  
Initializes a new instance of the VirtualSymbolicLink (AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink.html) class with the specified VirtualNodeName.

**Declaration**  
VirtualGroupCondition<T, TKey>

(AkiraNetwork.VirtualStorageLibrary)

2.html)

public VirtualSymbolicLink(VirtualNodeName name)

(AkiraNetwork.VirtualStorageLibrary)

**Parameters**

(AkiraNetwork.VirtualStorageLibrary)

**Type** VirtualItemAdapter<T>

**Name**

**Description**

(AkiraNetwork.VirtualStorageLibrary)

name

1.html)

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)

The name of the symbolic link.

VirtualItem<T>

(AkiraNetwork.VirtualStorageLibrary)

1.html)

### **VirtualSymbolicLink(VirtualNodeName, VirtualPath?)**

(AkiraNetwork.VirtualStorageLibrary)

Initializes a new instance of the VirtualSymbolicLink (AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink.html) class with the specified name and target path.

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeExtensions

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeListConditions

public VirtualSymbolicLink(VirtualNodeName name, VirtualPath? targetPath)

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeName

(AkiraNetwork.VirtualStorageLibrary)

**Type** VirtualNodeNotFoundException

**Name**

**Description**

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeName

name

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)

The name of the symbolic link.

VirtualNodeTypeFilter

Type	Name	Description
VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	<i>targetPath</i>	The target path of the symbolic link.

## VirtualSymbolicLink(VirtualNodeName, VirtualPath?, DateTime, DateTime)

(AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink.html)  
 Initializes a new instance of the VirtualSymbolicLink (AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink.html) class with the specified name, target path, creation date, and update date.  
 (AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink.html)

### Declaration

```
(AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink)
public VirtualSymbolicLink(VirtualNodeName name, VirtualPath? targetPath, DateTime createdDate,
                           DateTime updatedDate)
  : base(updatedDate)
  {
    VirtualDirectory
    (AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink)
  }
```

### Parameters

VirtualDirectoryAdapter<T>

Type	Name	Description
(AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink.html)	<i>name</i>	The name of the symbolic link.
VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	<i>targetPath</i>	The target path of the symbolic link.
DateTime (https://learn.microsoft.com/dotnet/api/system.datetime)	<i>createdDate</i>	The creation date.
DateTime (https://learn.microsoft.com/dotnet/api/system.datetime)	<i>updatedDate</i>	The update date.

## Properties

(AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink)

1.html

VirtualItem

(AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink)

VirtualItemAdapter<T>

(AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink)

1.html

VirtualItem<T>

(AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink)

VirtualNodeContext

Gets the type of the node. This class always returns SymbolicLink.

(AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink)

### Declaration

(AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink)

VirtualNodeListCondition<T> NodeType { get; }

(AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink)

VirtualNodeName

(AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink)

VirtualNodeNotFoundException

(AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink)

VirtualNodeType (AkiraNetwork.VirtualStorageLibrary.VirtualNodeType.html)

VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink)

### Overrides

VirtualNodeTypeFilter

### Description

Type	Description
VirtualNodeType	

VirtualNode.NodeType  
(AkiraNetwork.VirtualStorageLibrary.VirtualNode.html#AkiraNetwork\_VirtualStorageLibrary\_VirtualNode\_NodeType)

## TargetNodeType

Gets or sets the type of the target node of the symbolic link.

(AkiraNetwork.VirtualStorageLibrary.

**Declaration**

NotifyNodeDelegate

(AkiraNetwork.VirtualStorageLibrary.

public VirtualNodeType TargetNodeType { get; set; }

(AkiraNetwork.VirtualStorageLibrary.

**Property Value**

(AkiraNetwork.VirtualStorageLibrary.

**Type**

VirtualDirectory

(AkiraNetwork.VirtualStorageLibrary.

VirtualDirectoryAdapter<T>

(AkiraNetwork.VirtualStorageLibrary.

1.html)

**TargetException**

(AkiraNetwork.VirtualStorageLibrary.

Gets or sets the target path of the symbolic link.

VirtualGroupCondition<T, TKey>

(AkiraNetwork.VirtualStorageLibrary.

2.html)

VirtualPath

VirtualPath? TargetPath { get; set; }

(AkiraNetwork.VirtualStorageLibrary.

VirtualItem

(AkiraNetwork.VirtualStorageLibrary.

**Type**

VirtualItemAdapter<T>

(AkiraNetwork.VirtualStorageLibrary.

VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

.html)

VirtualItem<T>

(AkiraNetwork.VirtualStorageLibrary.

1.html)

## Methods

(AkiraNetwork.VirtualStorageLibrary.

VirtualNodeContext

**DeepClone(bool)**

(AkiraNetwork.VirtualStorageLibrary.

VirtualNodeExtensions

Creates a deep clone of the symbolic link.

(AkiraNetwork.VirtualStorageLibrary.

VirtualNodeListConditions

(AkiraNetwork.VirtualStorageLibrary.

VirtualNodeName

public override VirtualNode DeepClone(bool recursive = false)

(AkiraNetwork.VirtualStorageLibrary.

VirtualNodeNotFoundException

(AkiraNetwork.VirtualStorageLibrary.

VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary.

VirtualNodeTypeFilter

(AkiraNetwork.VirtualStorageLibrary.

VirtualStorageLibrary.

**Description**

**Description**

Type	Name	Description
bool ( <a href="https://learn.microsoft.com/dotnet/api/system.boolean">https://learn.microsoft.com/dotnet/api/system.boolean</a> )	recursive	If true, all links are recursively cloned.

## Returns

Type	Description
VirtualNode ( <a href="#">AkiraNetwork.VirtualStorageLibrary.VirtualNode.html</a> ) ( <a href="#">AkiraNetwork.VirtualStorageLibrary.VirtualNode</a> )	A deep clone of the symbolic link.

## Overrides

[NotifyNodeDelegate](#)  
[VirtualNode.DeepClone\(bool\)](#)  
[PatternMatch](#)  
[VirtualNode.DeepClone\(bool\)](#)  
[\(AkiraNetwork.VirtualStorageLibrary.VirtualNode.html#AkiraNetwork\\_VirtualStorageLibrary\\_VirtualNode\\_DeepClone\\_System\\_Boolean\)](#)  
[VirtualCycleDetector](#)

## Tostring()

[VirtualDirectory](#)  
[VirtualDirectoryAdapter<T>](#)  
 Returns a string representation of the symbolic link.  
[\(AkiraNetwork.VirtualStorageLibrary.VirtualDirectory\)](#)

## Declaration

[VirtualException](#)  
[\(AkiraNetwork.VirtualStorageLibrary.VirtualException\)](#)

[VirtualGroupCondition<T, TKey>](#)  
[\(AkiraNetwork.VirtualStorageLibrary.VirtualGroupCondition<T, TKey>\)](#)

[Returns](#)  
[2.html](#)

Type	Description
<a href="#">virtualID</a> ( <a href="#">AkiraNetwork.VirtualStorageLibrary.VirtualID</a> )	A string representation of the symbolic link.

## Overrides

[VirtualItemAdapter<T>](#)  
[object.ToString\(\)](#) (<https://learn.microsoft.com/dotnet/api/system.object.tostring>)  
[1.html](#)

## Update(VirtualNode)

[\(AkiraNetwork.VirtualStorageLibrary.VirtualNode\)](#)  
[1.html](#)

Updates the current symbolic link with the data from the specified node.

[VirtualNode](#)

[\(AkiraNetwork.VirtualStorageLibrary.VirtualNode\)](#)

[Parameters](#)  
[VirtualNodeListConditions](#)

Type	Name	Description
<a href="#">AkiraNetwork.VirtualStorageLibrary.VirtualNode</a> ( <a href="#">AkiraNetwork.VirtualStorageLibrary.VirtualNode</a> )	<a href="#">node</a>	The node to use for the update.

## Overrides

[\(AkiraNetwork.VirtualStorageLibrary.VirtualNode\)](#)  
[VirtualNode.Update\(VirtualNode\)](#)

[\(AkiraNetwork.VirtualStorageLibrary.VirtualNode.html#AkiraNetwork\\_VirtualStorageLibrary\\_VirtualNode\\_Update\\_AkiraNetwork\\_VirtualStorageLibrary\\_VirtualNode\\_\)](#)

## Exceptions

Type	Condition
ArgumentException ( <a href="https://learn.microsoft.com/dotnet/api/system.argumentexception">https://learn.microsoft.com/dotnet/api/system.argumentexception</a> )	Thrown if the specified node is not a VirtualSymbolicLink.

## Operators

### implicit operator VirtualSymbolicLink(VirtualPath?)

Performs an implicit conversion from VirtualPath ([AkiraNetwork.VirtualStorageLibrary.VirtualPath.html](#)) to VirtualSymbolicLink ([AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink.html](#)). The node name is automatically generated using the `VirtualNodeName.GenerateNodeName` method.

([AkiraNetwork.VirtualStorageLibrary](#))

#### Declaration

```
VirtualDirectoryAdapter<T>
{
    public static implicit operator VirtualSymbolicLink(VirtualPath? targetPath)
```

VirtualException

([AkiraNetwork.VirtualStorageLibrary](#))

#### Parameters

VirtualGroupCondition<T, TKey>

([AkiraNetwork.VirtualStorageLibrary](#))

#### Type

VirtualPath ([AkiraNetwork.VirtualStorageLibrary.VirtualPath.html](#))

VirtualID

([AkiraNetwork.VirtualStorageLibrary](#))

#### Returns

VirtualItem

([AkiraNetwork.VirtualStorageLibrary](#))

#### Type

VirtualItemAdapter<T>

VirtualSymbolicLink ([AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink.html](#))

([AkiraNetwork.VirtualStorageLibrary](#))

1.html)

VirtualItem<T>

([AkiraNetwork.VirtualStorageLibrary](#))

1.html)

### implicit operator VirtualSymbolicLink((VirtualNodeName nodeName, VirtualPath? targetPath))

VirtualNode

([AkiraNetwork.VirtualStorageLibrary](#))

Perform an implicit conversion from a tuple to VirtualSymbolicLink

([AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink.html](#)).

([AkiraNetwork.VirtualStorageLibrary](#))

#### Declaration

VirtualNodeExtensions

([AkiraNetwork.VirtualStorageLibrary](#))

public static implicit operator VirtualSymbolicLink((VirtualNodeName nodeName, VirtualPath?

targetPath) tuple)

([AkiraNetwork.VirtualStorageLibrary](#))

VirtualNodeName

([AkiraNetwork.VirtualStorageLibrary](#))

VirtualNotFoundException

([AkiraNetwork.VirtualStorageLibrary](#))

VirtualNodeType

([AkiraNetwork.VirtualStorageLibrary](#))

VirtualNodeTypeFilter

Type	Name	Description
(VirtualNodeName (AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html) nodeName ( <a href="https://learn.microsoft.com/dotnet/api/system.valuetuple-akiranetwork.virtualstoragelibrary.virtualnodename,akiranetwork.virtualstoragelibrary.virtualpath-.nodename">https://learn.microsoft.com/dotnet/api/system.valuetuple-akiranetwork.virtualstoragelibrary.virtualnodename,akiranetwork.virtualstoragelibrary.virtualpath-.nodename</a> ) , VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) targetPath ( <a href="https://learn.microsoft.com/dotnet/api/system.valuetuple-akiranetwork.virtualstoragelibrary.virtualnodename,akiranetwork.virtualstoragelibrary.virtualpath-targetpath">https://learn.microsoft.com/dotnet/api/system.valuetuple-akiranetwork.virtualstoragelibrary.virtualnodename,akiranetwork.virtualstoragelibrary.virtualpath-targetpath</a> ) NotifyNodeDelegate ) (AkiraNetwork.VirtualStorageLibrary.PatternMatch	tuple	The tuple to convert.

**Returns** VirtualCycleDetector

Type	AkiraNetwork.VirtualStorageLibrary	Description
VirtualDirectory	(AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink.html) (AkiraNetwork.VirtualStorageLibrary)	

# Implements 1.html

`IVirtualDeepCloneable<T>` ([AkiraNetwork.VirtualStorageLibrary.IVirtualDeepCloneable-1.html](#))  
([AkiraNetwork.VirtualStorageLibrary](#))

## VirtualGroupCondition<T, TKey>

# Extension Methods

# Extension Method (Extending Network Virtual Storage)

```
VirtualNodeExtensions.ResolveNodeType(VirtualNode)
```

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeExtensions.html#AkiraNetwork\_VirtualStorageLibrary\_VirtualNode  
Extensions\_ResolveNodeType\_AkiraNetwork\_VirtualStorageLibrary\_VirtualNode\_)

`VirtualTextFormatter.GenerateSingleTableDebugText< T >(T)`  
`(AkiraNetwork.VirtualStorageLibrary.AkiraVirtualTextFormatter)`

(AkiraNetwork.VirtualStorageLibrary.Utilities.VirtualTextFormatter.html#AkiraNetwork\_VirtualStorageLibrary\_Utils  
s\_VirtualTextFormatter\_Adapter\_<T>  
    (AkiraNetwork.VirtualStorageLibrary.Utilities.VirtualTextFormatter.html#AkiraNetwork\_VirtualStorageLibrary\_Utils  
s\_VirtualTextFormatter\_GenerateSingleTableDebugText\_1\_\_0\_)

1.html)  
VirtualItem<T>

`VirtualItem<T>`  
(`AkiraNetwork.VirtualStorageLibrary`)

1.html)

## VirtualNode

## VirtualNode (AkiraNetwork.VirtualStorageLibrary)

## VirtualNodeContext

## (AkiraNetwork.VirtualStorageLibrary VirtualNodeExtensions

## VirtualNodeExtensions (AlisraNetworks VirtualSt

## (AkiraNetwork.virtualStorageLibrary) VirtualNodeListConditions

(AkiraNetwork.VirtualStorage)

**VirtualNodeName**  
(AliyunNetworkVirtualStorageLibrary)

(AkiraNetwork.Virt)

## VirtualNodeNotFoundException (AkiraNetwork.VirtualStorageLibrary)

## VirtualNodeType

**VirtualStorageType**  
(AkiraNetwork VirtualStorageLibrary)

#### **VirtualNodeTypeFilter**

# Class VirtualSymbolicLinkAdapter<T>

## Inheritance

↳ [Object](#) (<https://learn.microsoft.com/dotnet/api/system.object>)  
↳ [VirtualSymbolicLinkAdapter<T>](#)  
    (AkiraNetwork.VirtualStorageLibrary)

## Inherited Members

NotifyNodeDelegate

object.Equals(object) ([https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals\(system-object\)](https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals(system-object)))  
PatternMatch  
(AkiraNetwork.VirtualStorageLibrary)  
object.Equals(object) ([https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals\(system-object-system-object\)](https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals(system-object-system-object)))  
VirtualCycleDetector  
(AkiraNetwork.VirtualStorageLibrary)  
object.GetHashCode() (<https://learn.microsoft.com/dotnet/api/system.object.gethashcode>)  
object.GetType() (<https://learn.microsoft.com/dotnet/api/system.object.gettype>)  
object.MemberwiseClone()  
(AkiraNetwork.VirtualStorageLibrary)  
object.ReferenceEquals(object1, object2) (<https://learn.microsoft.com/dotnet/api/system.object.referenceequals>)  
object.ToString() (<https://learn.microsoft.com/dotnet/api/system.object.tostring>)  
1.html)

**Namespace:** AkiraNetwork (AkiraNetwork.html).VirtualStorageLibrary (AkiraNetwork.VirtualStorageLibrary.html)  
VirtualException

**Assembly:** VirtualStorageLibrary.dll

**Syntax:** VirtualGroupCondition<T, TKey>

(AkiraNetwork.VirtualStorageLibrary)  
public class VirtualSymbolicLinkAdapter<T>  
    VirtualID  
    (AkiraNetwork.VirtualStorageLibrary)

## Type Parameters

VirtualItem

Name	Description
T	(AkiraNetwork.VirtualStorageLibrary) 1.html)

VirtualItem<T>

## Constructors

(AkiraNetwork.VirtualStorageLibrary)

1.html)

VirtualNode

## VirtualSymbolicLinkAdapter(VirtualStorage<T>)

### Declaration

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeContext

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeExtensions

public VirtualSymbolicLinkAdapter(VirtualStorage<T> storage)  
(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeListConditions

(AkiraNetwork.VirtualStorageLibrary)

### Parameters

VirtualStorage<T> storage

Type	Name	Description
VirtualNodeName (AkiraNetwork.VirtualStorageLibrary)		
VirtualStorage<T> storage (AkiraNetwork.VirtualStorageLibrary)	storage	

VirtualStorage<T> storage

VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeTypeFilter

# Properties

## this[VirtualPath, bool]

### Declaration

```
public VirtualSymbolicLink this[VirtualPath path, bool followLinks = true] { get; set; }
```

(AkiraNetwork.VirtualStorageLibrary)

NotifyNodeDelegate

### Parameters

(AkiraNetwork.VirtualStorageLibrary)

#### Type

PatternMatch

(AkiraNetwork.VirtualStorageLibrary)

#### VirtualPath

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

#### VirtualCycleDetector

(AkiraNetwork.VirtualStorageLibrary)

#### bool

(<https://learn.microsoft.com/dotnet/api/system.boolean>)

VirtualDirectory

(AkiraNetwork.VirtualStorageLibrary)

#### Property Value

VirtualDirectoryAdapter<T>

#### Type

AkiraNetwork.VirtualStorageLibrary

1.html)

#### VirtualSymbolicLink

(AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink.html)

VirtualException

(AkiraNetwork.VirtualStorageLibrary)

#### VirtualGroupCondition<T, TKey>

(AkiraNetwork.VirtualStorageLibrary)

#### VirtualTextFormatter

GenerateSingleTableDebugText<T>(T)

VirtualID

#### (AkiraNetwork.VirtualStorageLibrary.Uses.VirtualTextFormatter.html#AkiraNetwork\_VirtualStorageLibrary\_Utilities\_VirtualTextFormatter\_GenerateSingleTableDebugText\_1\_\_0\_)

VirtualItem

(AkiraNetwork.VirtualStorageLibrary)

VirtualItemAdapter<T>

(AkiraNetwork.VirtualStorageLibrary)

1.html)

VirtualItem<T>

(AkiraNetwork.VirtualStorageLibrary)

1.html)

VirtualNode

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeContext

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeExtensions

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeListConditions

(AkiraNetwork.VirtualStorageLibrary)

VirtualnodeName

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeNotFoundException

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeTypeFilter

## Extension Methods

# Namespace AkiraNetwork.VirtualStorage Library.Utilities

▼ Filter by title

## Classes

+ **AkiraNetwork.VirtualStorage**

**Library**

**VirtualTextFormatter**

            (AkiraNetwork.VirtualStorage)

            ([AkiraNetwork.VirtualStorageLibrary.Utilities.VirtualTextFormatter.html](#))

- **AkiraNetwork.VirtualStorage**  
    **Library.Utilities**  
        (AkiraNetwork.VirtualStorage)

    VirtualTextFormatter

        (AkiraNetwork.VirtualStorageLibrary)

+ **AkiraNetwork.VirtualStorage**

**Library.WildcardMatchers**

        (AkiraNetwork.VirtualStorage)

# Class VirtualTextFormatter

Utility class for generating text-based representations of the virtual storage's tree structure. Provides static methods for generating various debug texts.

## Inheritance

### AkiraNetwork.VirtualStorage

↳ [Object](#) (<https://learn.microsoft.com/dotnet/api/system.object>)

### (AkiraNetwork.VirtualStorage)

## Inherited Members

### AkiraNetwork.VirtualStorage

object.Equals(object, object) ([https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals\(system-object-object\)](https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals(system-object-object)))

### (AkiraNetwork.VirtualStorage)

object.Equals(object, object) ([https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals\(system-object-object-system-object-object\)](https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals(system-object-object-system-object-object)))

object.GetHashCode() (<https://learn.microsoft.com/dotnet/api/system.object.gethashcode>)

### (AkiraNetwork.VirtualStorage)

object.MemberwiseClone() (<https://learn.microsoft.com/dotnet/api/system.object.memberwiseclone>)

object.ReferenceEquals(object, object) (<https://learn.microsoft.com/dotnet/api/system.object.referenceequals>)

object.ToString() (<https://learn.microsoft.com/dotnet/api/system.object.tostring>)

**Namespace:** AkiraNetwork ([AkiraNetwork.html](#)).VirtualStorageLibrary ([AkiraNetwork.VirtualStorageLibrary.html](#)).Utilities ([AkiraNetwork.VirtualStorageLibrary.Utilities.html](#))

**Assembly:** VirtualStorageLibrary.dll

## Syntax

```
public static class VirtualTextFormatter
```

## Remarks

This class distinguishes between full-width and half-width characters based on specific character code ranges. The following ranges are considered full-width characters:

- 0x1100 to 0x115F: Korean Hangul Jamo characters
- 0xE80 to 0xA4CF (excluding 0x303F): CJK Unified Ideographs and compatibility characters
- 0xAC00 to 0xD7A3: Korean Hangul syllables
- 0xF900 to 0xFAFF: CJK compatibility ideographs
- 0xFE10 to 0xFE19: Vertical forms punctuation
- 0xFE30 to 0xFE6F: CJK compatibility forms
- 0xFF00 to 0xFF60: Full-width ASCII and symbols
- 0FFE0 to 0FFE6: Full-width special symbols

The distinction between full-width and half-width characters is important for ensuring proper alignment and formatting in text-based representations, especially when working with mixed-language data or presenting data in a tabular format. This classification helps in correctly calculating string widths and aligning elements in debug text outputs, such as file paths or tabular data, where accurate spacing is crucial.

# Methods

## GenerateLinkTableDebugText<T>(VirtualStorage<T>)

Generates and returns a table representation of symbolic link information.



### Declaration

#### + AkiraNetwork.VirtualStorage

```
public static string GenerateLinkTableDebugText<T>(this VirtualStorage<T> vs)
```

#### Library

#### (AkiraNetwork.VirtualStorage)

##### Parameters

#### - AkiraNetwork.VirtualStorage

#### Type Library.Utilities

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

#### Name

#### Description

vs The instance of the virtual storage.

### Returns

#### AkiraNetwork.VirtualStorage

#### Library.WildcardMatchers

#### Type

#### (AkiraNetwork.VirtualStorage)

string (<https://learn.microsoft.com/dotnet/api/system.string>)

#### Description

The generated debug text of the link table.

### Type Parameters

Name	Description
T	The type of data in the storage.

## GenerateSingleTableDebugText<T>(T)

Generates and returns a table representation of a single object's properties.

### Declaration

```
public static string GenerateSingleTableDebugText<T>(this T singleObject)
```

### Parameters

Type	Name	Description
T	singleObject	The object to be represented.

### Returns

Type	Description
string ( <a href="https://learn.microsoft.com/dotnet/api/system.string">https://learn.microsoft.com/dotnet/api/system.string</a> )	The generated debug text of the table.

### Type Parameters

Name	Description
<i>T</i>	The type of the object.

## GenerateTableDebugText<T>(IEnumerable<T>)

Generates and returns a table representation of the collection's contents.

### + AkiraNetwork.VirtualStorage

Declaration  
Library

#### (AkiraNetwork.VirtualStorage)

```
public static string GenerateTableDebugText<T>(this IEnumerable<T> enumerableObject)
```

### - AkiraNetwork.VirtualStorage

Library Utilities

Parameters

#### (AkiraNetwork.VirtualStorage)

##### Type

VirtualTextFormatter

##### IEnumerable

(AkiraNetwork.VirtualStorageLibrary

(<https://learn.microsoft.com/dotnet/api/system.collections.generic.enumerable>)

### + AkiraNetwork.VirtualStorage

#### <Library.WildcardMatchers

#### (AkiraNetwork.VirtualStorage)

Returns

Type	Description
string ( <a href="https://learn.microsoft.com/dotnet/api/system.string">https://learn.microsoft.com/dotnet/api/system.string</a> )	The generated debug text of the table.

Type Parameters

Name	Description
<i>T</i>	The type of objects in the collection.

## GenerateTreeDebugText<T>(VirtualStorage<T>, VirtualPath, bool, bool)

Generates and returns a text-based representation of the virtual storage tree structure.

Declaration

```
public static string GenerateTreeDebugText<T>(this VirtualStorage<T> vs, VirtualPath basePath, bool recursive = true, bool followLinks = false)
```

Parameters

Type	Name	Description
VirtualStorage<T>	vs	The instance of the virtual storage.

Type		Name	Description
VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)		<i>basePath</i>	The base path from which to start.
bool (https://learn.microsoft.com/dotnet/api/system.boolean) ▼		<i>recursive</i>	Whether to list subdirectories recursively.
bool (https://learn.microsoft.com/dotnet/api/system.boolean)		<i>followLinks</i>	Whether to follow symbolic links.

## + AkiraNetwork.VirtualStorage

### Library

Type	Description
-string (https://learn.microsoft.com/dotnet/api/system.string)	The generated debug text of the tree structure.

### Library.Utilities

#### Type AkiraNetwork.VirtualStorage

Name	Description
<i>T</i> (AkiraNetwork.VirtualStorageLibrary)	The type of data in the storage.

## + AkiraNetwork.VirtualStorage

### Library.WildcardMatchers

The generated text represents the tree structure starting from the specified base path. Each node is represented by the result of the `ToString` method overridden in the specific derived class of the node.

When `recursive` is true, subdirectories are listed recursively. If false, subdirectories are not included in the output.

When `followLinks` is true, symbolic links are replaced with their target paths in the output. If false, the symbolic links themselves are displayed.

The output format includes indents and special characters to denote directory hierarchies and symbolic links.  
Example:

```

/
├dir1/
│├subdir1/
││└item3
│└item2
└link-to-dir -> /dir1
└item1
└link-to-item -> /item1

```

# Namespace AkiraNetwork.VirtualStorage Library.WildcardMatchers

▼ Filter by title

## Classes

+ **AkiraNetwork.VirtualStorage**

**Library**

**WildcardMatcher**

        (AkiraNetwork.VirtualStorageLibrary.WildcardMatchers.DefaultWildcardMatcher.html)

**PowerShellWildcardMatcher**

        (AkiraNetwork.VirtualStorageLibrary.WildcardMatchers.PowerShellWildcardMatcher.html)

**WildcardMatchers**

        (AkiraNetwork.VirtualStorageLibrary.WildcardMatchers.DefaultWildcardMatcher.html)

            DefaultWildcardMatcher

            (AkiraNetwork.VirtualStorageLibrary.WildcardMatchers.DefaultWildcardMatcher.html)

            PowerShellWildcardMatcher

            (AkiraNetwork.VirtualStorageLibrary.WildcardMatchers.PowerShellWildcardMatcher.html)

# Class DefaultWildcardMatcher

## Inheritance

↳ [Object](#) (https://learn.microsoft.com/dotnet/api/system.object)  
↳ [DefaultWildcardMatcher](#)

## + [AkiraNetwork.VirtualStorage](#)

Implements

IVirtualWildcardMatcher ([AkiraNetwork.VirtualStorageLibrary.IVirtualWildcardMatcher.html](#))

## ([AkiraNetwork.VirtualStorage](#))

Inherited Members

## + [AkiraNetwork.VirtualStorage](#)

object.Equals(object) (https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals(system-object))

## ([AkiraNetwork.VirtualStorage](#))

object.Equals(object, object) (https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals(system-object-system-object))

## - [AkiraNetwork.VirtualStorage](#)

object.GetHashCode() (https://learn.microsoft.com/dotnet/api/system.object.gethashcode)

object.GetType() (https://learn.microsoft.com/dotnet/api/system.object.gettype)

object.MemberwiseClone() (https://learn.microsoft.com/dotnet/api/system.object.memberwiseclone)

object.ReferenceEquals(object, object) (https://learn.microsoft.com/dotnet/api/system.object.referenceequals)

object.ToString() (https://learn.microsoft.com/dotnet/api/system.object.tostring)

Namespace: [AkiraNetwork](#).[VirtualStorage](#).[Library](#).[WildcardMatchers](#) ([AkiraNetwork.VirtualStorageLibrary.WildcardMatchers.html](#))

Assembly: [VirtualStorageLibrary.dll](#)

## Syntax

```
public class DefaultWildcardMatcher : IVirtualWildcardMatcher
```

# Properties

## Count

Gets the number of wildcards.

## Declaration

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

## Property Value

Type	Description
int (https://learn.microsoft.com/dotnet/api/system.int32)	An integer representing the number of wildcards available.

## Patterns

Gets a collection of available patterns.

### Declaration

```
public IEnumerable<string> Patterns { get; }
```

### Property Value

Type	Description
<b>(AkiraNetwork.VirtualStorage)</b> IEnumerable (https://learn.microsoft.com/dotnet/api/system.collections.generic.ienumerable-1) <b>(AkiraNetwork.VirtualStorage)</b> Library.Utilities <string (https://learn.microsoft.com/dotnet/api/system.string)> <b>(AkiraNetwork.VirtualStorage)</b>	An enumerable collection of strings representing the available patterns.

### - AkiraNetwork.VirtualStorage

#### Library.WildcardMatchers

##### WildcardDictionary

###### (AkiraNetwork.VirtualStorage)

Gets a dictionary of wildcards and their corresponding regex patterns.

DefaultWildcardMatcher

#### Declaration

PowerShellWildcardMatcher

(AkiraNetwork.VirtualStorage)

```
public IReadOnlyDictionary<string, string> WildcardDictionary { get; }
```

### Property Value

Type	Description
ReadOnlyDictionary (https://learn.microsoft.com/dotnet/api/system.collections.objectmodel,readonlydictionary-2) <string (https://learn.microsoft.com/dotnet/api/system.string), string (https://learn.microsoft.com/dotnet/api/system.string)>	A read-only dictionary containing wildcards as keys and their corresponding regex patterns as values.

## Wildcards

Gets a collection of available wildcards.

### Declaration

```
public IEnumerable<string> Wildcards { get; }
```

### Property Value

Type	Description
IEnumerable ( <a href="https://learn.microsoft.com/dotnet/api/system.collections.generic.ienumerable-1">https://learn.microsoft.com/dotnet/api/system.collections.generic.ienumerable-1</a> ) <string ( <a href="https://learn.microsoft.com/dotnet/api/system.string">https://learn.microsoft.com/dotnet/api/system.string</a> )>	An enumerable collection of strings representing the available wildcards.

## + AkiraNetwork.VirtualStorage

### Methods

**(AkiraNetwork.VirtualStorage)**

**PatternMatcher(string, string)**

#### Library.Utilities

Determines whether the specified node name matches the specified pattern.

**(AkiraNetwork.VirtualStorage)**

Declaration

**- AkiraNetwork.VirtualStorage**

#### Library.WildcardMatchers

```
public bool PatternMatcher(string nodeName, string pattern)
```

**(AkiraNetwork.VirtualStorage)**

DefaultWildcardMatcher

**Parameters**

(AkiraNetwork.VirtualStorageLibrary.

Type	Name	Description
PowerShellWildcardMatcher (AkiraNetwork.VirtualStorageLibrary. string ( <a href="https://learn.microsoft.com/dotnet/api/system.string">https://learn.microsoft.com/dotnet/api/system.string</a> )	nodeName	The node name to check.
string ( <a href="https://learn.microsoft.com/dotnet/api/system.string">https://learn.microsoft.com/dotnet/api/system.string</a> )	pattern	The pattern to compare against.

Returns

Type	Description
bool ( <a href="https://learn.microsoft.com/dotnet/api/system.boolean">https://learn.microsoft.com/dotnet/api/system.boolean</a> )	True if the node name matches the pattern; otherwise, false.

## Implements

IVirtualWildcardMatcher (AkiraNetwork.VirtualStorageLibrary.IVirtualWildcardMatcher.html)

## Extension Methods

VirtualTextFormatter.GenerateSingleTableDebugText<T>(T)

(AkiraNetwork.VirtualStorageLibrary.Utilities.VirtualTextFormatter.html#AkiraNetwork\_VirtualStorageLibrary\_Utils  
s\_VirtualTextFormatter\_GenerateSingleTableDebugText\_1\_\_0\_)

# Class PowerShellWildcardMatcher

## Inheritance

↳ [Object](#) (<https://learn.microsoft.com/dotnet/api/system.object>)  
↳ [PowerShellWildcardMatcher](#)

## + [AkiraNetwork.VirtualStorage](#)

Implements

[IVirtualWildcardMatcher](#) ([AkiraNetwork.VirtualStorageLibrary.IVirtualWildcardMatcher.html](https://learn.microsoft.com/dotnet/api/AkiraNetwork.VirtualStorageLibrary.IVirtualWildcardMatcher.html))

## - [\(AkiraNetwork.VirtualStorage\)](#)

Inherited Members

## + [AkiraNetwork.VirtualStorage](#)

[object.Equals\(object\)](#) ([<https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals\(system-object\)>](https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals(system-object)))

## - [\(AkiraNetwork.VirtualStorage\)](#)

[object.GetHashCode\(\)](#) ([<https://learn.microsoft.com/dotnet/api/system.object.gethashcode>](https://learn.microsoft.com/dotnet/api/system.object.gethashcode))

[object.GetType\(\)](#) ([<https://learn.microsoft.com/dotnet/api/system.object.gettype>](https://learn.microsoft.com/dotnet/api/system.object.gettype))

[object.MemberwiseClone\(\)](#) ([<https://learn.microsoft.com/dotnet/api/system.object.memberwiseclone>](https://learn.microsoft.com/dotnet/api/system.object.memberwiseclone))

[object.ReferenceEquals\(object, object\)](#) ([<https://learn.microsoft.com/dotnet/api/system.object.referenceequals>](https://learn.microsoft.com/dotnet/api/system.object.referenceequals))

[object.ToString\(\)](#) ([<https://learn.microsoft.com/dotnet/api/system.object.tostring>](https://learn.microsoft.com/dotnet/api/system.object.tostring))

[PowerShellWildcardMatcher](#)  
Namespace: [AkiraNetwork](#) ([AkiraNetwork.WildcardMatchers.html](https://learn.microsoft.com/dotnet/api/AkiraNetwork.WildcardMatchers)).[VirtualStorageLibrary](#) ([AkiraNetwork.VirtualStorageLibrary.html](https://learn.microsoft.com/dotnet/api/AkiraNetwork.VirtualStorageLibrary)).[WildcardMatchers](#) ([AkiraNetwork.VirtualStorageLibrary.WildcardMatchers.html](https://learn.microsoft.com/dotnet/api/AkiraNetwork.VirtualStorageLibrary.WildcardMatchers))

Assembly: [VirtualStorageLibrary.dll](#)

## Syntax

```
public class PowerShellWildcardMatcher : IVirtualWildcardMatcher
```

# Properties

## Count

Gets the number of wildcards.

## Declaration

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

## Property Value

Type	Description
int ( <a href="https://learn.microsoft.com/dotnet/api/system.int32">https://learn.microsoft.com/dotnet/api/system.int32</a> )	An integer representing the number of wildcards available.

## Patterns

Gets a collection of available patterns.

### Declaration

```
public IEnumerable<string> Patterns { get; }
```

### Property Value

Type	Description
<b>(AkiraNetwork.VirtualStorage)</b> IEnumerable (https://learn.microsoft.com/dotnet/api/system.collections.generic.ienumerable-1) <b>(AkiraNetwork.VirtualStorage)</b> Library.Utilities <string (https://learn.microsoft.com/dotnet/api/system.string)> <b>(AkiraNetwork.VirtualStorage)</b>	An enumerable collection of strings representing the available patterns.

### - AkiraNetwork.VirtualStorage

#### Library.WildcardMatchers

##### WildcardDictionary

###### (AkiraNetwork.VirtualStorage)

Gets a dictionary of wildcards and their corresponding regex patterns.

DefaultWildcardMatcher

#### Declaration

PowerShellWildcardMatcher

(AkiraNetwork.VirtualStorage)

```
public IReadOnlyDictionary<string, string> WildcardDictionary { get; }
```

### Property Value

Type	Description
ReadOnlyDictionary (https://learn.microsoft.com/dotnet/api/system.collections.objectmodel,readonlydictionary-2) <string (https://learn.microsoft.com/dotnet/api/system.string), string (https://learn.microsoft.com/dotnet/api/system.string)>	A read-only dictionary containing wildcards as keys and their corresponding regex patterns as values.

## Wildcards

Gets a collection of available wildcards.

### Declaration

```
public IEnumerable<string> Wildcards { get; }
```

### Property Value

Type	Description
IEnumerable ( <a href="https://learn.microsoft.com/dotnet/api/system.collections.generic.ienumerable-1">https://learn.microsoft.com/dotnet/api/system.collections.generic.ienumerable-1</a> ) <string ( <a href="https://learn.microsoft.com/dotnet/api/system.string">https://learn.microsoft.com/dotnet/api/system.string</a> )>	An enumerable collection of strings representing the available wildcards.

## + AkiraNetwork.VirtualStorage

### Methods

**(AkiraNetwork.VirtualStorage)**

**PatternMatcher(string, string)**

#### Library.Utilities

Determines whether the specified node name matches the specified pattern.

**(AkiraNetwork.VirtualStorage)**

Declaration

**- AkiraNetwork.VirtualStorage**

#### Library.WildcardMatchers

```
public bool PatternMatcher(string nodeName, string pattern)
```

**(AkiraNetwork.VirtualStorage)**

DefaultWildcardMatcher

**Parameters**  
(AkiraNetwork.VirtualStorageLibrary)

Type	Name	Description
PowerShellWildcardMatcher (AkiraNetwork.VirtualStorageLibrary)	<i>nodeName</i>	The node name to check.
string ( <a href="https://learn.microsoft.com/dotnet/api/system.string">https://learn.microsoft.com/dotnet/api/system.string</a> )	<i>pattern</i>	The pattern to compare against.

Returns

Type	Description
bool ( <a href="https://learn.microsoft.com/dotnet/api/system.boolean">https://learn.microsoft.com/dotnet/api/system.boolean</a> )	True if the node name matches the pattern; otherwise, false.

## Implements

IVirtualWildcardMatcher (AkiraNetwork.VirtualStorageLibrary.IVirtualWildcardMatcher.html)

## Extension Methods

VirtualTextFormatter.GenerateSingleTableDebugText<T>(T)

(AkiraNetwork.VirtualStorageLibrary.Utilities.VirtualTextFormatter.html#AkiraNetwork\_VirtualStorageLibrary\_Utils\_VirtualTextFormatter\_GenerateSingleTableDebugText\_1\_\_0\_)