

# Namespace AkiraNetwork.VirtualStorage Library

▼ Filter by title

## Classes

### - **AkiraNetwork.VirtualStorage**

#### **Library**

#### **VirtualCycleDetector**

(AkiraNetwork.VirtualStorageLibrary.VirtualCycleDetector.html)

ActionNodeDelegate in virtual symbolic links. Determines if a given link is part of a cycle.

(AkiraNetwork.VirtualStorageLibrary)

IVirtualDeepCloneable<T>

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

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

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

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

NotifyNodeDelegate

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

(AkiraNetwork.VirtualStorageLibrary)

PatternMatch

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

(AkiraNetwork.VirtualStorageLibrary)

VirtualDirectory

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

(AkiraNetwork.VirtualStorageLibrary)

VirtualException

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

VirtualGroupCondition<T, TKey>

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

(AkiraNetwork.VirtualStorageLibrary)

2.html)

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

(AkiraNetwork.VirtualStorageLibrary)

VirtualItem

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

(AkiraNetwork.VirtualStorageLibrary)

VirtualItem<T>

(AkiraNetwork.VirtualStorageLibrary)

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

VirtualNode

Represents a virtual item. This class serves as the base class for the VirtualItem<T>

(AkiraNetwork.VirtualStorageLibrary)

(AkiraNetwork.VirtualStorageLibrary.VirtualItem-1.html) type and is used solely for type checking of VirtualItem

(AkiraNetwork.VirtualStorageLibrary.VirtualItem.html).

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeExtensions

(AkiraNetwork.VirtualStorageLibrary)

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

VirtualNodeListConditions

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

VirtualNodeName

types of content.

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeNotFoundException

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.

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

Represents an abstract class for nodes.

### **- AkiraNetwork.VirtualStorage**

#### **Library**

#### **VirtualNodeContext**

([AkiraNetwork.VirtualStorageLibrary](#))

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

ActionNodeDelegate

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.

IVirtualDeepCloneable<T>

([AkiraNetwork.VirtualStorageLibrary](#))

#### **VirtualNodeExtensions**

([VirtualNodeExtensions](#))

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

([AkiraNetwork.VirtualStorageLibrary](#))

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

NotifyNodeDelegate

([AkiraNetwork.VirtualStorageLibrary](#))

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

([AkiraNetwork.VirtualStorageLibrary](#))

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

VirtualCycleDetector

([AkiraNetwork.VirtualStorageLibrary](#))

VirtualDirectory

([AkiraNetwork.VirtualStorageLibrary](#))

## **VirtualNodeNotFoundException**

([VirtualNodeNotFoundException](#))

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

([AkiraNetwork.VirtualStorageLibrary](#))

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

VirtualGroupCondition<T, TKey>

([AkiraNetwork.VirtualStorageLibrary](#))

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

VirtualID

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

VirtualItem

([AkiraNetwork.VirtualStorageLibrary](#))

## **VirtualSortCondition<T>**

([AkiraNetwork.VirtualStorageLibrary](#))

1.html)

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

([AkiraNetwork.VirtualStorageLibrary](#))

VirtualNodeContext

([AkiraNetwork.VirtualStorageLibrary](#))

## **VirtualStorageExtensions**

([VirtualStorageExtensions](#))

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

([AkiraNetwork.VirtualStorageLibrary](#))

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

VirtualNodeListConditions

([AkiraNetwork.VirtualStorageLibrary](#))

VirtualNodeName

([AkiraNetwork.VirtualStorageLibrary](#))

## **VirtualStorageSettings**

([VirtualStorageSettings](#))

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

VirtualNodeNotFoundException

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

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

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

### **- AkiraNetwork.VirtualStorage**

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

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

(AkiraNetwork.VirtualStorage)

ActionNodeDelegate

(AkiraNetwork.VirtualStorageLibrary)

IVirtualDeepCloneable

(AkiraNetwork.VirtualStorageLibrary) 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.

(AkiraNetwork.VirtualStorageLibrary)

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

PatternMatch

(AkiraNetwork.VirtualStorageLibrary)

VirtualCycleDetector

(AkiraNetwork.VirtualStorageLibrary)

VirtualDirectory

(AkiraNetwork.VirtualStorageLibrary)

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

VirtualException

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.

(AkiraNetwork.VirtualStorageLibrary)

2.html

VirtualID

## **VirtualNodeListConditions**

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

VirtualItem

Represents the conditions for creating a list of virtual nodes.

(AkiraNetwork.VirtualStorageLibrary)

VirtualItem<T>

(AkiraNetwork.VirtualStorageLibrary)

1.html

VirtualNode

(AkiraNetwork.VirtualStorageLibrary)

## **IVirtualDeepCloneable<T>**

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

Provides functionality to create deep clones of type T.

VirtualNodeExtensions

(AkiraNetwork.VirtualStorageLibrary)

## **IVirtualWildcardMatcher**

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

Provides functionality for wildcard matching.

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeNotFoundException

# Enums

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

Specifies the types of virtual nodes.

▼

## **VirtualNodeTypeFilter**

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

Specifies the filters for virtual node types.

## **(AkiraNetwork.VirtualStorageLibrary)**

### **ActionNodeDelegate**

#### **(AkiraNetwork.VirtualStorageLibrary)**

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

### **ActionNodeDelegate**

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

#### **IVirtualWildcardMatcher**

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

#### **(AkiraNetwork.VirtualStorageLibrary)**

##### **NotifyNodeDelegate**

### **NotifyNodeDelegate**

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

#### **(AkiraNetwork.VirtualStorageLibrary)**

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

##### **VirtualCycleDetector**

#### **(AkiraNetwork.VirtualStorageLibrary)**

### **PatternMatch**

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

#### **(AkiraNetwork.VirtualStorageLibrary)**

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

##### **VirtualException**

#### **(AkiraNetwork.VirtualStorageLibrary)**

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

#### **(AkiraNetwork.VirtualStorageLibrary)**

##### **2.html**

##### **VirtualID**

#### **(AkiraNetwork.VirtualStorageLibrary)**

##### **VirtualItem**

#### **(AkiraNetwork.VirtualStorageLibrary)**

##### **VirtualItem<T>**

#### **(AkiraNetwork.VirtualStorageLibrary)**

##### **1.html**

##### **VirtualNode**

#### **(AkiraNetwork.VirtualStorageLibrary)**

##### **VirtualNodeContext**

#### **(AkiraNetwork.VirtualStorageLibrary)**

##### **VirtualNodeExtensions**

#### **(AkiraNetwork.VirtualStorageLibrary)**

##### **VirtualNodeListConditions**

#### **(AkiraNetwork.VirtualStorageLibrary)**

##### **VirtualnodeName**

#### **(AkiraNetwork.VirtualStorageLibrary)**

##### **VirtualNodeNotFoundException**

# 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

Syntax

## (AkiraNetwork.VirtualStorage)

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

IVirtualDeepCloneable<T>

Parameters

| Type                   | Name            | Description                              |
|------------------------|-----------------|--|
| VirtualWildcardMatcher |                 |  |
| VirtualDirectory       | parentDirectory | The parent directory of the target node. |
| VirtualNodeDelegate    |                 |  |
| VirtualDirectoryName   | nodeName        | The name of the target node.             |
| VirtualPath            | nodePath        | The path of the target node.             |
| VirtualCycleDetector   |                 |  |
| VirtualPath            |                 |  |
| VirtualDirectory       |                 |  |

Returns

| Type             | Description   |
|------------------|---|
| VirtualException | Returns true to continue node traversal, or false to stop node traversal. |

Extension Methods

|   |  |
|---|--|
| VirtualTextFormatter<T>.GenerateSingleTableDebugText<T>(T)  |  |
| (AkiraNetwork.VirtualStorageLibrary.Utilities.VirtualTextFormatter.html#AkiraNetwork_VirtualStorageLibrary_Utilsi |  |
| s_VirtualTextFormatter_GenerateSingleTableDebugText_1__0_)  |  |
| VirtualNode   |  |
| (AkiraNetwork.VirtualStorageLibrary)  |  |
| VirtualNodeContext  |  |
| (AkiraNetwork.VirtualStorageLibrary)  |  |
| VirtualNodeExtensions   |  |
| (AkiraNetwork.VirtualStorageLibrary)  |  |
| VirtualNodeListConditions   |  |
| (AkiraNetwork.VirtualStorageLibrary)  |  |
| VirtualNodeName   |  |
| (AkiraNetwork.VirtualStorageLibrary)  |  |
| VirtualNodeNotFoundException  |  |

# 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

```
public interface IVirtualDeepCloneable<T>  
    where T : class  
        new();  
    IVirtualDeepCloneable<T>  
        DeepClone(bool recursive);
```

Type Parameters

1.html)

| Name                   | Description |
|------------------------|-------------|
| VirtualWildcardMatcher |             |

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

NotifyNodeDelegate

(AkiraNetwork.VirtualStorageLibrary)

PatternMatch

(AkiraNetwork.VirtualStorageLibrary)

VirtualCycleDetector

(AkiraNetwork.VirtualStorageLibrary)

VirtualDirectory

(AkiraNetwork.VirtualStorageLibrary)

DeepClone(bool)

(AkiraNetwork.VirtualStorageLibrary)

VirtualException

Creates a deep clone of the object.

(AkiraNetwork.VirtualStorageLibrary)

Declaration

virtual IVirtualDeepCloneable<T, TKey>

(AkiraNetwork.VirtualStorageLibrary)

T DeepClone(bool recursive = false)

VirtualID

(AkiraNetwork.VirtualStorageLibrary)

Parameters

VirtualItem

| Type           | Name      | Description  |
|----------------|-----------|--|
| VirtualItem<T> | 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. |

bool (AkiraNetwork.VirtualStorageLibrary)

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

VirtualNodeContext

Returns (AkiraNetwork.VirtualStorageLibrary)

| Type                                 | Description |
|--------------------------------------|-------------|
| (AkiraNetwork.VirtualStorageLibrary) |             |

T (AkiraNetwork.VirtualStorageLibrary)

VirtualNodeListConditions (AkiraNetwork.VirtualStorageLibrary) The cloned object.

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeName

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeNotFoundException

(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  
VirtualException  
(AkiraNetwork.VirtualStorageLibrar  
VirtualGroupCondition<T, TKey>  
(AkiraNetwork.VirtualStorageLibrar  
2.html)  
VirtualID  
(AkiraNetwork.VirtualStorageLibrar  
VirtualItem  
(AkiraNetwork.VirtualStorageLibrar  
VirtualItem<T>  
(AkiraNetwork.VirtualStorageLibrar  
1.html)  
VirtualNode  
(AkiraNetwork.VirtualStorageLibrar  
VirtualNodeContext  
(AkiraNetwork.VirtualStorageLibrar  
VirtualNodeExtensions  
(AkiraNetwork.VirtualStorageLibrar  
VirtualNodeListConditions  
(AkiraNetwork.VirtualStorageLibrar  
VirtualnodeName  
(AkiraNetwork.VirtualStorageLibrar  
VirtualNodeNotFoundException  
(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.VirtualStorageLibrary)

### Syntax

```
ActionNodeDelegate  
  (AkiraNetwork.VirtualStorageLibrary  
  public interface IVirtualWildcardMatcher  
  VirtualDeepCloneable<T>  
  (AkiraNetwork.VirtualStorageLibrary  
  1.html)  
  IVirtualWildcardMatcher  
  (AkiraNetwork.VirtualStorageLibrary  
  NotifyNodeDelegate  
  (AkiraNetwork.VirtualStorageLibrary  
  PatternMatch  
  (AkiraNetwork.VirtualStorageLibrary  
  VirtualDirectory  
  Gets VirtualNodeDelegates.  
  VirtualNodeDelegate  
  (AkiraNetwork.VirtualStorageLibrary  
 Declaration  
  VirtualDirectory  
  (AkiraNetwork.VirtualStorageLibrary  
  int Count { get; }  
  (AkiraNetwork.VirtualStorageLibrary  
  VirtualGroupCondition<T, TKey>  
  (AkiraNetwork.VirtualStorageLibrary  
  2.html)
```

## Properties

### Count

```
(AkiraNetwork.VirtualStorageLibrary  
 VirtualGroupCondition<T, TKey>
```

Gets VirtualNodeDelegates.

```
(AkiraNetwork.VirtualStorageLibrary  
 VirtualDirectory  
 (AkiraNetwork.VirtualStorageLibrary  
 int Count { get; }  
 (AkiraNetwork.VirtualStorageLibrary  
 VirtualGroupCondition<T, TKey>  
 (AkiraNetwork.VirtualStorageLibrary  
 2.html)
```

### Property Value

| Type | Description  |
|------|--|
| int  | An integer representing the number of wildcards available. |

## Patterns

```
(AkiraNetwork.VirtualStorageLibrary  
 1.html)
```

Gets a collection of available patterns.

```
VirtualNode  
 (AkiraNetwork.VirtualStorageLibrary  
 VirtualNodeContext  
 (AkiraNetwork.VirtualStorageLibrary  
 IEnumerable<string> Patterns { get; }  
 VirtualNodeExtensions  
 (AkiraNetwork.VirtualStorageLibrary  
 VirtualNodeExtensions  
 2.html)
```

### Property Value

```
VirtualNodeListConditions  
 (AkiraNetwork.VirtualStorageLibrary  
 VirtualNodeName  
 (AkiraNetwork.VirtualStorageLibrary  
 VirtualNodeNotFoundException  
 (AkiraNetwork.VirtualStorageLibrary  
 VirtualNodeType  
 2.html)
```

| Type   | Description  |
|--|--|
| IEnumerable<br>( <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> )<br><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

### WildcardDictionaryVirtualStorage

ActionNodeDelegate

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

Declaration

DeepCloneable<T>

(AkiraNetwork.VirtualStorageLibrary)

1.html

readonly Dictionary<string, string> WildcardDictionary { get; }

IVirtualWildcardMatcher

(AkiraNetwork.VirtualStorageLibrary)

Property Value

NotifyNodeDelegate

AkiraNetwork.VirtualStorageLibrary

PatternMatch

ReadOnlyDictionary

(AkiraNetwork.VirtualStorageLibrary)

(<https://learn.microsoft.com/dotnet/api/system.collections.objectmodel.virtualcycledetector.readonlydictionary>)

(AkiraNetwork.VirtualStorageLibrary)

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

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

(AkiraNetwork.VirtualStorageLibrary)

VirtualException

(AkiraNetwork.VirtualStorageLibrary)

VirtualGroupCondition<T, TKey>

(AkiraNetwork.VirtualStorageLibrary)

Gets a collection of available wildcards.

VirtualID

Declaration

(AkiraNetwork.VirtualStorageLibrary)

VirtualItem

IEnumerable<string, Wildcard>

(AkiraNetwork.VirtualStorageLibrary)

VirtualItem<T>

(AkiraNetwork.VirtualStorageLibrary)

Property Value

1.html

Type

VirtualNode

AkiraNetwork.VirtualStorageLibrary

VirtualNodeContext

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

(AkiraNetwork.VirtualStorageLibrary)

<VirtualNodeExtensions

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

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeListConditions

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeName

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeNotFoundException

(AkiraNetwork.VirtualStorageLibrary)

PatternMatcher(string, string)

VirtualNodeType

## Methods

VirtualNode

VirtualNodeContext

VirtualNodeName

VirtualNodeNotFoundException

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeType

Determines whether the specified node name matches the specified pattern.

## Declaration

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



### Parameters

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

Returns [VirtualWildcardMatcher](#)  
[1.html](#)

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

VirtualGroupCondition<T, TKey>

(AkiraNetwork.VirtualStorageLibrary)

2.html

VirtualID

(AkiraNetwork.VirtualStorageLibrary)

VirtualItem

(AkiraNetwork.VirtualStorageLibrary)

VirtualItem<T>

(AkiraNetwork.VirtualStorageLibrary)

1.html

VirtualNode

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeContext

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeExtensions

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeListConditions

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeName

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeNotFoundException

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeType

# Delegate NotifyNodeDelegate

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

Filter by title

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

**Assembly:** VirtualStorageLibrary.dll

ActionNodeDelegate

**Syntax**

```
(AkiraNetwork.VirtualStorageLibrary.IVirtualDeepCloneable<T>
public delegate void NotifyNodeDelegate(VirtualPath path, VirtualNode? node)
(AkiraNetwork.VirtualStorageLibrary.1.html)
```

IVirtualWildcardMatcher

**Parameters**

| Type               | Name        | Description                                  |
|--------------------|-------------|--|
| NotifyNodeDelegate | <i>path</i> | The virtual path of the node being notified. |
| VirtualNode        | <i>node</i> | The instance of the node being notified.     |

VirtualDirectory  
(AkiraNetwork.VirtualStorageLibrary.1.html)

## Extension Methods

VirtualTextFormatter.GenerateSingleTableDebugText<T>(T)  
VirtualGroupCondition<T, TKey>  
(AkiraNetwork.VirtualStorageLibrary.Utilities.VirtualTextFormatter.html#AkiraNetwork\_VirtualStorageLibrary\_Utilsilites\_VirtualTextFormatter\_GenerateSingleTableDebugText\_1\_\_0\_)  
2.html)

VirtualID

(AkiraNetwork.VirtualStorageLibrary.1.html)

VirtualItem

(AkiraNetwork.VirtualStorageLibrary.1.html)

VirtualItem<T>

(AkiraNetwork.VirtualStorageLibrary.1.html)

1.html)

VirtualNode

(AkiraNetwork.VirtualStorageLibrary.1.html)

VirtualNodeContext

(AkiraNetwork.VirtualStorageLibrary.1.html)

VirtualNodeExtensions

(AkiraNetwork.VirtualStorageLibrary.1.html)

VirtualNodeListConditions

(AkiraNetwork.VirtualStorageLibrary.1.html)

VirtualnodeName

(AkiraNetwork.VirtualStorageLibrary.1.html)

VirtualNodeNotFoundException

(AkiraNetwork.VirtualStorageLibrary.1.html)

VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary.1.html)

# 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)
```

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.

(AkiraNetwork.VirtualStorageLibrary)

**Returns:** VirtualException

Type (AkiraNetwork.VirtualStorageLibrary)

**Description**

VirtualGroupCondition<T, TKey>

bool (AkiraNetwork.VirtualStorageLibrary)

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

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

) VirtualID

(AkiraNetwork.VirtualStorageLibrary)

VirtualItem

(AkiraNetwork.VirtualStorageLibrary)

VirtualItem<T>

VirtualTextFormatter.GenerateSingleTableDebugText<T>(T)

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

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

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeContext

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeExtensions

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeListConditions

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeName

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeNotFoundException

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeTypeFilter

# Class VirtualCycleDetector

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

Filter by title

## Inheritance

(AkiraNetwork.VirtualStorageLibrary)  
↳ object (https://learn.microsoft.com/dotnet/api/system.object)  
↳ VirtualDeepCloneable<T>  
↳ VirtualCycleDetector  
(AkiraNetwork.VirtualStorageLibrary)

## Inherited Members

IVirtualWildcardMatcher  
object.Equals(object) (https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals(system-object))  
NotifyNodeDelegate  
object.Equals(object, object) (https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals(system-object-system-object))  
PatternMatch  
object.GetHashCode() (https://learn.microsoft.com/dotnet/api/system.object.gethashcode)  
(AkiraNetwork.VirtualStorageLibrary)  
object.GetType() (https://learn.microsoft.com/dotnet/api/system.object.gettype)  
VirtualCycleDetector  
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)  
VirtualDirectory  
object.ToString() (https://learn.microsoft.com/dotnet/api/system.object.tostring)  
(AkiraNetwork.VirtualStorageLibrary)

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

Assembly: AkiraNetwork.dll (AkiraNetwork.dll)

## Syntax

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

## Constructors

VirtualCycleDetector(VirtualNodeContext, VirtualNodeExtensions, VirtualNodeListConditions, VirtualNodeName, VirtualNodeType, VirtualNotFoundException, VirtualNodeTypeFilter, Count, VirtualStorageLibrary)

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

Declarations

```
(AkiraNetwork.VirtualStorageLibrary)
    VirtualNodeExtensions
        (AkiraNetwork.VirtualStorageLibrary)
    public VirtualCycleDetector()
        VirtualNodeListConditions
            (AkiraNetwork.VirtualStorageLibrary)
        VirtualnodeName
            (AkiraNetwork.VirtualStorageLibrary)
```

## Properties

VirtualNode  
VirtualNotFoundException  
VirtualNodeType  
VirtualNodeTypeFilter  
Count  
VirtualStorageLibrary

Gets the number of detected cycles (AkiraNetwork.VirtualStorageLibrary)

## 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> )<br>IVirtualDeepCloneable<T><br>(AkiraNetwork.VirtualStorageLibrary.<br>1.html)<br>VirtualWildcardMatcher<br>(AkiraNetwork.VirtualStorageLibrary.<br>Gets the dictionary of detected cycles.<br>NotifyNodeDelegate<br>Declaration<br>(AkiraNetwork.VirtualStorageLibrary.<br>PatternMatch<br>p(AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink) CycleDictionary { get; }<br>VirtualCycleDetector<br>(AkiraNetwork.VirtualStorageLibrary.<br>VirtualDirectory<br>Property Value<br>VirtualStorageLibrary | The number of detected cycles. |

## CycleDictionary

VirtualWildcardMatcher

Gets the dictionary of detected cycles.

NotifyNodeDelegate

Declaration  
(AkiraNetwork.VirtualStorageLibrary.

PatternMatch

p(AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink) CycleDictionary { get; }

VirtualCycleDetector

(AkiraNetwork.VirtualStorageLibrary.

VirtualDirectory  
Property Value  
VirtualStorageLibrary

| Type  | Description   |
|---|---|
| VirtualException<br>Dictionary<br>(AkiraNetwork.VirtualStorageLibrary.<br>( <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> )<br>VirtualGroupCondition<T, TKey><br>< (AkiraNetwork.VirtualStorageLibrary.<br>< 2.html)<br>VirtualID<br>VirtualID<br>(AkiraNetwork.VirtualStorageLibrary.VirtualID.html)<br>(AkiraNetwork.VirtualStorageLibrary.<br>, VirtualItem<br>VirtualSymbolicLink<br>(AkiraNetwork.VirtualStorageLibrary.<br>(AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLi<br>VirtualItem<T><br>nk.html)<br>> (AkiraNetwork.VirtualStorageLibrary.<br>1.html) | A dictionary where the keys are<br>VirtualID<br>(AkiraNetwork.VirtualStorageLibrary.VirtualID.html)<br>s and the values are the corresponding<br>VirtualSymbolicLink<br>(AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink.<br>html)<br>s. |

## Methods

VirtualNodeExtensions

(AkiraNetwork.VirtualStorageLibrary.

Clear()

VIRTUALNODEListConditions  
(AkiraNetwork.VirtualStorageLibrary.  
Clears the dictionary of detected cycles.  
VirtualNodeName

Declaration  
(AkiraNetwork.VirtualStorageLibrary.

VirtualNodeNotFoundException

(AkiraNetwork.VirtualStorageLibrary.

VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary.

VirtualNodeTypeFilter

(AkiraNetwork.VirtualStorageLibrary.

## IsNodeInCycle(VirtualSymbolicLink)

Determines whether the specified VirtualSymbolicLink  
(AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink.html) is part of a cycle.

### Declaration

```
public bool IsNodeInCycle(VirtualSymbolicLink link)  
(AkiraNetwork.VirtualStorageLibrary.IVirtualDeepCloneable<T>
```

### Parameters

| Type   | Name | Description  |
|--|------|--|
| VirtualSymbolicLink<br>(AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink.html) | link | The VirtualSymbolicLink<br>(AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink.html) to check. |

### Returns

| Type   | Description  |
|--|--|
| virtualDirectory<br>(AkiraNetwork.VirtualStorageLibrary.VirtualDirectory.html) | True if the link is part of a cycle; otherwise, false. |

## Extension Methods

|  |  |
|--|--|
| VirtualTextFormatter.GenerateSingleTableDebugText<T>(T<br>VirtualID<br>(AkiraNetwork.VirtualStorageLibrary.Utilities.VirtualTextFormatter.html#AkiraNetwork_VirtualStorageLibrary_Utils<br>s_VirtualTextFormatter_GeneratesSingleTableDebugText_1__0_) | VirtualItem<br>(AkiraNetwork.VirtualStorageLibrary.VirtualItem.html)                     |
| VirtualAllItem<T><br>(AkiraNetwork.VirtualStorageLibrary.VirtualAllItem.html)  | VirtualNode<br>(AkiraNetwork.VirtualStorageLibrary.VirtualNode.html)                     |
| VirtualNodeContext<br>(AkiraNetwork.VirtualStorageLibrary.VirtualNodeContext.html)   | VirtualNodeExtensions<br>(AkiraNetwork.VirtualStorageLibrary.VirtualNodeExtensions.html) |
| VirtualNodeListConditions<br>(AkiraNetwork.VirtualStorageLibrary.VirtualNodeListConditions.html)   | VirtualNodeName<br>(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)             |
| VirtualNodeNotFoundException<br>(AkiraNetwork.VirtualStorageLibrary.VirtualNodeNotFoundException.html)   | VirtualNodeType<br>(AkiraNetwork.VirtualStorageLibrary.VirtualNodeType.html)             |
| VirtualNodeTypeFilter<br>(AkiraNetwork.VirtualStorageLibrary.VirtualNodeTypeFilter.html)   |  |

# Class VirtualDirectory

Represents a virtual directory. A virtual directory functions as a container for other nodes. It can contain VirtualItem (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.VirtualSymbolicLink.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.VirtualSymbolicLink.html) according to the specified display conditions.

PatternMatch

Inheritance

VirtualCycleDetector

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

↳ VirtualNode (AkiraNetwork.VirtualStorageLibrary.VirtualNode.html)

↳ VirtualDirectory

↳ VirtualDirectory

(AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink.html)

Implementation

IVirtualDeepCloneable (AkiraNetwork.VirtualStorageLibrary.IVirtualDeepCloneable-1.html) <VirtualNode

(AkiraNetwork.VirtualStorageLibrary.VirtualNode.html)>

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

(AkiraNetwork.VirtualStorageLibrary.VirtualNode.html)>

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

(AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink.html)

Inherited Members

VirtualItem

VirtualNode (AkiraNetwork.VirtualStorageLibrary.VirtualNode.html)

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

VirtualNode (AkiraNetwork.CreatedDataStorageLibrary.VirtualNode.html)

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

VirtualNode (AkiraNetwork.UpdatedDataStorageLibrary.VirtualNode.html)

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

VirtualNode (AkiraNetwork.VirtualStorageLibrary.VirtualNode.html)

VirtualNode (AkiraNetwork.UpdatedDataStorageLibrary.VirtualNode.html)

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

VirtualNode (AkiraNetwork.VirtualStorageLibrary.VirtualNode.html)

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

VirtualNode (AkiraNetwork.BeforeConditionStorage.VirtualNode.html)

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

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

object (AkiraNetwork.Object (https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals(system-object)))

VirtualNodeNotFoundException

object (AkiraNetwork.Object (https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals(system-object)))

VirtualNode (AkiraNetwork.Object (https://learn.microsoft.com/dotnet/api/system.object.gethashcode))

object (AkiraNetwork.Object (https://learn.microsoft.com/dotnet/api/system.object.gettype))

VirtualNode (AkiraNetwork.Object (https://learn.microsoft.com/dotnet/api/system.object.memberwiseclone))

VirtualPath

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.VirtualDirectory.html)
1.html)
```

## Constructors

VirtualNodeMatcher

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeMatcher.html)

NotifyNodeDelegate

### VirtualDirectory()

(AkiraNetwork.VirtualStorageLibrary.VirtualDirectory.html)

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

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

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

VirtualDirectory

**Declaration**

(AkiraNetwork.VirtualStorageLibrary.VirtualException.html)

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

(AkiraNetwork.VirtualStorageLibrary.VirtualGroupCondition.html)

2.html)

## VirtualDirectory(VirtualNodeName)

(AkiraNetwork.VirtualStorageLibrary.VirtualItem.html)

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

(AkiraNetwork.VirtualStorageLibrary.VirtualItem.html)

**Declaration**

(AkiraNetwork.VirtualStorageLibrary.VirtualDirectory.html)

1.html)

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

VirtualNode

(AkiraNetwork.VirtualStorageLibrary.VirtualNode.html)

**Parameters**

VirtualNodeContext

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeExtensions.html)

| Type | Name | Description |
|------|------|-------------|
|------|------|-------------|

VirtualNodeExtensions

VirtualNodeName

(AkiraNetwork.VirtualStorageLibrary.VirtualnodeName.html)

(AkiraNetwork.VirtualStorageLibrary.VirtualnodeName.html)

VirtualnodeName

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeNotFoundException.html)

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

with the specified name, creation date, and update date.

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeNotFoundException.html)

**Declaration**

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeTypeFilter.html)

VirtualPath

```
public VirtualDirectory(VirtualNodeName name, DateTime createdDate, DateTime updatedDate)
```

## Parameters

| Type   |  | Name        | Description                |
|--|--|-------------|----------------------------|
| VirtualNodeName<br>(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)   |  | name        | The name of the directory. |
| DateTime<br>(https://learn.microsoft.com/dotnet/api/system.datetime<br>1.html) |  | createdDate | The creation date.         |
| DateTime<br>(https://learn.microsoft.com/dotnet/api/system.datetime<br>2.html) |  | updatedDate | The update date.           |

## Properties

**Count**  
(AkiraNetwork.VirtualStorageLibrary)

VirtualCycleDetector

**Count**  
(AkiraNetwork.VirtualStorageLibrary)

VirtualDirectory

Gets the total number of items within the directory.

VirtualException

**Declaration**  
(AkiraNetwork.VirtualStorageLibrary)

VirtualGroupCondition<T, TKey>

public int Count { get; }

(AkiraNetwork.VirtualStorageLibrary  
2.html)

VirtualID

**Property Value**  
(AkiraNetwork.VirtualStorageLibrary)

**Type** VirtualItem

**Description**

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

VirtualItem<T>

(AkiraNetwork.VirtualStorageLibrary  
1.html)

## DirectoryCount

VirtualNodeContext

**Declaration**  
(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeExtensions

public int DirectoryCount { get; }

VirtualNodeListConditions

(AkiraNetwork.VirtualStorageLibrary)

**Property Value**

VirtualNodeName

**Type** (AkiraNetwork.VirtualStorageLibrary)

**Description**

VirtualNodeNotFoundException

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

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeTypeFilter

(AkiraNetwork.VirtualStorageLibrary)

VirtualPath

**DirectoryViewCount**  
(AkiraNetwork.VirtualStorageLibrary)

VirtualPath

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

## Declaration

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

## Property Value

| Type   | Description |
|--|-------------|
| <code>virtualDeepCloneable&lt;T&gt;</code><br><code>(AkiraNetwork.VirtualStorageLibrary.int (https://learn.microsoft.com/dotnet/api/system.int32).html)</code> |             |

## IVirtualWildcardMatcher (AkiraNetwork.VirtualStorageLibrary)

this [VirtualNodeName]

(AkiraNetwork.VirtualStorageLibrary

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

(AkiraNetwork.VirtualStorageLibrary)

## Declaration

### VirtualCycleDetector

```
(AkiraNetwork.VirtualStorageLibrary  
public VirtualNode this[VirtualNodeName name] { get; set; }  
VirtualDirectory
```

(AkiraNetwork.VirtualStorageLibrary)

## VirtualException Parameters

Virtual ID  
Property Value

(AkiraNetwork.VirtualStorageLibrary)

| Type   | Description                       |
|--|-----------------------------------|
| VirtualItem<br>(AkiraNetwork.VirtualStorageLibrary.VirtualNode.html)<br>VirtualItem<T> | The node with the specified name. |

## **Exceptions**

| Type   | Condition                        |
|--|----------------------------------|
| virtualNode<br>(AkiraNetwork.VirtualStorageLibrary.VirtualNodeNotFoundException)             | Thrown if the node is not found. |
| VirtualNodeContext<br>(AkiraNetwork.VirtualStorageLibrary.VirtualNodeNotFoundException.html) |                                  |
| (AkiraNetwork.VirtualStorageLibrary.VirtualNodeNotFoundException.html)                       |                                  |

## VirtualNodeExtensions (AkiraNetwork.VirtualStorageLibrary)

## Item ~~Victor~~ ~~Call~~ No.

(AkiraNetwork VirtualStorage) library

**VirtualDirectoryName** Gets the number of items within the directory.

## Declaration

(AkiraNetwork.VirtualStorageLibrary  
VirtualNodeNotFoundException)

```
(AkiraNetwork.VirtualStorageLibrary  
public int ItemCount { get; }  
VirtualNodeType
```

(AkiraNetwork.VirtualStorageLibrary)

**VirtualNodeTypeFilter**  
(AkiraNetwork.VirtualStorageLibrary)

## VirtualPath

| 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**

VirtualWildcardMatcher

([AkiraNetwork.VirtualStorageLibrary](#))

VirtualCycleDetector

([AkiraNetwork.VirtualStorageLibrary](#))

VirtualDirectory

([AkiraNetwork.VirtualStorageLibrary](#))

VirtualException

Enum ([AkiraNetwork.VirtualStorageLibrary](#))

VirtualGroupCondition<T, TKey>

([AkiraNetwork.VirtualStorageLibrary](#))

2.html)

public IEnumerable<VirtualNodeName> NodeNames { get; }

([AkiraNetwork.VirtualStorageLibrary](#))

VirtualItem

([AkiraNetwork.VirtualStorageLibrary](#))

**Type:** VirtualItem<T>

**Description**

([AkiraNetwork.VirtualStorageLibrary](#))

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

1.html)

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

VirtualNode

([AkiraNetwork.VirtualStorageLibrary](#))

VirtualNodeContext

([AkiraNetwork.VirtualStorageLibrary](#))

**NodeType:**

([AkiraNetwork.VirtualStorageLibrary](#))

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

([AkiraNetwork.VirtualStorageLibrary](#))

**Declaration:** [VirtualNodeListConditions](#)

([AkiraNetwork.VirtualStorageLibrary](#))

public override VirtualNodeType NodeType { get; }

([AkiraNetwork.VirtualStorageLibrary](#))

VirtualNodeNotFoundException

**Property Value:** ([AkiraNetwork.VirtualStorageLibrary](#))

**Type:** VirtualNodeType

([AkiraNetwork.VirtualStorageLibrary](#))

**Description**

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

([AkiraNetwork.VirtualStorageLibrary](#))

**Overrides:** [VirtualPath](#)

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

## Nodes

Enumerates all nodes within the directory.

**Declaration**  
 (AkiraNetwork.VirtualStorageLibrary.VirtualNode.html#AkiraNetwork\_VirtualStorageLibrary\_VirtualNode\_Nodes)  
`public IEnumerable<VirtualNode> Nodes { get; }`

### Property Value

| Type                                  | Description  |
|---------------------------------------|--|
| PatternMatch.IEnumerable<VirtualNode> | (AkiraNetwork.VirtualStorageLibrary.VirtualNode.html#AkiraNetwork_VirtualStorageLibrary_VirtualNode_Nodes) |

## NodesView

Gets a view of nodes based on the current display conditions.  
`VirtualGroupCondition<T, TKey>`

**Declaration**  
 (AkiraNetwork.VirtualStorageLibrary.VirtualNode.html#AkiraNetwork\_VirtualStorageLibrary\_VirtualNode\_NodesView)  
`public VirtualGroupCondition<T, TKey> NodesView { get; }`

**Property Value**  
 (AkiraNetwork.VirtualStorageLibrary.VirtualNode.html#AkiraNetwork\_VirtualStorageLibrary\_VirtualNode\_NodesView)  
`VirtualGroupCondition<T, TKey>`

| Type                           | Description  |
|--------------------------------|--|
| VirtualGroupCondition<T, TKey> | (AkiraNetwork.VirtualStorageLibrary.VirtualNode.html#AkiraNetwork_VirtualStorageLibrary_VirtualNode_NodesView) |

**NodesViewCount**  
 Gets the number of nodes based on the current display conditions.

**Declaration**  
 (AkiraNetwork.VirtualStorageLibrary.VirtualNode.html#AkiraNetwork\_VirtualStorageLibrary\_VirtualNode\_NodesViewCount)  
`public int NodesViewCount { get; }`

**Property Value**  
 (AkiraNetwork.VirtualStorageLibrary.VirtualNode.html#AkiraNetwork\_VirtualStorageLibrary\_VirtualNode\_NodesViewCount)  
`int`

## SymbolicLinkCount

Gets the number of symbolic links within the directory.

### Declaration

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

| Property Value  | Type  |
|---|---|
| IVirtualDeepCloneable<T><br>(AkiraNetwork.VirtualStorageLibrary.VirtualDeepCloneable<T>.html) | int ( <a href="https://learn.microsoft.com/dotnet/api/system.int32">https://learn.microsoft.com/dotnet/api/system.int32</a> )<br>(AkiraNetwork.VirtualStorageLibrary.VirtualInt32.html) |

| Property Value  | Type  | Description |
|---|---|-------------|
| IVirtualWildcardMatcher<br>(AkiraNetwork.VirtualStorageLibrary.VirtualWildcardMatcher.html) | int ( <a href="https://learn.microsoft.com/dotnet/api/system.int32">https://learn.microsoft.com/dotnet/api/system.int32</a> )<br>(AkiraNetwork.VirtualStorageLibrary.VirtualInt32.html) |             |

## SymbolicLinkViewCount

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

| Declaration   | Type   |
|---|--|
| AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLinkViewCount<br>(AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLinkViewCount.html) | VirtualDirectory<br>(AkiraNetwork.VirtualStorageLibrary.VirtualDirectory.html) |

| Property Value  | Type  | Description |
|---|---|-------------|
| IVirtualMatchupCondition<T, TKey><br>(AkiraNetwork.VirtualStorageLibrary.VirtualMatchupCondition<T, TKey>.html) | int ( <a href="https://learn.microsoft.com/dotnet/api/system.int32">https://learn.microsoft.com/dotnet/api/system.int32</a> )<br>(VirtualMatchupCondition.html) |             |

## Methods

| Add(VirtualNode, bool)  | Type  |
|---|---|
| (AkiraNetwork.VirtualStorageLibrary.VirtualNodeExtensions.Add.html) | (AkiraNetwork.VirtualStorageLibrary.VirtualNodeExtensions.html) |

| Add(VirtualNode, bool)  | Type  | Description                   |
|---|---|-------------------------------|
| (AkiraNetwork.VirtualStorageLibrary.VirtualNodeExtensions.Add.html) | (AkiraNetwork.VirtualStorageLibrary.VirtualNodeExtensions.html) | Adds a node to the directory. |

| Add(VirtualNode, bool)  | Type  | Description                   |
|---|---|-------------------------------|
| (AkiraNetwork.VirtualStorageLibrary.VirtualNodeExtensions.Add.html) | (AkiraNetwork.VirtualStorageLibrary.VirtualNodeExtensions.html) | Adds a node to the directory. |

| Add(VirtualNode, bool)  | Type  | Description                   |
|---|---|-------------------------------|
| (AkiraNetwork.VirtualStorageLibrary.VirtualNodeExtensions.Add.html) | (AkiraNetwork.VirtualStorageLibrary.VirtualNodeExtensions.html) | Adds a node to the directory. |

| Add(VirtualNode, bool)  | Type  | Description                   |
|---|---|-------------------------------|
| (AkiraNetwork.VirtualStorageLibrary.VirtualNodeExtensions.Add.html) | (AkiraNetwork.VirtualStorageLibrary.VirtualNodeExtensions.html) | Adds a node to the directory. |

| Add(VirtualNode, bool)  | Type  | Description                   |
|---|---|-------------------------------|
| (AkiraNetwork.VirtualStorageLibrary.VirtualNodeExtensions.Add.html) | (AkiraNetwork.VirtualStorageLibrary.VirtualNodeExtensions.html) | Adds a node to the directory. |

| Add(VirtualNode, bool)  | Type  | Description                   |
|---|---|-------------------------------|
| (AkiraNetwork.VirtualStorageLibrary.VirtualNodeExtensions.Add.html) | (AkiraNetwork.VirtualStorageLibrary.VirtualNodeExtensions.html) | Adds a node to the directory. |

| Add(VirtualNode, bool)  | Type  | Description                   |
|---|---|-------------------------------|
| (AkiraNetwork.VirtualStorageLibrary.VirtualNodeExtensions.Add.html) | (AkiraNetwork.VirtualStorageLibrary.VirtualNodeExtensions.html) | Adds a node to the directory. |

| Add(VirtualNode, bool)  | Type  | Description                   |
|---|---|-------------------------------|
| (AkiraNetwork.VirtualStorageLibrary.VirtualNodeExtensions.Add.html) | (AkiraNetwork.VirtualStorageLibrary.VirtualNodeExtensions.html) | Adds a node to the directory. |

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

## Returns

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

## Exceptions

| Type  | Condition   |
|---|---|
| <code>IVirtualWildcardMatcher</code><br>( <a href="#">AkiraNetwork.VirtualStorageLibrary.Arguments.Matcher.html</a> )   | Thrown if an invalid node name is specified.        |
| <code>InvalidOperationException</code><br>( <a href="https://learn.microsoft.com/dotnet/api/system.invalidoperationexception">https://learn.microsoft.com/dotnet/api/system.invalidoperationexception</a> ) | Thrown if a node with the same name already exists. |

## AddDirectory(VirtualDirectoryName, bool)

Adds a directory with the specified name.  
[2.html](#)

### Declaration

```
VirtualDirectory AddDirectory(VirtualDirectoryName name, bool allowOverwrite = false)
```

### Parameters

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

| Type   | Name                        | Description  |
|--|-----------------------------|--|
| <code>VirtualNode</code><br><code>VirtualDirectoryName</code><br>( <a href="#">AkiraNetwork.VirtualStorageLibrary.VirtualDirectoryName.html</a> )  | <code>name</code>           | The name of the directory.                                       |
| <code>bool</code><br><code>VirtualNodeExtensions</code><br>( <a href="https://learn.microsoft.com/dotnet/api/system.boolean">https://learn.microsoft.com/dotnet/api/system.boolean</a> ) | <code>allowOverwrite</code> | If true, allows overwriting an existing node with the same name. |

## Returns

| Type  | Description          |
|---|----------------------|
| <code>VirtualNodeNotFoundException</code><br>( <a href="#">AkiraNetwork.VirtualStorageLibrary.VirtualNodeNotFoundException.html</a> ) | The added directory. |

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

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

Adds an item with the specified name.

## Declaration

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



### Parameters

| Type   | Name           | Description  |
|--|----------------|--|
| AkiraNetwork.VirtualStorageLibrary   | name           | The name of the item.  |
| VirtualNodeName<br>IVirtualWildcardMatcher<br>(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)<br>NotifyNodeDelegate                        | itemData       | The data of the item.  |
| T<br>PatternMatch<br>bool<br>(AkiraNetwork.VirtualStorageLibrary<br>(https://learn.microsoft.com/dotnet/api/system.boolean))<br>VirtualCycleDetector | allowOverwrite | If true, allows overwriting an existing node with the same name. |

### Returns

AkiraNetwork.VirtualStorageLibrary

| Type   | Description     |
|--|-----------------|
| VirtualException<br>(AkiraNetwork.VirtualStorageLibrary<br>VirtualItem<AkiraNetwork.VirtualStorageLibrary.VirtualItem-1.html> <T><br>VirtualGroupCondition, Key<br>(AkiraNetwork.VirtualStorageLibrary | The added item. |

### Type Parameters

| Name | Description                |
|------|----------------------------|
| T    | The data type of the item. |

## AddSymbolicLink(VirtualNodeName, VirtualPath, bool)

Adds a symbolic link with the specified name.

(AkiraNetwork.VirtualStorageLibrary

## Declaration

VirtualNodeContext

(AkiraNetwork.VirtualStorageLibrary

```
virtual NodeExtension<VirtualPath> AddSymbolicLink(VirtualNodeName name, VirtualPath targetPath, bool ignoreCase = false)
```

(AkiraNetwork.VirtualStorageLibrary

### Parameters

VirtualNodeName

| Type                               | Name | Description                    |
|------------------------------------|------|--------------------------------|
| AkiraNetwork.VirtualStorageLibrary | name | The name of the symbolic link. |

| Type   | Name                  | Description  |
|--|-----------------------|--|
| VirtualPath<br>(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)   | <i>targetPath</i>     | The target path of the symbolic link.                            |
| bool<br>(https://learn.microsoft.com/dotnet/api/system.boolean)<br>VirtualDeepCloneable<T><br>(AkiraNetwork.VirtualStorageLibrary.VirtualDeepCloneable.html) | <i>allowOverwrite</i> | If true, allows overwriting an existing node with the same name. |

Returns [VirtualSymbolicLink.html](#)

| Type   | Description              |
|--|--------------------------|
| VirtualWildcardMatcher<br>(AkiraNetwork.VirtualStorageLibrary.VirtualWildcardMatcher.html) | The added symbolic link. |

## DeepClone(bool)

VirtualCycleDetector

Creates a deep clone of the directory.

VirtualDirectory

Declaration

```
public NetworkVirtualStorageLibrary.VirtualNode DeepClone(bool recursive = false)
```

VirtualGroupCondition<T, TKey>

Parameters

2.html)

| Type   | Name             | Description   |
|--|------------------|---|
| virtualID<br>(AkiraNetwork.VirtualStorageLibrary.VirtualID.html) | <i>recursive</i> | If true, all nodes within the directory are recursively cloned. |

Returns [VirtualNode.html](#)

| Type   | Description                    |
|--|--------------------------------|
| VirtualNode<br>(AkiraNetwork.VirtualStorageLibrary.VirtualNode.html) | A deep clone of the directory. |

## Overloads

VirtualNodeExtensions

VirtualNode.DeepClone(bool)

(AkiraNetwork.VirtualStorageLibrary.VirtualNode.html#AkiraNetwork\_VirtualStorageLibrary\_VirtualNode\_DeepClone\_System\_Boolean)

e\_System\_Boolean)VirtualStorageLibrary

VirtualnodeName

(AkiraNetwork.VirtualStorageLibrary.VirtualStorageLibrary.html#AkiraNetwork\_VirtualStorageLibrary\_VirtualStorageLibrary\_VirtualnodeName)

VirtualNodeNotFoundException

## DirectoryExists(VirtualNodeName)

(AkiraNetwork.VirtualStorageLibrary.VirtualStorageLibrary.html#AkiraNetwork\_VirtualStorageLibrary\_VirtualStorageLibrary\_DirectoryExists\_System\_String)

Checks whether a directory with the specified name exists.

VirtualNodeType

Declaration

```
public static bool DirectoryExists(string virtualNodeName)
```

VirtualNodeTypeFilter

(AkiraNetwork.VirtualStorageLibrary.VirtualStorageLibrary.html#AkiraNetwork\_VirtualStorageLibrary\_VirtualStorageLibrary\_VirtualNodeTypeFilter)

VirtualPath



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

#### Parameters

| Type   | Name | Description                |
|--|------|----------------------------|
| VirtualNodeName<br>(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html) | name | The name of the directory. |

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)

#### Returns

| Type  | Description |
|---|-------------|
| IVirtualWildcardMatcher<br>(AkiraNetwork.VirtualStorageLibrary.VirtualWildcardMatcher.html) |             |

VirtualDirectoryDelegate

(AkiraNetwork.VirtualStorageLibrary.VirtualDirectory.html)

PatternMatch

(AkiraNetwork.VirtualStorageLibrary.VirtualMatch.html)

#### Exceptions

| Type   | Condition                        |
|--|----------------------------------|
| VirtualCycleDetector<br>(AkiraNetwork.VirtualStorageLibrary.VirtualCycleDetector.html)         |                                  |
| VirtualNotFoundException<br>(AkiraNetwork.VirtualStorageLibrary.VirtualNotFoundException.html) | Thrown if the node is not found. |

VirtualException

InvalidOperationException

(AkiraNetwork.VirtualStorageLibrary.InvalidOperationException.html)

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

Thrown if the specified node is not a directory.

(AkiraNetwork.VirtualStorageLibrary.VirtualGroupCondition.html)

2.html)

#### GetEnumerator()

(AkiraNetwork.VirtualStorageLibrary.VirtualItem.html)

Gets **VirtualItem** for nodes based on the current display conditions.

(AkiraNetwork.VirtualStorageLibrary.VirtualItem.html)

#### Declaration

VirtualItem<T>

(AkiraNetwork.VirtualStorageLibrary.VirtualItem.html)

```
public IEnumarator<VirtualNode> GetEnumerator()
```

VirtualNode

(AkiraNetwork.VirtualStorageLibrary.VirtualNode.html)

#### Returns

VirtualNodeContext

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeContext.html)

#### Type

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeContext.html)

#### Description

IEnumarator  
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeContext.html)

An enumerator for filtered nodes.

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

(AkiraNetwork.VirtualStorageLibrary.VirtualNode.html)

VirtualNodeName

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)

#### GetItem<T>(VirtualNodeName)

(AkiraNetwork.VirtualStorageLibrary.VirtualItem.html)

Gets the item with the specified name.

VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeType.html)

#### Declaration

VirtualNodeTypeFilter

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeTypeFilter.html)

VirtualPath

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

#### Parameters

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

Return Value  
AkiraNetwork.VirtualStorageLibrary.VirtualItem<T>

| Type                    | Description |
|-------------------------|-------------|
| IVirtualWildcardMatcher |             |

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

NotifyNodeDelegate

Type Parameters  
(AkiraNetwork.VirtualStorageLibrary)

PatternMatch

Name  
(AkiraNetwork.VirtualStorageLibrary)

T VirtualCycleDetector The data type of the item.

(AkiraNetwork.VirtualStorageLibrary)

VirtualDirectory

(AkiraNetwork.VirtualStorageLibrary)

Type VirtualException

(AkiraNetwork.VirtualStorageLibrary)

VirtualNotFoundException

(AkiraNetwork.VirtualStorageLibrary)

VirtualGroupCondition<T, TKey>

(AkiraNetwork.VirtualStorageLibrary)

(AkiraNetwork.VirtualStorageLibrary)

InvalidOperationException

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

(AkiraNetwork.VirtualStorageLibrary)

VirtualItem

(AkiraNetwork.VirtualStorageLibrary)

VirtualItem<T>

(AkiraNetwork.VirtualStorageLibrary)

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

1.html)

Declaration

VirtualNode

(AkiraNetwork.VirtualStorageLibrary)

IVirtualNodeContext<VirtualNode> GetNodesView()

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeExtensions

Returns

(AkiraNetwork.VirtualStorageLibrary)

Type virtualNodeListConditions

(AkiraNetwork.VirtualStorageLibrary)

IEnumerable<VirtualNodeName>

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

(AkiraNetwork.VirtualStorageLibrary)

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

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary)

GetSymbolicLink(VirtualNodeName)

(AkiraNetwork.VirtualStorageLibrary)

Gets the symbolic link with the specified name.

VirtualPath

## Declaration

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

### Parameters

| Type  | Name | Description                    |
|---|------|--------------------------------|
| VirtualNodeName<br>(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)<br>1.html) | name | The name of the symbolic link. |

### Returns

| Type  | Description                                |
|---|--|
| VirtualWildcardMatcher<br>(AkiraNetwork.VirtualStorageLibrary.NotifyNodeDelegate) | The symbolic link with the specified name. |

### Exceptions

| Type   | Condition  |
|--|--|
| VirtualDirectory<br>(AkiraNetwork.VirtualStorageLibrary.VirtualNotFoundException)  | Thrown if the node is not found.                     |
| VirtualException<br>(AkiraNetwork.VirtualStorageLibrary.VirtualNodeNotFoundException.html)<br>I) (AkiraNetwork.VirtualStorageLibrary.VirtualGroupCondition<T, TKey><br>InvalidOperationException<br>(AkiraNetwork.VirtualStorageLibrary. <a href="https://learn.microsoft.com/dotnet/api/system.invalidoperationexception.2.html">https://learn.microsoft.com/dotnet/api/system.invalidoperationexception.2.html</a> )<br>n) VirtualID<br>(AkiraNetwork.VirtualStorageLibrary.VirtualItem<br>VirtualItem<br>(AkiraNetwork.VirtualStorageLibrary.VirtualItem<T>)) | Thrown if the specified node is not a symbolic link. |

## ItemExists(VirtualNodeName)

### Parameters

| Type  | Name | Description                    |
|---|------|--------------------------------|
| VirtualNodeListConditions<br>(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName) | name | The name of the item to check. |

### Returns

| Type   | Description                                |
|--|--|
| VirtualNodeType<br>(AkiraNetwork.VirtualStorageLibrary. <a href="https://learn.microsoft.com/dotnet/api/system.boolean">https://learn.microsoft.com/dotnet/api/system.boolean</a> )<br>VirtualNodeTypeFilter<br>(AkiraNetwork.VirtualStorageLibrary.VirtualPath) | 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><br>(AkiraNetwork.VirtualStorageLibrary.html) |             |                                |
|--|---|-------------|--------------------------------|
| Type   |   | Name        | Description                    |
| <code>VirtualWildcardMatcher</code><br><code>VirtualNodeName</code><br>(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html) |   | <i>name</i> | The name of the node to check. |
| <code>NotifyNodeDelegate</code>  |   |             |                                |

| <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.

## Declaration (Akira N)

lation  
?html)

```
VirtualID  
public void Remove(VirtualNode node)  
(AkiraNetwork VirtualStorage) library
```

## VirtualItem

| Parameter   | Network.VirtualStorageLibrary.VirtualItem<T><br>(AkiraNetwork.VirtualStorageLibrary) | Name        | Description         |
|-------------|--|-------------|---------------------|
| VirtualNode | (AkiraNetwork.VirtualStorageLibrary.VirtualNode.html)                                | <i>node</i> | The node to remove. |

Excom (Akira N)

| <b>Type</b>  | <b>Condition</b>                 |
|--|----------------------------------|
| VirtualNodeContext<br>(AkiraNetwork.VirtualStorageLibrary)                                     |                                  |
| VirtualNodeException<br>(AkiraNetwork.VirtualStorageLibrary.VirtualNodeNotFoundException.html) | Thrown if the node is not found. |

**SymbolicLinkExists(VirtualNodeName)**

#### VirtualNodeNotFoundException

**FileExists** Checks whether a file with the specified name exists.

### **DefaultVirtualNodeType**

(AlienNet and Virtex-4) can withstand

(AkiraNetwork.virtual)

```
public bool SymbolicLinkExists(VirtualnodeName name)
```

## (AkiraNetwork.VirtualStorageLibrary)

## VirtualPath

## Parameters

| Type   | Name | Description                             |
|--|------|---|
| VirtualNodeName<br>(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html) | name | The name of the symbolic link to check. |

## Returns

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

## IVirtualWildcardMatcher

### (AkiraNetwork.VirtualStorageLibrary.VirtualNode)

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

## Tostring()

### (AkiraNetwork.VirtualStorageLibrary.VirtualNode)

Returns a string representation of the directory.

## PatternMatch

### (AkiraNetwork.VirtualStorageLibrary.VirtualNode)

VirtualCycleDetector  
(AkiraNetwork.VirtualStorageLibrary.VirtualNode)  
public override string ToString()  
VirtualDirectory  
(AkiraNetwork.VirtualStorageLibrary.VirtualNode)

## Return

### VirtualException

| Type   | Description                               |
|--|---|
| VirtualGroupCondition<T, TKey><br>(AkiraNetwork.VirtualStorageLibrary.VirtualNode) | A string representation of the directory. |

## VirtualID

### (AkiraNetwork.VirtualStorageLibrary.VirtualNode)

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

## VirtualItem

### (AkiraNetwork.VirtualStorageLibrary.VirtualNode)

## Update(VirtualNode)

### (AkiraNetwork.VirtualStorageLibrary.VirtualNode)

Updates the current directory with the data from the specified node.

## VirtualNode

### (AkiraNetwork.VirtualStorageLibrary.VirtualNode)

VirtualNodeContext  
public override void Update(VirtualNode node)  
(AkiraNetwork.VirtualStorageLibrary.VirtualNode)

## VirtualNodeExtensions

### (AkiraNetwork.VirtualStorageLibrary.VirtualNode)

| Type  | Name | Description                   |
|---|------|-------------------------------|
| VirtualNodeListConditions<br>(AkiraNetwork.VirtualStorageLibrary.VirtualNode) | node | The node used for the update. |

## VirtualNodeNotFoundException

### (AkiraNetwork.VirtualStorageLibrary.VirtualNode)

VirtualNode.Update(VirtualNode)

(AkiraNetwork.VirtualStorageLibrary.VirtualNode.html#AkiraNetwork\_VirtualStorageLibrary\_VirtualNode\_Update\_A  
(AkiraNetwork.VirtualStorageLibrary.VirtualNode)  
kiraNetwork\_VirtualStorageLibrary\_VirtualNode\_)

## VirtualNodeTypeFilter

### (AkiraNetwork.VirtualStorageLibrary.VirtualNode)

## VirtualPath

| Type   | Condition   |
|--|---|
| ArgumentException<br>( <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 VirtualDirectory. |

1

# Operators

VirtualDeep

IVirtualDeepCloneable<T>

## (AkiraNetwork.VirtualStorageLibrary)

## **operator +(VirtualDirectory, VirtualNode)**

## IVirtualWildcardMatcher

Over([Add New](#)) add virtual strategy block a node to the directory.

## NotifyNodeDelegate

**Declaration** (AkiraNetwork VirtualStorage library)

```
PatternMatch  
public static VirtualDirectory operator +(VirtualDirectory directory, VirtualNode node)  
{  
    return new VirtualDirectory(directory, node);  
}
```

## VirtualCycleDetector

#### - VirtualCycleDetector

## **Virtual Directories**

| Type   | Name      | Description                               |
|--|-----------|---|
| VirtualDirectory<br>(AkiraNetwork.VirtualStorageLibrary)                     |           |   |
| VirtualException<br>VirtualDirectory<br>(AkiraNetwork.VirtualStorageLibrary) | directory | The directory to which the node is added. |
| VirtualGroupCondition<T, TKey><br>(AkiraNetwork.VirtualStorageLibrary)       |           |   |
| VirtualNode<br>(AkiraNetwork.VirtualStorageLibrary)                          | node      | The node to add.                          |

## **VirtualID**

Return AkiraNetwork.VirtualStorageLibrary

| Type  | Description                        |
|---|------------------------------------|
| VirtualItem<br>(AkiraNetwork.VirtualStorageLibrary) | The directory with the added node. |

**VirtualNode  
operator + (VirtualDirectory, VirtualNodeName)**

(AkiraNetwork.VirtualStorageLibrary)

**VirtualNodeContext**  
Overloads the addition operator to add

(AkiraNetwork.VirtualSt

## VirtualNodeExtensions

## (AkiraNetwork.VirtualStorageLibrary)

## **VirtualNodeListConditions** directory

© AkiraNetwork. VirtualStorageLibrary

## VirtualNodeName

(AkiraNetwork.VirtualStorageLibrary)

#### **VirtualNodeNotFoundException**

| Type             | (AkiraNetwork.VirtualStorageLibrary.VirtualNodeType)       | Name             | Description                               |
|------------------|--|------------------|---|
| VirtualDirectory | (AkiraNetwork.VirtualStorageLibrary.VirtualDirectory.html) | <i>directory</i> | The directory to which the node is added. |

## (AkiraNetwork.VirtualStorageLibrary

## VirtualPath

| Type   | Name     | Description                        |
|--|----------|------------------------------------|
| VirtualNodeName<br>(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html) | nodeName | The node name of directory to add. |

## Returns

| Type  | Description                        |
|---|------------------------------------|
| IVirtualDeepCloneable<T><br>VirtualDirectory (AkiraNetwork.VirtualStorageLibrary.VirtualDirectory.html)<br>(AkiraNetwork.VirtualStorageLibrary.VirtualDirectory.html) | The directory with the added node. |

## implicit operator VirtualDirectory(VirtualNodeName)

NotifyNodeDelegate  
Implicitly converts the specified node name to a VirtualDirectory  
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)  
(AkiraNetwork.VirtualStorageLibrary.VirtualDirectory.html).

### Declaration

```
VirtualCycleDetector
(AkiraNetwork.VirtualStorageLibrary.VirtualGroupCondition<T, TKey>
public static implicit operator VirtualDirectory(VirtualNodeName nodeName)
    VirtualDirectory
    (AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)
```

### Parameters

| Type   | Name     | Description               |
|--|----------|---------------------------|
| VirtualNodeName<br>(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html) | nodeName | The node name to convert. |

### Return Value

| Type  | Description |
|---|-------------|
| VirtualItem<br>(AkiraNetwork.VirtualStorageLibrary.VirtualDirectory.html) |             |

## implicit operator VirtualDirectory(string)

VirtualNodeContext  
Implicitly converts the specified node name to a VirtualDirectory  
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)  
(AkiraNetwork.VirtualStorageLibrary.VirtualDirectory.html).

### Declaration

```
VirtualNodeListConditions
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)
VirtualNodeName
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)
```

### Parameters

| Type   | Name     | Description               |
|--|----------|---------------------------|
| VirtualNodeType<br>string ( <a href="https://learn.microsoft.com/dotnet/api/system.string">https://learn.microsoft.com/dotnet/api/system.string</a> )<br>(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html) | nodeName | The node name to convert. |

### Return Value

| Type                             | Description |
|----------------------------------|-------------|
| VirtualNodeFilter<br>VirtualPath |             |

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

## operator -(VirtualDirectory, VirtualNode)

Overloads the subtraction operator to remove a node from the directory.

IVirtualDeepCloneable<T>

Declaration: Network.VirtualStorageLibrary

1.html)

VirtualWildcardMatcher

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

(AkiraNetwork.VirtualStorageLibrary)

NotifyNodeDelegate

Parameters:

(AkiraNetwork.VirtualStorageLibrary)

Type: PatternMatch

(AkiraNetwork.VirtualStorageLibrary)

VirtualDirectory

(AkiraNetwork.VirtualStorageLibrary.VirtualDirectory.html)

(AkiraNetwork.VirtualStorageLibrary)

VirtualNode

(AkiraNetwork.VirtualStorageLibrary.VirtualNode.html)

VirtualException

(AkiraNetwork.VirtualStorageLibrary)

Returns: VirtualGroupCondition<T, TKey>

Type: AkiraNetwork.VirtualStorageLibrary

2.html)

VirtualDirectory

(AkiraNetwork.VirtualStorageLibrary.VirtualDirectory.html)

(AkiraNetwork.VirtualStorageLibrary)

VirtualItem

(AkiraNetwork.VirtualStorageLibrary)

Implements:

IVirtualItem

(AkiraNetwork.VirtualStorageLibrary)

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

1.html)

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

VirtualNode

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

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeContext

(AkiraNetwork.VirtualStorageLibrary)

Extension Methods:

VirtualNodeExtensions

VirtualTextFormatter.GenerateSingleTableDebugText<T>(T)

(AkiraNetwork.VirtualStorageLibrary)

(AkiraNetwork.VirtualStorageLibrary.Utilities.VirtualTextFormatter.html#AkiraNetwork\_VirtualStorageLibrary\_Utilsilites\_VirtualNodeListConditions)

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

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeExtensions.ResolveNodeType(VirtualNode)

VirtualNodeName

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

(AkiraNetwork.VirtualStorageLibrary)

Extensions.ResolveNodeType\_AkiraNetwork\_VirtualStorageLibrary\_VirtualNode\_

\_VirtualNodeNotFoundException

VirtualTextFormatter.GenerateTableDebugText<T>(IEnumerable<T>)

(AkiraNetwork.VirtualStorageLibrary)

(AkiraNetwork.VirtualStorageLibrary.Utilities.VirtualTextFormatter.html#AkiraNetwork\_VirtualStorageLibrary\_Utilsilites\_VirtualTextFormatter\_GenerateTableDebugText\_1\_System\_Collections\_Generic\_IEnumerable\_0\_)

(AkiraNetwork.VirtualStorageLibrary)

VirtualStorageExtensions.ApplySortConditions<T>(IEnumerable<T>, List<VirtualSortCondition<T>>?)

VirtualNodeTypeFilter

(AkiraNetwork.VirtualStorageLibrary.VirtualStorageExtensions.html#AkiraNetwork\_VirtualStorageLibrary\_VirtualSto)

(AkiraNetwork.VirtualStorageLibrary)

VirtualPath

```
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
VirtualException
(AkiraNetwork.VirtualStorageLibrar
VirtualGroupCondition<T, TKey>
(AkiraNetwork.VirtualStorageLibrar
2.html)
VirtualID
(AkiraNetwork.VirtualStorageLibrar
VirtualItem
(AkiraNetwork.VirtualStorageLibrar
VirtualItem<T>
(AkiraNetwork.VirtualStorageLibrar
1.html)
VirtualNode
(AkiraNetwork.VirtualStorageLibrar
VirtualNodeContext
(AkiraNetwork.VirtualStorageLibrar
VirtualNodeExtensions
(AkiraNetwork.VirtualStorageLibrar
VirtualNodeListConditions
(AkiraNetwork.VirtualStorageLibrar
VirtualnodeName
(AkiraNetwork.VirtualStorageLibrar
VirtualNodeNotFoundException
(AkiraNetwork.VirtualStorageLibrar
VirtualNodeType
(AkiraNetwork.VirtualStorageLibrar
VirtualNodeTypeFilter
(AkiraNetwork.VirtualStorageLibrar
VirtualPath
```

# Class VirtualException

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

Filter by title

## Inheritance

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

## Implements

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

VirtualCycleDetector

## Inherited Members

ExceptionBaseException() (<https://learn.microsoft.com/dotnet/api/system.exception.getbaseexception>)  
ExceptionGetType() (<https://learn.microsoft.com/dotnet/api/system.exception.gettype>)  
ExceptionToString() (<https://learn.microsoft.com/dotnet/api/system.exception.tostring>)  
ExceptionData (<https://learn.microsoft.com/dotnet/api/system.exception.data>)  
ExceptionHelpLink () (<https://learn.microsoft.com/dotnet/api/system.exception.helplink>)  
ExceptionHRESULT () ([https://learn.microsoft.com/dotnet/api/system.exception\\_hresult](https://learn.microsoft.com/dotnet/api/system.exception_hresult))  
ExceptionInnerException () (<https://learn.microsoft.com/dotnet/api/system.exception.innerexception>)  
ExceptionMessage () (<https://learn.microsoft.com/dotnet/api/system.exception.message>)  
ExceptionSource () (<https://learn.microsoft.com/dotnet/api/system.exception.source>)  
ExceptionStackTrace () (<https://learn.microsoft.com/dotnet/api/system.exception.stacktrace>)  
ExceptionTargetSite () (<https://learn.microsoft.com/dotnet/api/system.exception.targetsite>)  
ExceptionSerializeObjectState () (<https://learn.microsoft.com/dotnet/api/system.exception.serializeobjectstate>)  
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)))  
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)))  
object.GetHashCode () (<https://learn.microsoft.com/dotnet/api/system.object.gethashcode>)  
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)

**Namespace:** AkiraNetwork (<https://learn.microsoft.com/dotnet/api/AkiraNetwork.html>).VirtualStorageLibrary (<https://learn.microsoft.com/dotnet/api/AkiraNetwork.VirtualStorageLibrary.html>)

**Assembly:** VirtualStorageLibrary.dll

(AkiraNetwork.VirtualStorageLibrary)

## Syntax

VirtualNodeName

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeNotFoundException : VirtualException : Exception, ISerializable

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeTypeFilter

(AkiraNetwork.VirtualStorageLibrary)

VirtualPath

## Constructors

### VirtualException()

(AkiraNetwork.VirtualStorageLibrary)

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

## Declaration

```
public VirtualException()
```

▼

### VirtualException(string)

1.html)

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

#### Declaration

(AkiraNetwork.VirtualStorageLibrary)

PatternMatch

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

(AkiraNetwork.VirtualStorageLibrary)

VirtualCycleDetector

Parameters

(AkiraNetwork.VirtualStorageLibrary)

Type VirtualDirectory

(AkiraNetwork.VirtualStorageLibrary)

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

Name

Description

message

The error message.

### VirtualException(string, Exception)

2.html)

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. (AkiraNetwork.VirtualStorageLibrary)

VirtualItem

Parameters

(AkiraNetwork.VirtualStorageLibrary)

Type VirtualItem<T>

(AkiraNetwork.VirtualStorageLibrary) message, Exception innerException)

1.html)

VirtualNode

Parameters

(AkiraNetwork.VirtualStorageLibrary)

Type VirtualNodeContext

(AkiraNetwork.VirtualStorageLibrary)

string

(VirtualNodeExtensions

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

(AkiraNetwork.VirtualStorageLibrary)

Name

Description

message

The error message.

Exception VirtualNodeListConditions

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

VirtualnodeName

(AkiraNetwork.VirtualStorageLibrary)

innerException

The exception that is the cause of this exception.

## Implements

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

VirtualNodeTypeFilter

(AkiraNetwork.VirtualStorageLibrary)

VirtualPath

(AkiraNetwork.VirtualStorageLibrary)

# Extension Methods

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



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

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

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

**Inherited Members**

(AkiraNetwork.VirtualStorageLibrary)  
object.Equals(object) ([https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals\(system-object\)](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)))  
VirtualCycleDetector  
(AkiraNetwork.VirtualStorageLibrary)  
object.GetHashCode() (<https://learn.microsoft.com/dotnet/api/system.object.gethashcode>)  
VirtualDirectory  
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>)  
(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**

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

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

| Name        | Description                         |
|-------------|-------------------------------------|
| VirtualNode | The type of the item to be grouped. |
| T           | The type of the item to be grouped. |

**Constructors**

VirtualGroupConditions  
(AkiraNetwork.VirtualStorageLibrary)

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

VirtualNodeNotFoundException  
Initializes a new instance of the VirtualGroupCondition<T, TKey>  
(AkiraNetwork.VirtualStorageLibrary)

(VirtualGroupCondition-2.html) class with the specified grouping property and order.

VirtualNodeType  
(AkiraNetwork.VirtualStorageLibrary)

Declaration  
NodeFilter  
(AkiraNetwork.VirtualStorageLibrary)

VirtualPath  
(AkiraNetwork.VirtualStorageLibrary)

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

## Parameters

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

## Properties

VirtualException  
(AkiraNetwork.VirtualStorageLibrary)

VirtualCycleDetector  
(AkiraNetwork.VirtualStorageLibrary)

VirtualHD  
(AkiraNetwork.VirtualStorageLibrary)

VirtualGroupCondition<T, TKey>  
(AkiraNetwork.VirtualStorageLibrary)

Gets or sets a value indicating whether the grouping order is ascending. True if the order is ascending; otherwise, false.  
[2.html](#)

Declaration

(AkiraNetwork.VirtualStorageLibrary)

```
public bool Ascending { get; set; }
```

VirtualItem<T>  
(AkiraNetwork.VirtualStorageLibrary)

Property Value  
(AkiraNetwork.VirtualStorageLibrary)

| Type   | Description   |
|--|---|
| VirtualNode<br>bool<br>(AkiraNetwork.VirtualStorageLibrary)<br>( <a href="https://learn.microsoft.com/dotnet/api/system.boolean">https://learn.microsoft.com/dotnet/api/system.boolean</a> )<br>lean<br>(AkiraNetwork.VirtualStorageLibrary) | A boolean value indicating whether the grouping order is ascending. |

## GroupBy

VirtualNode  
(AkiraNetwork.VirtualStorageLibrary)

Gets VirtualNode property used for grouping.

VirtualNode  
(AkiraNetwork.VirtualStorageLibrary)

Declaration

(AkiraNetwork.VirtualStorageLibrary)

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

VirtualNodeType  
(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeTypeFilter  
(AkiraNetwork.VirtualStorageLibrary)

Property Value  
(AkiraNetwork.VirtualStorageLibrary)

VirtualPath  
(AkiraNetwork.VirtualStorageLibrary)

| Type  | Description  |
|---|--|
| Expression<br>( <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> )<br><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

(AkiraNetwork.VirtualStorageLibrary.html)

VirtualTextFormatter.MatcherSingleTableDebugText<T>(T)  
(AkiraNetwork.VirtualStorageLibrary.Utilities.VirtualTextFormatter.html#AkiraNetwork\_VirtualStorageLibrary\_Utilities\_VirtualTextFormatter\_MatcherSingleTableDebugText\_1\_\_0\_)

(AkiraNetwork.VirtualStorageLibrary.PatternMatch)

(AkiraNetwork.VirtualStorageLibrary.VirtualCycleDetector)

(AkiraNetwork.VirtualStorageLibrary.VirtualDirectory)

(AkiraNetwork.VirtualStorageLibrary.VirtualException)

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

(AkiraNetwork.VirtualStorageLibrary.VirtualID)  
(AkiraNetwork.VirtualStorageLibrary.VirtualItem)

(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.VirtualPath)

(AkiraNetwork.VirtualStorageLibrary.VirtualTable)

# Struct VirtualID

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.

(AkiraNetwork.VirtualStorageLibrary  
1.html)

## Implements

IVirtualWildcardMatcher  
IEquatable<VirtualID>  
(AkiraNetwork.VirtualStorageLibrary.VirtualID.html)  
NotifyNodeDelegate

## Inherited Members

(AkiraNetwork.VirtualStorageLibrary)

PatternMatch  
ValueType.Equals(object) (https://learn.microsoft.com/dotnet/api/system.valuetype.equals)  
ValueType.GetHashCode() (https://learn.microsoft.com/dotnet/api/system.valuetype.gethashcode)  
object.Equals(object, object) (https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals(system-object-system-object))  
VirtualDirectory  
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

**Syntax**

(AkiraNetwork.VirtualStorageLibrary  
2.html)

**VirtualID** Readonly record struct VirtualID : IEquatable<VirtualID>

(AkiraNetwork.VirtualStorageLibrary)

VirtualItem

(AkiraNetwork.VirtualStorageLibrary)

## Constructors

(AkiraNetwork.VirtualStorageLibrary  
1.html)

### VirtualID()

(AkiraNetwork.VirtualStorageLibrary  
VirtualNode)

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

**Declaration**

(AkiraNetwork.VirtualStorageLibrary  
VirtualNodeListConditions)

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeName

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeNotFoundException

(AkiraNetwork.VirtualStorageLibrary)

## Properties

(AkiraNetwork.VirtualStorageLibrary  
VirtualNodeType)

VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary  
VirtualNodeTypeFilter)

(AkiraNetwork.VirtualStorageLibrary)

Gets the unique identifier.  
VirtualPath

(AkiraNetwork.VirtualStorageLibrary  
Declaration)

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

## Property Value

| Type  | Description  |
|---|--|
| Guid<br>( <a href="https://learn.microsoft.com/dotnet/api/system.guid">https://learn.microsoft.com/dotnet/api/system.guid</a> ) | 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. |

IVirtualWildcardMatcher  
(AkiraNetwork.VirtualStorageLibrary)

NotifyNodeDelegate  
(AkiraNetwork.VirtualStorageLibrary)

PatternMatch  
(AkiraNetwork.VirtualStorageLibrary)

## Methods

VirtualCycleDetector  
(AkiraNetwork.VirtualStorageLibrary)

Returns a string representation of this instance's unique identifier.

(AkiraNetwork.VirtualStorageLibrary)

Declaration  
VirtualException  
(AkiraNetwork.VirtualStorageLibrary)

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

## Returns

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

## Overrides

(AkiraNetwork.VirtualStorageLibrary)

ValueType.ToString() (<https://learn.microsoft.com/dotnet/api/system.valuetype.tostring>)  
1.html)

VirtualNode

## Implements

(AkiraNetwork.VirtualStorageLibrary)

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

VirtualNodeExtensions

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeListConditions

VirtualTextFormatter.GenerateSingleTableDebugText<T>(T)

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

s\_VirtualNodeName  
(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeNotFoundException

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeTypeFilter

(AkiraNetwork.VirtualStorageLibrary)

VirtualPath

(AkiraNetwork.VirtualStorageLibrary)

## Extension Methods

(AkiraNetwork.VirtualStorageLibrary)

VirtualTextFormatter.GenerateSingleTableDebugText<T>(T)

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

s\_VirtualNodeName  
(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeNotFoundException

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeTypeFilter

(AkiraNetwork.VirtualStorageLibrary)

VirtualPath

(AkiraNetwork.VirtualStorageLibrary)



## **Assembly: VirtualStorageLibrary.dll**

## Syntax

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

1

# Constructors

(AkiraNetwork.VirtualStorageLibrary  
1.html)

**VirtualItem(VirtualNodeName)**

(AkiraNetwork.VirtualStorageLibrary

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

Nethaji  
(Akira N)

(AK) aNetwork.virtualStorageLibrary  
declaration

```
PatternMatch  
(AkiraNetwork.VirtualStorageLibrar  
protected VirtualItem(VirtualNodeName name)  
VirtualCycleDetector
```

### Virtual Directories

| Type  | Name | Description           |
|---|------|-----------------------|
| (AkiraNetwork.VirtualStorageLibrary.VirtualException) | name | The name of the item. |

**VirtualItem(VirtualNodeName, DateTime)**

(AkiraNetwork.VirtualStorageLibrary)

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

## Declaration

**VirtualItem<T>**

(VirtualNetworkVirtualStorageLibraryModeName name, DateTime createdDate)  
1.html)

VirtualN

**Parameters** (AkiraNetwork VirtualStorage library)

| Type   | Name               | Description                    |
|--|--------------------|--------------------------------|
| VirtualNodeContext<br>(AkiraNetwork.VirtualStorageLibrary)                       |                    |                                |
| VirtualNodeName<br>VirtualNodeExtensions<br>(AkiraNetwork.VirtualStorageLibrary) | <i>name</i>        | The name of the item.          |
| VirtualNodeListConditions<br>(AkiraNetwork.VirtualStorageLibrary)                | <i>createdDate</i> | The creation date of the item. |

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

(ARRA NETWORK: VirtualStorageList) [1]

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

## VirtualNetwork.virtual

**Node type filter**  
(All Nodes and Mappings)

(AkiraNetw

## VirtualPath

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

## Parameters

| Type   |  | Name               | Description                       |
|--|--|--------------------|-----------------------------------|
| VirtualNodeName<br>(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)   | 1.html)  | <i>name</i>        | The name of the item.             |
| DateTime<br>VirtualWildcardMatcher<br>(https://learn.microsoft.com/dotnet/api/system.datetime)<br>(AkiraNetwork.VirtualStorageLibrary) | VirtualWildcardMatcher<br>(https://learn.microsoft.com/dotnet/api/system.datetime) | <i>createdDate</i> | The creation date of the item.    |
| DateTime<br>NodeDelegate<br>(https://learn.microsoft.com/dotnet/api/system.datetime)   | NodeDelegate<br>(https://learn.microsoft.com/dotnet/api/system.datetime)           | <i>updatedDate</i> | The last update date of the item. |

# Methods

## VirtualDirectory (AkiraNetwork.VirtualStorageLibrary)

# DeepClone(bool)

(AkiraNetwork.VirtualStorageLibrary)

## Create Virtual Environments and Test Key Features

(Akira Nakamura's studio original sketch)

**VirtualID Declaration**

**VirtualItem**

public override abstract VirtualItem<T>

## Parameters

Type virtualNode

bool [VirtualNodeContext](https://learn.microsoft.com/dotnet/api/AkiraNetwork.VirtualStorageLibrary.VirtualNodeContext)  
(<https://learn.microsoft.com/dotnet/api/AkiraNetwork.VirtualStorageLibrary>)

**VirtualNodeExtensions**  
**Returns**

(AkiraNetwork.VirtualStorageLibrary)  
**Type**: virtualNodeListConditions

(AkiraNetwork.VirtualStorageLibrary  
VirtualNode  
VirtualNodeName  
(All Nodes))

(AkiraNetwork.VirtualStorageLibrary.VirtualNode.html)  
VirtualNodeNotFoundException

**Overrides** [VirtualStorageLibrary.OverriddenProperties](#)

VirtualNode type  
VirtualNode DeepClone(bpool)  
AkiraNetwork VirtualStorageLibrary  
(AkiraNetwork VirtualStorageLibrary Virt

(AKT) VirtualNodeLibrary.virtual  
e\_System.Boolean) VirtualStorageLibrary.

VirtualPath  
(AkiraNetwork.VirtualStorageLibrary)

| Type   | Name      | Description                                      |
|--|-----------|--|
| bool (AkiraNetwork.VirtualStorageLibrary.VirtualNodeContext (https://learn.microsoft.com/dotnet/api/system.boolean) (Akira Network Virtual Storage Library)) | recursive | In the case of items, this parameter is ignored. |

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

**Overrides** [VirtualStorageLibrary.OverriddenProperties](#)

VirtualNode type  
VirtualNode DeepClone(bpool)  
AkiraNetwork VirtualStorageLibrary  
(AkiraNetwork VirtualStorageLibrary Virt

(AKT) VirtualNodeLibrary.virtual  
e\_System.Boolean) VirtualStorageLibrary.

VirtualPath  
(AkiraNetwork.VirtualStorageLibrary)

# Implements

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

## Extension Methods

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

# 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.VirtualItem<T>.html)  
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.

NotifyNodeDelegate

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

Inheritance

PatternMatch

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

↳ VirtualNodeDetector

(AkiraNetwork.VirtualStorageLibrary.VirtualNode.html)

↳ VirtualItem (AkiraNetwork.VirtualStorageLibrary.VirtualItem.html)

↳ VirtualDirectory

↳ VirtualItem<T>

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

Implements

VirtualException

IVirtualDeepCloneable (AkiraNetwork.VirtualStorageLibrary.IVirtualDeepCloneable-1.html) <VirtualNode

(AkiraNetwork.VirtualStorageLibrary.VirtualNode.html)>

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

(AkiraNetwork.VirtualStorageLibrary.VirtualItem.html)

Inherited Members

VirtualID

VirtualNode.Name

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

VirtualItem

VirtualNode.CreatedDate

(AkiraNetwork.VirtualStorageLibrary.VirtualNode.html)

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

ate)

(AkiraNetwork.VirtualStorageLibrary.VirtualNode.html)

VirtualNode.UpdatedDate

VirtualNode.html

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

Date)

(AkiraNetwork.VirtualStorageLibrary.VirtualNode.html)

VirtualNode.VID

VirtualNodeContext

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

(AkiraNetwork.VirtualStorageLibrary.VirtualNode.html)

VirtualNode.IsReferencedInStorage

(AkiraNetwork.VirtualStorageLibrary.VirtualNode.html)

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

cedinStorage)

VirtualNodeListConditions

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.VirtualItem.html)

VirtualNodeName

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.VirtualItem.html)

VirtualNodeNotFoundException

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

(AkiraNetwork.VirtualStorageLibrary.VirtualItem.html)

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

(AkiraNetwork.VirtualStorageLibrary.VirtualItem.html)

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

(AkiraNetwork.VirtualStorageLibrary.VirtualItem.html)

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

(AkiraNetwork.VirtualStorageLibrary.VirtualItem.html)

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

Assembly VirtualStorageLibrary.dll

(AkiraNetwork.VirtualStorageLibrary.VirtualItem.html)



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

(AkiraNetwork.VirtualStorageLibrary-1.html)

## VirtualItem(VirtualNodeName, T?, DateTime)

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

(AkiraNetwork.VirtualStorageLibrary)

### Declaration

PatternMatch

(AkiraNetwork.VirtualStorageLibrary)

```
public VirtualItem(VirtualNodeName name, T? itemData, DateTime createdDate)
```

(AkiraNetwork.VirtualStorageLibrary)

### Parameters

(AkiraNetwork.VirtualStorageLibrary)

#### Type

VirtualException

#### Name

#### Description

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeName

*name*

The name of the item.

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

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)

2.html)

T VirtualID

(AkiraNetwork.VirtualStorageLibrary)

*itemData*

The data associated with this item. It is user-defined and can hold any type of data.

VirtualItem

DateTime

(AkiraNetwork.VirtualStorageLibrary)

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

(AkiraNetwork.VirtualStorageLibrary)

1.html)

VirtualNode

(AkiraNetwork.VirtualStorageLibrary)

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

VirtualNodeContext

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

(AkiraNetwork.VirtualStorageLibrary)

### Declaration

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeListConditions

```
public VirtualItem(VirtualNodeName name, T? itemData, DateTime createdDate, DateTime updatedDate)
```

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeName

(AkiraNetwork.VirtualStorageLibrary)

### Parameters

(AkiraNetwork.VirtualStorageLibrary)

#### Type

VirtualNodeType

#### Name

#### Description

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeName

*name*

The name of the item.

VirtualPath

(AkiraNetwork.VirtualStorageLibrary)

| Type   | Name               | Description   |
|--|--------------------|---|
| T  | <i>itemData</i>    | The data associated with this item. It is user-defined and can hold any type of data. |
| DateTime<br>( <a href="https://learn.microsoft.com/dotnet/api/system.datetime">https://learn.microsoft.com/dotnet/api/system.datetime</a> )  | <i>createdDate</i> | The creation date of the item.  |
| DateTime<br>( <a href="https://learn.microsoft.com/dotnet/api/system.datetime">https://learn.microsoft.com/dotnet/api/system.datetime</a> )<br>VirtualWildcardMatcher<br>( <a href="https://learn.microsoft.com/dotnet/api/system.datetime">https://learn.microsoft.com/dotnet/api/system.datetime</a> )<br>NotifyNodeDelegate<br>(AkiraNetwork.VirtualStorageLibrary.VirtualWildcardMatcher.NotifyNodeDelegate) | <i>updatedDate</i> | The last update date of the item.   |
| PatternMatch<br>(AkiraNetwork.VirtualStorageLibrary.VirtualWildcardMatcher.PatternMatch)   |                    |   |

## Properties

### ItemData

Gets or sets the data associated with this item. This value can be null.  
VirtualException

### Declaration

```
AkiraNetwork.VirtualStorageLibrary.VirtualGroupCondition<T, TKey>
public virtual T ItemData { get; set; }
```

VirtualID

### PropertyValue

#### Type

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

### NodeType

Gets the type of the node.  
VirtualNode

### Declaration

```
VirtualNodeExtensions
public override VirtualNodeType NodeType { get; }
public static class VirtualNodeListConditions
{
    public static void AddConditions(IList<VirtualNode> nodes)
    {
        foreach (VirtualNode node in nodes)
        {
            if (node.NodeType == VirtualNodeType.Folder)
            {
                node.AddCondition("IsFolder", true);
            }
            else if (node.NodeType == VirtualNodeType.File)
            {
                node.AddCondition("IsFile", true);
            }
        }
    }
}
```

### PropertyName

#### Type

| Type                         | Description   |
|------------------------------|---|
| VirtualNodeNotFoundException | (AkiraNetwork.VirtualStorageLibrary.VirtualNodeNotFoundException) |
| VirtualNodeType              | (AkiraNetwork.VirtualStorageLibrary.VirtualNodeType)              |

### Overrides

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

# Methods

## DeepClone(bool)

Creates a deep clone of this item.

### Declaration

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

IVirtualWildcardMatcher

(AkiraNetwork.VirtualStorageLibrary)
Parameters

NotifyNodeDelegate

| Type  | Name      | Description                                |
|---|-----------|--|
| AkiraNetwork.VirtualStorageLibrary PatternMatch bool (AkiraNetwork.VirtualStorageLibrary ( <a href="https://learn.microsoft.com/dotnet/api/system.boolean">https://learn.microsoft.com/dotnet/api/system.boolean</a> ) VirtualCycleDetector | recursive | If true, perform a deep clone recursively. |

(AkiraNetwork.VirtualStorageLibrary)
Returns

VirtualDirectory

| Type  | Description  |
|---|--|
| AkiraNetwork.VirtualStorageLibrary VirtualException VirtualNode (AkiraNetwork.VirtualStorageLibrary (AkiraNetwork.VirtualStorageLibrary.VirtualGroupCondition<T, TKey> INode.html)) (AkiraNetwork.VirtualStorageLibrary 2.html) | A deep clone of the current item as a VirtualNode (AkiraNetwork.VirtualStorageLibrary.VirtualNode.html). |

### Overrides

VirtualNode.DeepClone(bool)
VirtualNode.DeepClone()
(AkiraNetwork.VirtualStorageLibrary.VirtualItem.html#AkiraNetwork\_VirtualStorageLibrary\_VirtualItem\_DeepClone\_
System.Boolean)
VirtualStorageLibrary

VirtualItem<T>

(AkiraNetwork.VirtualStorageLibrary)
**Dispose()**

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

### Declaration

VirtualNodeContext
(AkiraNetwork.VirtualStorageLibrary)
VirtualNodeExtensions
VirtualNode.Dispose()
(AkiraNetwork.VirtualStorageLibrary)
VirtualNodeListConditions
(AkiraNetwork.VirtualStorageLibrary)
VirtualNodeName
(AkiraNetwork.VirtualStorageLibrary)

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

VirtualNodeType
(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeTypeFilter

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

VirtualPath

(AkiraNetwork.VirtualStorageLibrary)

### Parameters

| Type  | Name             | Description   |
|---|------------------|---|
| bool<br>( <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;<br>false to release only unmanaged resources. |

▼

## ~VirtualItem()

(AkiraNetwork.VirtualStorageLibrary.VirtualItem-1.html)  
Finalizer for the VirtualItem<T> (AkiraNetwork.VirtualStorageLibrary.VirtualItem-1.html) class.  
VirtualWildcardMatcher

### Declaration

```
NotifyNodeDelegate
protected VirtualItem()
PatternMatch
VirtualCycleDetector
```

## ToString()

(AkiraNetwork.VirtualStorageLibrary.VirtualItem-1.html)  
Returns the string representation of the current object.

VirtualException

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

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

### Returns

(AkiraNetwork.VirtualStorageLibrary.VirtualItem)

| Type        | Description                                  |
|-------------|--|
| VirtualItem | A string that represents the current object. |

### Overrides

(AkiraNetwork.VirtualStorageLibrary.VirtualItem-1.html)  
object.ToString() (<https://learn.microsoft.com/dotnet/api/system.object.tostring>)  
VirtualNode

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeContext)

## Update(VirtualNode)

(AkiraNetwork.VirtualStorageLibrary.VirtualNode)

Updates the node's data with the specified node's data.

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeListConditions)

```
public override void Update(VirtualNode node)
VirtualNodeName
```

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeNotFoundException)

(AkiraNetwork.VirtualStorageLibrary.VirtualNode)

Parameters  
Type

VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary.VirtualNode)

VirtualNodeTypeFilter

(AkiraNetwork.VirtualStorageLibrary.VirtualNode.html)

(AkiraNetwork.VirtualStorageLibrary.VirtualPath)

### Overrides

(AkiraNetwork.VirtualStorageLibrary.VirtualPath)

### Type

Name

| Type            | Name        | Description                             |
|-----------------|-------------|---|
| VirtualNodeType | <i>node</i> | 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<br><a href="https://learn.microsoft.com/dotnet/api/system.argumentexception">https://learn.microsoft.com/dotnet/api/system.argumentexception</a><br>(AkiraNetwork.VirtualStorageLibrary.VirtualNode.html#AkiraNetwork_VirtualStorageLibrary_VirtualNode_ArgumentException) | Thrown if the specified node is not of the same type as the current item. |

## Operators

VirtualNodeDelegate

(AkiraNetwork.VirtualStorageLibrary.VirtualNode.html#AkiraNetwork\_VirtualStorageLibrary\_VirtualNodeDelegate)

PatternMatch

## implicit operator VirtualItem<T>(VirtualnodeName)

VirtualCycleDetector

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

VirtualDirectory

(AkiraNetwork.VirtualStorageLibrary.VirtualItem.html#AkiraNetwork\_VirtualStorageLibrary\_VirtualItem\_VirtualDirectory)

Declaration

VirtualException

(AkiraNetwork.VirtualStorageLibrary.VirtualItem.html#AkiraNetwork\_VirtualStorageLibrary\_VirtualItem\_VirtualException)

```
public static implicit operator VirtualItem<T>(VirtualnodeName name)
  VirtualGroupCondition<T, TKey>
  (AkiraNetwork.VirtualStorageLibrary.VirtualItem.html#AkiraNetwork_VirtualStorageLibrary_VirtualItem_VirtualGroupCondition_T_KKey)
```

## Parameters

| Type  | Name | Description           |
|---|------|-----------------------|
| VirtualID<br>(AkiraNetwork.VirtualStorageLibrary.VirtualItem.html#AkiraNetwork_VirtualStorageLibrary_VirtualItem_VirtualID) |      |                       |
| VirtualnodeName<br>(AkiraNetwork.VirtualStorageLibrary.VirtualnodeName.html)  | name | The name of the node. |

## Returns

VirtualItem<T>

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

## implicit operator VirtualItem<T>(string)

VirtualnodeName

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

VirtualNodeNotFoundException

Declaration

VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary.VirtualItem.html#AkiraNetwork\_VirtualStorageLibrary\_VirtualItem\_VirtualNodeType)

VirtualNodeTypeFilter

(AkiraNetwork.VirtualStorageLibrary.VirtualItem.html#AkiraNetwork\_VirtualStorageLibrary\_VirtualItem\_VirtualNodeTypeFilter)

## Parameters

VirtualPath

(AkiraNetwork.VirtualStorageLibrary.VirtualItem.html#AkiraNetwork\_VirtualStorageLibrary\_VirtualItem\_VirtualPath)

| Type   | Name | Description           |
|--|------|-----------------------|
| string ( <a href="https://learn.microsoft.com/dotnet/api/system.string">https://learn.microsoft.com/dotnet/api/system.string</a> ) | name | The name of the node. |

## Returns

| Type  | Description  |
|---|--|
| VirtualItem<br>(AkiraNetwork.VirtualStorageLibrary.VirtualItem-1.html)<br>allItem-1.html)<br><T>IVirtualWildcardMatcher<br>(AkiraNetwork.VirtualStorageLibrary.NotifyNodeDelegate<br>(AkiraNetwork.VirtualStorageLibrary) | A<br>VirtualItem<T> (AkiraNetwork.VirtualStorageLibrary.VirtualItem-1.html)<br>initialized with the specified node name. |

## implicit operator VirtualItem<T>((VirtualnodeName, T? itemData))

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

### Declaration

```

VirtualException
public static implicit operator VirtualItem<T>((VirtualnodeName nodeName, T? itemData) tuple)
(AkiraNetwork.VirtualStorageLibrary)
VirtualGroupCondition<T, TKey>
(AkiraNetwork.VirtualStorageLibrary)

```

### Parameters

| Type  | Name | Description |
|---|------|-------------|
| VirtualID<br>(AkiraNetwork.VirtualStorageLibrary) |      |             |

### Returns

| Type  | Description |
|---|-------------|
| VirtualNodeExtensions<br>(AkiraNetwork.VirtualStorageLibrary) |             |

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

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  
VirtualPath  
GeneratenodeName(string)  
(AkiraNetwork.VirtualStorageLibrary)

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

## Declaration

```
public static implicit operator VirtualItem<T>(T? itemData)
```

### Parameters

(AkiraNetwork.VirtualStorageLibrary.

| Type | Name     | Description   |
|------|----------|---|
| T    | itemData | The data associated with this item. It is user-defined and can hold any type of data. |

NotifyNodeDelegate

Returns (AkiraNetwork.VirtualStorageLibrary.

| Type                                    | Description  |
|---|--|
| PatternMatch                            |  |
| (AkiraNetwork.VirtualStorageLibrary.    |  |
| VirtualItem                             | A  |
| VIRTUALCYCLEDETECTOR                    | VirtualItem<T> (AkiraNetwork.VirtualStorageLibrary.VirtualItem-1.html) |
| ITEMV1HTML                              | initialized with the specified data.                                   |
| <T>(AkiraNetwork.VirtualStorageLibrary. |  |

VirtualException

(AkiraNetwork.VirtualStorageLibrary.

## Implements

VirtualGroupCondition<T, TKey>

(AkiraNetwork.VirtualStorageLibrary.

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

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

VIRTUALID

(AkiraNetwork.VirtualStorageLibrary.

VirtualItem

(AkiraNetwork.VirtualStorageLibrary.

VirtualTextFormatter.GenerateSingleTableDebugText<T>(T)

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

VirtualNodeExtensions.ResolveNodeType(VirtualNode)

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeExtensions.html#AkiraNetwork\_VirtualStorageLibrary\_VirtualNodeExtensions\_ResolveNodeType\_AkiraNetwork\_VirtualStorageLibrary\_VirtualNode\_)

VIRTUALNODECONTEXT

(AkiraNetwork.VirtualStorageLibrary.

VirtualNodeExtensions

(AkiraNetwork.VirtualStorageLibrary.

VirtualNodeListConditions

(AkiraNetwork.VirtualStorageLibrary.

VirtualnodeName

(AkiraNetwork.VirtualStorageLibrary.

VirtualNodeNotFoundException

(AkiraNetwork.VirtualStorageLibrary.

VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary.

VirtualNodeTypeFilter

(AkiraNetwork.VirtualStorageLibrary.

VirtualPath

(AkiraNetwork.VirtualStorageLibrary.

# Class VirtualNode

Represents an abstract class for nodes.

Filter by title

## Inheritance

(AkiraNetwork.VirtualStorageLibrary)

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

↳ VirtualNode

↳ VirtualWildcardMatcher

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

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

↳ NotifyNodeDelegate

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

## Implements

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

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

↳ VirtualStorageLibrary

## Inherited Members

VirtualDirectory

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)))

VirtualException

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)))

VirtualGroupConditionType

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 ([<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))

VirtualStorageLibrary

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

VirtualItem<T>

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

## Syntax

[VirtualNode.html](#)

VirtualNode

↳ [AkiraNetwork.VirtualStorageLibrary](#).VirtualNode : IVirtualDeepCloneable<VirtualNode>

VirtualNodeContext

↳ [AkiraNetwork.VirtualStorageLibrary](#)

VirtualNodeExtensions

↳ [AkiraNetwork.VirtualStorageLibrary](#)

VirtualNodeListConditions

↳ [AkiraNetwork.VirtualStorageLibrary](#)

VirtualNodeName

↳ [AkiraNetwork.VirtualStorageLibrary](#)

VirtualNodeNotNotFoundException

VirtualNodeNotFoundException

**Declaration**

[AkiraNetwork.VirtualStorageLibrary](#)

VirtualNodeType

↳ [AkiraNetwork.VirtualStorageLibrary](#).VirtualNode

VirtualNodeTypeFilter

↳ [AkiraNetwork.VirtualStorageLibrary](#)

VirtualPath

↳ [AkiraNetwork.VirtualStorageLibrary](#)

**Parameters**

VirtualPath

↳ [AkiraNetwork.VirtualStorageLibrary](#)

## Constructors

### VirtualNode(VirtualNodeName)

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

VirtualNodeNotFoundException

VirtualNodeNotFoundException

**Declaration**

[AkiraNetwork.VirtualStorageLibrary](#)

VirtualNodeType

↳ [AkiraNetwork.VirtualStorageLibrary](#).VirtualNode

VirtualNodeTypeFilter

↳ [AkiraNetwork.VirtualStorageLibrary](#)

VirtualPath

↳ [AkiraNetwork.VirtualStorageLibrary](#)

| 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

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

### Parameters

PatternMatch

| Type   | Name        | Description               |
|--|-------------|---------------------------|
| (AkiraNetwork.VirtualStorageLibrary)   |             |                           |
| VirtualCycleDetector   |             |                           |
| VirtualNodeName  | name        | The name of node.         |
| (AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)  |             |                           |
| (AkiraNetwork.VirtualStorageLibrary)   |             |                           |
| DateTime ( <a href="https://learn.microsoft.com/dotnet/api/system.datetime">https://learn.microsoft.com/dotnet/api/system.datetime</a> ) | createdDate | The created date of node. |
| VirtualException   |             |                           |
| (AkiraNetwork.VirtualStorageLibrary)   |             |                           |
| VirtualGroupCondition<T, TKey>   |             |                           |
| (AkiraNetwork.VirtualStorageLibrary)   |             |                           |

## VirtualNode(VirtualNodeName, DateTime, DateTime)

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

### Declaration

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

### Parameters

VirtualNode

| Type   | Name        | Description               |
|--|-------------|---------------------------|
| (AkiraNetwork.VirtualStorageLibrary)   |             |                           |
| VirtualNodeContext   |             |                           |
| VirtualNodeName  | name        | The name of node.         |
| (AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)  |             |                           |
| (AkiraNetwork.VirtualStorageLibrary)   |             |                           |
| DateTime ( <a href="https://learn.microsoft.com/dotnet/api/system.datetime">https://learn.microsoft.com/dotnet/api/system.datetime</a> ) | createdDate | The created date of node. |
| VirtualNodeListConditions  |             |                           |
| (AkiraNetwork.VirtualStorageLibrary)   |             |                           |
| DateTime ( <a href="https://learn.microsoft.com/dotnet/api/system.datetime">https://learn.microsoft.com/dotnet/api/system.datetime</a> ) | updatedDate | The updated date of node. |
| (AkiraNetwork.VirtualStorageLibrary)   |             |                           |

## Properties

### CreatedDate

(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 |
|---|-------------|
| (AkiraNetwork.VirtualStorageLibrary.html)<br>DateTime ( <a href="https://learn.microsoft.com/dotnet/api/system.datetime">https://learn.microsoft.com/dotnet/api/system.datetime</a> )<br>VirtualWildcardMatcher |             |

(AkiraNetwork.VirtualStorageLibrary)

NotifyNodeDelegate

(AkiraNetwork.VirtualStorageLibrary)

PatternMatch

## IsReferencedInStorage

Gets (AkiraNetwork.VirtualStorageLibrary) if referenced in storage. If this property is true, the node is referenced from storage.

Otherwise VirtualCycleDetector

(AkiraNetwork.VirtualStorageLibrary)

VirtualDirectory

(AkiraNetwork.VirtualStorageLibrary)

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

(AkiraNetwork.VirtualStorageLibrary)

VirtualGroupCondition<T, TKey>

## Property Value

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

VirtualID

(AkiraNetwork.VirtualStorageLibrary)

VirtualItem

(AkiraNetwork.VirtualStorageLibrary)

## Name

VirtualItem<T>

(AkiraNetwork.VirtualStorageLibrary)

Gets the name of node.

1.html)

VirtualNode

(AkiraNetwork.VirtualStorageLibrary)

```
public string VirtualName { get; }
```

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeExtensions

## Property Value

| Type  | Description |
|---|-------------|
| VirtualNodeListConditions<br>(AkiraNetwork.VirtualStorageLibrary) |             |

VirtualNodeName

(AkiraNetwork.VirtualStorageLibrary.VIRTUALNAME.html)

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeNotFoundException

(AkiraNetwork.VirtualStorageLibrary)

## NodeType

VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeTypeFilter

## Declaration

(AkiraNetwork.VirtualStorageLibrary)

VirtualPath

(AkiraNetwork.VirtualStorageLibrary)



```
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. |

(AkiraNetwork.VirtualStorageLibrary)

## Returns

| Type                  | Description  |
|-----------------------|--|
| PatternMatch          |  |
| VirtualNode           | Cloned   |
| VirtualNodeDelegate   | VirtualNode  |
| .html                 | (AkiraNetwork.VirtualStorageLibrary.VirtualNode.html)                          |
| VirtualDirectory      | VirtualDirectory instance  |
| VirtualException      |  |
| VirtualStorageLibrary |  |
| VirtualGroup<T, TKey> |  |
| VirtualStorageLibrary | Updates the VirtualNode (AkiraNetwork.VirtualStorageLibrary.VirtualNode.html). |
| VirtualItem<T>        | 2.html)  |

## Update(VirtualNode)

Updates the VirtualNode (AkiraNetwork.VirtualStorageLibrary.VirtualNode.html).

## Declaration

(AkiraNetwork.VirtualStorageLibrary)

```
public abstract void Update(VirtualNode node)
```

(AkiraNetwork.VirtualStorageLibrary)

VirtualItem<T>

## Parameters

| Type        | Name | Description     |
|-------------|------|-----------------|
| VirtualNode | node | Value to update |

## Implementation

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeExtensions

(AkiraNetwork.VirtualStorageLibrary)

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

VirtualNodeListConditions

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeUtilities

(AkiraNetwork.VirtualStorageLibrary)

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

VirtualNodeExtensions.ResolveNodeType(VirtualNode)

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeExtensions.VirtualNodeExtensions.html#AkiraNetwork\_VirtualStorageLibrary\_VirtualNodeExtensions\_ResolveNodeType\_AkiraNetwork\_VirtualStorageLibrary\_VirtualNode\_(AkiraNetwork.VirtualStorageLibrary)

VirtualPath

(AkiraNetwork.VirtualStorageLibrary)



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

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

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

**Inherited Members**

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

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

**Assembly** [\(AkiraNetwork.VirtualStorageLibrary 2.html\)](#)

**Syntax**

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

public class VirtualNodeContext  
  VirtualItem

[\(AkiraNetwork.VirtualStorageLibrary\)](#)  
VirtualItem<T>

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

VirtualNode

**VirtualNodeContext(VirtualNode?, VirtualPath, VirtualDirectory?, int, int, VirtualPath?, bool, VirtualSymbolicLink?)**

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

**Declaration**

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

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

VirtualNodeType

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

VirtualNodeTypeFilter

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

VirtualPath

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

| Type   | Name                 | Description   |
|--|----------------------|---|
| VirtualNode<br>(AkiraNetwork.VirtualStorageLibrary.VirtualNode.html)   | <i>node</i>          | The node being traversed.   |
| VirtualPath<br>(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)<br>(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)  | <i>traversalPath</i> | The path used for node traversal.   |
| VirtualDirectory<br>VirtualWildcardMatcher<br>(AkiraNetwork.VirtualStorageLibrary.VirtualDirectory.html)<br>NotifyNodeDelegate<br>int (AkiraNetwork.VirtualStorageLibrary.VirtualDepth.html)<br>(https://learn.microsoft.com/dotnet/api/system.int32)<br>(AkiraNetwork.VirtualStorageLibrary.VirtualDepth.html)  | <i>parentNode</i>    | The parent directory of the node.   |
| int VirtualCycleDetector<br>int (AkiraNetwork.VirtualStorageLibrary.VirtualDepth.html)<br>VirtualDirectory<br>VirtualDepth<br>VirtualDepthException<br>(AkiraNetwork.VirtualStorageLibrary.VirtualDepth.html)<br>VirtualGroupCondition<T, TKey><br>bool (AkiraNetwork.VirtualStorageLibrary.VirtualDepth.html)<br>VirtualID<br>m.boolean<br>(AkiraNetwork.VirtualStorageLibrary.VirtualDepth.html) | <i>depth</i>         | The depth of the node. The root is considered 0.                                  |
| VirtualPathException<br>(AkiraNetwork.VirtualStorageLibrary.VirtualPathException.html)   | <i>index</i>         | The index of the node. Corresponds to the enumeration order within the directory. |
| VirtualPathException<br>(AkiraNetwork.VirtualStorageLibrary.VirtualPathException.html)<br>VirtualGroupCondition<T, TKey><br>bool (AkiraNetwork.VirtualStorageLibrary.VirtualDepth.html)<br>VirtualID<br>m.boolean<br>(AkiraNetwork.VirtualStorageLibrary.VirtualDepth.html)  | <i>resolvedPath</i>  | The result of resolving a symbolic link.  |
| VirtualSymbolicLink<br>(AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink.html)<br>VirtualSymbolicLink<T><br>VirtualSymbolicLink<br>(AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink.html)   | <i>resolved</i>      | Indicates whether the symbolic link has been resolved.                            |
| VirtualSymbolicLink<br>(AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink.html)<br>VirtualSymbolicLink<T><br>VirtualSymbolicLink<br>(AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink.html)   | <i>resolvedLink</i>  | The resolved symbolic link.   |

## Properties

### Depth

Gets `VirtualNode.Depth` and can only be set within the assembly. The root is considered 0.

(AkiraNetwork.VirtualStorageLibrary.VirtualDepth.html)

### Declaration

`VirtualNodeName`

(AkiraNetwork.VirtualStorageLibrary.VirtualDepth.html)

`VirtualNodeDepthException`

(AkiraNetwork.VirtualStorageLibrary.VirtualDepth.html)

`VirtualNodeType`

(AkiraNetwork.VirtualStorageLibrary.VirtualDepth.html)

### Type

`VirtualNodeTypeFilter`

(AkiraNetwork.VirtualStorageLibrary.VirtualDepth.html)

`int`

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

`VirtualPath`

(AkiraNetwork.VirtualStorageLibrary.VirtualDepth.html)

### Description

The depth of the node.

## 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  
1.html)
```

### Property Value

IVirtualWildcardMatcher

**Type** AkiraNetwork.VirtualStorageLibrary

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

PatternMatch

(AkiraNetwork.VirtualStorageLibrary

### Node

#### VirtualCycleDetector

(AkiraNetwork.VirtualStorageLibrary

Gets the node and can only be set within the assembly.

VirtualDirectory

(AkiraNetwork.VirtualStorageLibrary

VirtualException

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

(AkiraNetwork.VirtualStorageLibrary

### Property Value

VirtualID

**Type** AkiraNetwork.VirtualStorageLibrary

VirtualNode (AkiraNetwork.VirtualStorageLibrary.VirtualNode.html)

**Description**

The current node.

### ParentDirectory

1.html

VirtualNode

Gets the parent directory of the node and can only be set within the assembly.

(AkiraNetwork.VirtualStorageLibrary

### Declaration

VirtualNodeContext

(AkiraNetwork.VirtualStorageLibrary

VirtualNodeExtension? ParentDirectory { get; }

(AkiraNetwork.VirtualStorageLibrary

VirtualNodeListConditions

(AkiraNetwork.VirtualStorageLibrary

**Type** VirtualNodeName

(AkiraNetwork.VirtualStorageLibrary

VirtualDirectory (AkiraNetwork.VirtualStorageLibrary.VirtualDirectory.html)

**Description**

The instance of the parent directory.

VirtualNodeNotFoundException

(AkiraNetwork.VirtualStorageLibrary

VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary

### Resolved

(AkiraNetwork.VirtualStorageLibrary

VirtualNodeTypeFilter

Gets a value indicating whether the symbolic link has been resolved and can only be set within the assembly.

(AkiraNetwork.VirtualStorageLibrary

### Declaration

(AkiraNetwork.VirtualStorageLibrary



## Property Value

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

▼

## Methods

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

IVirtualWildcardMatcher

Tostring()

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

NotifyNodeDelegate

Returns a string representation of this instance's information.

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

Declaration

Match

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

protected override string ToString()

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

VirtualDirectory

Returns

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

Type

VirtualException

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

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

VirtualGroupCondition<T, TKey>

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

Description

A string representing this instance's information.

Overrides

2.html

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

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

## Extension Methods

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

VirtualTextFormatter.GenerateSingleTableDebugText<T>(T)

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) #AkiraNetwork\_VirtualStorageLibrary\_Utils\_VirtualTextFormatter\_GenerateSingleTableDebugText\_1\_\_0\_

VirtualNode

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

VirtualNodeContext

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

VirtualNodeExtensions

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

VirtualNodeListConditions

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

VirtualnodeName

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

VirtualNodeNotFoundException

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

VirtualNodeTypeFilter

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

VirtualPath

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

# Class VirtualNodeExtensions

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

Filter by title

## Inheritance

|   |
|---|
| (AkiraNetwork.VirtualStorageLibrary.html)   |
| ↳ object (https://learn.microsoft.com/dotnet/api/system.object)   |
| ↳ VirtualNodeExtensions   |
| (AkiraNetwork.VirtualStorageLibrary.html)   |
| NotifyNodeDelegate  |
| object.Equals(object) (https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals(system-object))                       |
| (AkiraNetwork.VirtualStorageLibrary.html)   |
| PatternMatch  |
| object.Equals(object, object) (https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals(system-object-system-object)) |
| (AkiraNetwork.VirtualStorageLibrary.html)   |
| VirtualCycleDetector  |
| object.GetHashCode() (https://learn.microsoft.com/dotnet/api/system.object.gethashcode)   |
| (AkiraNetwork.VirtualStorageLibrary.html)   |
| object.GetType() (https://learn.microsoft.com/dotnet/api/system.object.gettype)   |
| (AkiraNetwork.VirtualStorageLibrary.html)   |
| VirtualDirectory  |
| object.MemberwiseClone() (https://learn.microsoft.com/dotnet/api/system.object.memberwiseclone)   |
| (AkiraNetwork.VirtualStorageLibrary.html)   |
| object.ReferenceEquals(object, object) (https://learn.microsoft.com/dotnet/api/system.object.referenceequals)                                 |
| (AkiraNetwork.VirtualStorageLibrary.html)   |
| VirtualException  |
| object.ToString() (https://learn.microsoft.com/dotnet/api/system.object.tostring)   |
| (AkiraNetwork.VirtualStorageLibrary.html)   |

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

Assembly AkiraNetwork.VirtualStorageLibrary (AkiraNetwork.VirtualStorageLibrary.html)

## Syntax

|   |
|---|
| VirtualID                                 |
| (AkiraNetwork.VirtualStorageLibrary.html) |
| VirtualItem                               |
| (AkiraNetwork.VirtualStorageLibrary.html) |
| VirtualItem<T>                            |
| (AkiraNetwork.VirtualStorageLibrary.html) |

## Methods

VirtualNode

(AkiraNetwork.VirtualStorageLibrary.html)

### ResolveNodeType(VirtualNode)

(AkiraNetwork.VirtualStorageLibrary.html)

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

(VirtualNodeExtensions)

Otherwise, returns the type of the node itself.

(AkiraNetwork.VirtualStorageLibrary.html)

VirtualNodeListConditions

(AkiraNetwork.VirtualStorageLibrary.html)

VirtualNodeName

public static VirtualNodeType ResolveNodeType(this VirtualNode node)

(AkiraNetwork.VirtualStorageLibrary.html)

VirtualNodeNotFoundException

(AkiraNetwork.VirtualStorageLibrary.html)

Parameters

VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary.html)

VirtualNode

(AkiraNetwork.VirtualStorageLibrary.html)

VirtualPath

(AkiraNetwork.VirtualStorageLibrary.html)

| Type            | Name | Description                               |
|-----------------|------|---|
| VirtualNodeType | node | The virtual node to resolve the type for. |

## Returns

| Type   | Description   |
|--|---|
| VirtualNodeType<br>(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.VirtualNodeType.html)  
IVirtualWildcardMatcher  
(AkiraNetwork.VirtualStorageLibrary.VirtualWildcardMatcher.html)  
NotifyNodeDelegate  
(AkiraNetwork.VirtualStorageLibrary.NotifyNodeDelegate.html)  
PatternMatch  
(AkiraNetwork.VirtualStorageLibrary.PatternMatch.html)  
VirtualCycleDetector  
(AkiraNetwork.VirtualStorageLibrary.VirtualCycleDetector.html)  
VirtualDirectory  
(AkiraNetwork.VirtualStorageLibrary.VirtualDirectory.html)  
VirtualException  
(AkiraNetwork.VirtualStorageLibrary.VirtualException.html)  
VirtualGroupCondition<T, TKey>  
(AkiraNetwork.VirtualStorageLibrary.VirtualGroupCondition.html)  
VirtualID  
(AkiraNetwork.VirtualStorageLibrary.VirtualID.html)  
VirtualItem  
(AkiraNetwork.VirtualStorageLibrary.VirtualItem.html)  
VirtualItem<T>  
(AkiraNetwork.VirtualStorageLibrary.VirtualItemT.html)  
VirtualNode  
(AkiraNetwork.VirtualStorageLibrary.VirtualNode.html)  
VirtualNodeContext  
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeContext.html)  
VirtualNodeExtensions  
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeExtensions.html)  
VirtualNodeListConditions  
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeListConditions.html)  
VirtualnodeName  
(AkiraNetwork.VirtualStorageLibrary.VirtualnodeName.html)  
VirtualNodeNotFoundException  
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeNotFoundException.html)  
VirtualNodeType  
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeType.html)  
VirtualNodeTypeFilter  
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeTypeFilter.html)  
VirtualPath  
(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

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

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

Assembly: [VirtualStorageLibrary.dll](#)

## Syntax

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

## Constructors

VirtualNodeListConditions()  
VirtualID  
(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeListConditions()  
(AkiraNetwork.VirtualStorageLibrary)

Initializes a new instance of the VirtualNodeListConditions  
(AkiraNetwork.VirtualStorageLibrary).VirtualNodeListConditions.html) class with default conditions.

Declaration  
VirtualNode  
(AkiraNetwork.VirtualStorageLibrary)  
public VirtualNodeListConditions()  
VirtualNodeContext

(AkiraNetwork.VirtualStorageLibrary)  
VirtualNodeExtensions

(AkiraNetwork.VirtualStorageLibrary)  
VirtualNodeListConditions

VirtualNodeListConditions(VirtualNodeTypeFilter,  
VirtualGroupCondition<VirtualNode, object>?,  
List<VirtualSortCondition<VirtualNode>>?)  
VirtualNodeName

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

(AkiraNetwork.VirtualStorageLibrary)  
VirtualNodeType  
VirtualNodeListConditions(VirtualNodeTypeFilter filter, VirtualGroupCondition<Virtual

Node, object>? groupCondition, List<VirtualSortCondition<VirtualNode>>? sortConditions)

VirtualPath  
(AkiraNetwork.VirtualStorageLibrary)

## Parameters

| Type   | Name                  | Description                                   |
|--|-----------------------|---|
| VirtualNodeTypeFilter<br>(AkiraNetwork.VirtualStorageLibrary.VirtualNodeTypeFilter.html)   | <i>filter</i>         | The filter criteria for the virtual nodes.    |
| VirtualGroupCondition<br>(AkiraNetwork.VirtualStorageLibrary.VirtualGroupCondition-2.html)<br><VirtualNode (AkiraNetwork.VirtualStorageLibrary.VirtualNode.html), object (https://learn.microsoft.com/dotnet/api/system.object)>   | <i>groupCondition</i> | The grouping condition for the virtual nodes. |
| List IVirtualWildcardMatcher<br>(AkiraNetwork.VirtualStorageLibrary<br>(https://learn.microsoft.com/dotnet/api/system.collections.generic.list-1)<br>NotifyNodeDelegate<br>< (AkiraNetwork.VirtualStorageLibrary<br>PatternMatch<br>VirtualSortCondition<br>(AkiraNetwork.VirtualStorageLibrary<br>(AkiraNetwork.VirtualStorageLibrary.VirtualSortCondition-1.html)<br><VirtualNode (AkiraNetwork.VirtualStorageLibrary.VirtualNode.html)>><br>(AkiraNetwork.VirtualStorageLibrary<br>VirtualDirectory<br>(AkiraNetwork.VirtualStorageLibrary<br>VirtualException<br>(AkiraNetwork.VirtualStorageLibrary | <i>sortConditions</i> | The sorting conditions for the virtual nodes. |

## Properties

VirtualNodeTypeFilter  
(AkiraNetwork.VirtualStorageLibrary

VirtualGroupCondition<T, TKey>

(AkiraNetwork.VirtualStorageLibrary

### Filter

2.html)

VirtualID

Specifies the filter criteria, determining which types of virtual nodes to include.

(AkiraNetwork.VirtualStorageLibrary

### Declaration

VirtualItem  
(AkiraNetwork.VirtualStorageLibrary

VirtualItem<T> VirtualTypeFilter Filter { readonly get; set; }

(AkiraNetwork.VirtualStorageLibrary

1.html)

### Property Value

VirtualNode  
(AkiraNetwork.VirtualStorageLibrary

| Type                  | Description   |
|-----------------------|---|
| VirtualNodeTypeFilter | The filter specifying the types of virtual nodes. The default value is All. |

|  |  |
|--|--|
| VirtualNodeContext                                 |  |
| VirtualNodeTypeFilter                              | The filter specifying the types of virtual nodes. The default value is All.  |
| (AkiraNetwork.VirtualStorageLibrary                | (AkiraNetwork.VirtualStorageLibrary  |
| (AkiraNetwork.VirtualStorageLibrary                | All  |
| VirtualNodeExtensions                              |  |
| array<VirtualNodeTypeFilter> VirtualNodeTypeFilter | (AkiraNetwork.VirtualStorageLibrary.VirtualNodeTypeFilter.html#AkiraNetwork_VirtualStorageLibrary_VirtualNodeTypeFilter_All) |
| (AkiraNetwork.VirtualStorageLibrary                |  |
| VirtualNodeListConditions                          |  |
| (AkiraNetwork.VirtualStorageLibrary                |  |

## GroupCondition

VirtualNodeName  
(AkiraNetwork.VirtualStorageLibrary

VirtualNodeNotFoundException  
(AkiraNetwork.VirtualStorageLibrary

Specifies the grouping condition, determining how to group the virtual nodes.

(AkiraNetwork.VirtualStorageLibrary

### Declaration

VirtualNodeTypeFilter

(AkiraNetwork.VirtualStorageLibrary

public VirtualGroupCondition<VirtualNode, object>? GroupCondition { readonly get; set; }

(AkiraNetwork.VirtualStorageLibrary

## Property Value

| Type  | Description   |
|---|---|
| VirtualGroupCondition<br>(AkiraNetwork.VirtualStorageLibrary.VirtualGroupCondition-2.html)<br><<br>VirtualNode<br>(AkiraNetwork.VirtualStorageLibrary.VirtualNode.html)<br>, object ( <a href="https://learn.microsoft.com/dotnet/api/system.object">https://learn.microsoft.com/dotnet/api/system.object</a> )><br>IVirtualWildcardMatcher<br>(AkiraNetwork.VirtualStorageLibrary)<br>NotifyNodeDelegate<br>(AkiraNetwork.VirtualStorageLibrary) | The condition for grouping virtual nodes.<br>The default value is <code>null</code> . You can specify <code>null</code> if no grouping condition is needed. |
| SortConditions<br>(AkiraNetwork.VirtualStorageLibrary)<br>PatternMatch<br>Specifies the sorting conditions, determining the order in which the virtual nodes are arranged.<br>(AkiraNetwork.VirtualStorageLibrary)  |   |

## SortConditions

(AkiraNetwork.VirtualStorageLibrary)

PatternMatch

Specifies the sorting conditions, determining the order in which the virtual nodes are arranged.

(AkiraNetwork.VirtualStorageLibrary)

VirtualCycleDetector

(AkiraNetwork.VirtualStorageLibrary)

VirtualDirectory

```
public List<VirtualSortCondition<VirtualNode>>? SortConditions { readonly get; set; }
```

(AkiraNetwork.VirtualStorageLibrary)

VirtualException

## Property Value

(AkiraNetwork.VirtualStorageLibrary)

| Type   | Description   |
|--|---|
| VirtualGroupCondition<T, TKey><br>(AkiraNetwork.VirtualStorageLibrary)<br>List 2.html<br>( <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> )<br>list-1(AkiraNetwork.VirtualStorageLibrary)<br>< VirtualItem<br>VirtualSortCondition<br>(AkiraNetwork.VirtualStorageLibrary.VirtualSortCondition-1.html)<br>< (AkiraNetwork.VirtualStorageLibrary)<br>VirtualNode<br>(AkiraNetwork.VirtualStorageLibrary.VirtualNode.html)<br>>> (AkiraNetwork.VirtualStorageLibrary) | 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

(AkiraNetwork.VirtualStorageLibrary)

VirtualTextFormatter.GenerateSingleTableDebugText<T>(T)

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

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeNotFoundException

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeTypeFilter

(AkiraNetwork.VirtualStorageLibrary)

VirtualPath

(AkiraNetwork.VirtualStorageLibrary)

# Class VirtualNodeName

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

**Inheritance**  
[\(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html\)](#)

↳ object (<https://learn.microsoft.com/dotnet/api/system.object>)  
  ↳ VirtualWildcardMatcher  
    ↳ VirtualNodeName  
      ↳ [\(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html\)](#)

**Implements**

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

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

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

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

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

**Inherited Members**

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

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

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

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>)

VirtualID

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

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

**Assembly:** VirtualStorageLibrary.dll

VirtualItem

**Syntax** [\(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html\)](#)

VirtualItem<T>

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

VirtualNode

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

VirtualNodeContext

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

VirtualNodeExtensions

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

VirtualNodeListConditions

**Constructors** [\(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html\)](#)

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

class VirtualNodeName with the specified name.

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

**Declaration** [\(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html\)](#)

VirtualNodeNotFoundException

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

public VirtualNodeName(string name)

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

VirtualNodeTypeFilter

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

VirtualPath

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

**Parameters** [\(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html\)](#)

| Type   | Name | Description           |
|--|------|-----------------------|
| string ( <a href="https://learn.microsoft.com/dotnet/api/system.string">https://learn.microsoft.com/dotnet/api/system.string</a> ) | name | The name of the node. |

## Properties

|  |                                    |   |
|--|------------------------------------|---|
| (AkiraNetwork.VirtualStorageLibrary.html)  | IsRoot                             | VirtualWildcardMatcher                            |
| Indicates whether the node is the root node.   | NotifyNodeDelegate                 |   |
| (AkiraNetwork.VirtualStorageLibrary.html)  | Declaration                        | AkiraNetwork.VirtualStorageLibrary                |
|  | Type                               | PatternMatch                                      |
| (AkiraNetwork.VirtualStorageLibrary.html)  |                                    | VirtualCycleDetector                              |
| (AkiraNetwork.VirtualStorageLibrary.html)  | Property Value                     | VirtualDirectory                                  |
| Type   | AkiraNetwork.VirtualStorageLibrary |   |
| VirtualException   |                                    |   |
| bool ( <a href="https://learn.microsoft.com/dotnet/api/system.boolean">https://learn.microsoft.com/dotnet/api/system.boolean</a> ) |                                    | true if this is the root node; otherwise, false . |

## Name

|   |  |
|---|--|
| VirtualID                                 |  |
| Gets the name of the node.                |  |
| (AkiraNetwork.VirtualStorageLibrary.html) |  |
| VirtualItem                               |  |
| (AkiraNetwork.VirtualStorageLibrary.html) |  |
| VirtualItem<T>                            |  |
| public string Name { get; }               |  |
| (AkiraNetwork.VirtualStorageLibrary.html) |  |
| 1.html)                                   |  |
| VirtualNode                               |  |

## Property Value

| Type               | Description |
|--------------------|-------------|
| VirtualNodeContext |             |

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

The name of this node.

## Methods

### CompareTo(VirtualNodeName?)

|   |                 |
|---|-----------------|
| (AkiraNetwork.VirtualStorageLibrary.html) | VirtualNodeType |
| (AkiraNetwork.VirtualStorageLibrary.html) | VirtualNodeName |

Compares this instance with a specified VirtualNodeName

(AkiraNetwork.VirtualStorageLibrary.html).

VirtualNodeTypeFilter

(AkiraNetwork.VirtualStorageLibrary.html)

VirtualPath

(AkiraNetwork.VirtualStorageLibrary.html)

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

## Parameters

| Type  | Name  | Description  |
|---|-------|--|
| VirtualNodeName<br>(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)<br>IVirtualWildcardMatcher<br>(AkiraNetwork.VirtualStorageLibrary.NotifyNodeDelegate<br>(AkiraNetwork.VirtualStorageLibrary. | other | The VirtualNodeName<br>(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)<br>to compare with. |

## Returns

| Type   | Description  |
|--|--|
| PatternMatch<br>(AkiraNetwork.VirtualStorageLibrary.int VirtualCycleDetector<br>(http://learn.microsoft.com/dotnet/api/system.io.virtualdirectory<br>nt32)VirtualDirectory<br>(AkiraNetwork.VirtualStorageLibrary.VirtualException<br>(AkiraNetwork.VirtualStorageLibrary.VirtualGroupCondition<T, TKey><br>(AkiraNetwork.VirtualStorageLibrary.Compared this instance with a specified object.<br>2.html) | A value indicating the relative order of the instances being compared. |

## CompareTo(object?)

Compares this instance with a specified object.  
2.html)

## Declaration

```
VirtualItem CompareTo(object? obj)
```

VirtualItem<T>

## Parameters

| Type   | Name | Description                 |
|--|------|-----------------------------|
| VirtualNode<br>object (https://learn.microsoft.com/dotnet/api/system.object)<br>(AkiraNetwork.VirtualStorageLibrary. | obj  | The object to compare with. |

VirtualNodeContext

## Returns

| Type   | Description  |
|--|--|
| VirtualNodeExtensions<br>(AkiraNetwork.VirtualStorageLibrary.int VirtualNodeListConditions<br>(http://learn.microsoft.com/dotnet/api/system.io.virtualnode<br>nt32)VirtualnodeName<br>(AkiraNetwork.VirtualStorageLibrary. | A value indicating the relative order of the instances being compared. |

## Exceptions

| Type   | Condition  |
|--|--|
| VirtualNodeType<br>Arg<br>(AkiraNetwork.VirtualStorageLibrary.VirtualNodeType<br>(https://learn.microsoft.com/dotnet/api/system.argument<br>arg<br>(AkiraNetwork.VirtualStorageLibrary.VirtualPath<br>(AkiraNetwork.VirtualStorageLibrary. | Thrown when the object is not a VirtualNodeName<br>(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  
1.html)
```

### Parameters

|      |   |       |   |
|------|---|-------|---|
| Type | AkiraNetwork.VirtualStorageLibrary.IVirtualWildcardMatcher  | Name  | Description   |
|      | NotifyNodeDelegate<br>VirtualNodeName<br>(AkiraNetwork.VirtualStorageLibrary<br>(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)<br>PatternMatch<br>(AkiraNetwork.VirtualStorageLibrary<br>VirtualCycleDetector<br>(AkiraNetwork.VirtualStorageLibrary | other | The<br>VirtualNodeName<br>(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)<br>to compare with this instance. |

### Returns

VirtualDirectory

### Type

### Description

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

VirtualID

(AkiraNetwork.VirtualStorageLibrary

## Equals(object?)

(AkiraNetwork.VirtualStorageLibrary

Determines whether the specified object is equal to this instance.

### Declaration

```
1.html)  
VirtualNode  
public override bool Equals(object? obj)  
(AkiraNetwork.VirtualStorageLibrary  
VirtualNodeContext
```

### Parameters

|      |  |      |  |
|------|--|------|--|
| Type | AkiraNetwork.VirtualStorageLibrary.VirtualNodeExtensions         | Name | Description  |
|      | object<br>(https://learn.microsoft.com/dotnet/api/system.object) | obj  | The object to compare with this instance.<br>(AkiraNetwork.VirtualStorageLibrary |

### Returns

VirtualNodeName

(AkiraNetwork.VirtualStorageLibrary

### Type

### Description

|                              |      |   |                    |
|------------------------------|------|---|--------------------|
| VirtualNodeNotFoundException | bool | true if the specified object is equal to this instance;<br>(AkiraNetwork.VirtualStorageLibrary<br>VirtualNodeType<br>(https://learn.microsoft.com/dotnet/api/system.bo<br>(AkiraNetwork.VirtualStorageLibrary<br>olean) | otherwise, false . |
|------------------------------|------|---|--------------------|

VirtualNodeTypeFilter

(AkiraNetwork.VirtualStorageLibrary

### Overrides

VirtualPath

(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)))

## GenerateNodeName(string)

Generates a new node name using the specified prefix.

### Declaration

```
(AkiraNetwork.VirtualStorageLibrary  
1.html)  
public static VirtualNodeName GenerateNodeName(string prefix)  
VirtualWildcardMatcher
```

(AkiraNetwork.VirtualStorageLibrary)

### Parameters

(AkiraNetwork.VirtualStorageLibrary)

Type PatternMatch

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

Name prefix

Description The prefix for the node name.

### Returns

(AkiraNetwork.VirtualStorageLibrary)

VirtualDirectory

Type (AkiraNetwork.VirtualStorageLibrary)

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

(AkiraNetwork.VirtualStorageLibrary)

Type VirtualItem

Condition

(AkiraNetwork.VirtualStorageLibrary)

ArgumentException

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

Thrown when the prefix is an empty string.

## GetHashCode()

VirtualNodeContext

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeExtensions

### Declaration

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeListConditions

public int GetHashCode()

VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary)

Returns VirtualNodeNotFoundException

Type (AkiraNetwork.VirtualStorageLibrary)

Description

VirtualNodeType

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

A hash code for this instance.

### Overrides

(AkiraNetwork.VirtualStorageLibrary)

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

(AkiraNetwork.VirtualStorageLibrary)

## IsValidNodeName(VirtualnodeName)

Checks if a node name is valid.

### Declaration

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

**Parameters**  
1.html)

| Type  | Name     | Description             |
|---|----------|-------------------------|
| VirtualWildcardMatcher<br>(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html) | nodeName | The node name to check. |

PatternMatch  
**Returns**

| Type   | Description  |
|--|--|
| VirtualCycleDetector<br>(AkiraNetwork.VirtualStorageLibrary.VirtualDirectoryName.html) | true if the node name is valid; otherwise, false . |

VirtualException

(AkiraNetwork.VirtualStorageLibrary.VirtualDirectoryName.html)

## ResetCounter()

(AkiraNetwork.VirtualStorageLibrary.VirtualDirectoryName.html)

Resets the counter used for node name generation.

### Declaration

```
(AkiraNetwork.VirtualStorageLibrary.VirtualDirectoryName.html)
```

```
VirtualItem  
public static void ResetCounter()
```

(AkiraNetwork.VirtualStorageLibrary.VirtualDirectoryName.html)

## ToString()

VirtualNode

Returns the name of the node as a string.  
VirtualNodeContext

### Declaration

```
(AkiraNetwork.VirtualStorageLibrary.VirtualDirectoryName.html)
```

```
VirtualNodeExtensions  
public static string ToString()
```

(AkiraNetwork.VirtualStorageLibrary.VirtualDirectoryName.html)

**Returns** VirtualnodeName

| Type   | Description                       |
|--|-----------------------------------|
| AkiraNetwork.VirtualStorageLibrary.VirtualDirectoryName.html | The name of the node as a string. |

VirtualNodeType

### Overrides

```
(AkiraNetwork.VirtualStorageLibrary.VirtualDirectoryName.html)
```

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

(AkiraNetwork.VirtualStorageLibrary.VirtualDirectoryName.html)

## Operators

(AkiraNetwork.VirtualStorageLibrary.VirtualDirectoryName.html)

## **operator ==(VirtualnodeName?, VirtualnodeName?)**

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

### Declaration

```

public static bool operator ==(VirtualnodeName? left, VirtualnodeName? right)
(AkiraNetwork.VirtualStorageLibrary.VirtualnodeName.html)

```

### Parameters

| Type   | Name         | Description   |
|--|--------------|---|
| VirtualID<br>(AkiraNetwork.VirtualStorageLibrary.VirtualID.html) | <i>left</i>  | The left VirtualnodeName (AkiraNetwork.VirtualStorageLibrary.VirtualnodeName.html) to compare.  |
| VirtualID<br>(AkiraNetwork.VirtualStorageLibrary.VirtualID.html) | <i>right</i> | The right VirtualnodeName (AkiraNetwork.VirtualStorageLibrary.VirtualnodeName.html) to compare. |

### Returns

| Type  | Description   |
|---|---|
| bool<br>(AkiraNetwork.VirtualStorageLibrary.VirtualStorageLibrary.html) | true if the specified instances are equal; otherwise, false . |

## **implicit operator string(VirtualnodeName)**

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

### Declaration

```

public implicit operator string(VirtualnodeName nodeName)
(AkiraNetwork.VirtualStorageLibrary.VirtualStorageLibrary.html)

```

### Parameters

| Type  | Name            | Description   |
|---|-----------------|---|
| VirtualNodeNotFoundException<br>(AkiraNetwork.VirtualStorageLibrary.VirtualStorageLibrary.html) | <i>nodeName</i> | The VirtualnodeName (AkiraNetwork.VirtualStorageLibrary.VirtualnodeName.html) to convert. |

(AkiraNetwork.VirtualStorageLibrary.VirtualStorageLibrary.html)

## 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](#)).  
[T.html](#))

### Declaration

(AkiraNetwork.VirtualStorageLibrary)

```
public static implicit operator VirtualNodeName(string name)
```

(AkiraNetwork.VirtualStorageLibrary)

PatternMatch

### Parameters

(AkiraNetwork.VirtualStorageLibrary)

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

### Return Value

VirtualException

| Type                           | Description |
|--------------------------------|-------------|
| VirtualGroupCondition<T, TKey> |             |

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

[2.html](#))

VirtualID

(AkiraNetwork.VirtualStorageLibrary)

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

(AkiraNetwork.VirtualStorageLibrary)

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

(AkiraNetwork.VirtualStorageLibrary)

### Declaration

VirtualNode

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

VirtualNodeContext

(AkiraNetwork.VirtualStorageLibrary)

### Parameters

VirtualNodeExtensions

| Type   | Name         | Description   |
|--|--------------|---|
| AkiraNetwork.VirtualStorageLibrary.VirtualNodeListConditions<br>VirtualNodeName<br>(AkiraNetwork.VirtualStorageLibrary)<br>(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html) | <i>left</i>  | The left VirtualNodeName<br>(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)<br>to compare.  |
| VirtualNodeType<br>(AkiraNetwork.VirtualStorageLibrary)<br>VirtualNodeTypeFilter<br>(AkiraNetwork.VirtualStorageLibrary)<br>VirtualPath<br>(AkiraNetwork.VirtualStorageLibrary)      | <i>right</i> | The right VirtualNodeName<br>(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)<br>to compare. |

## Returns

| Type  | Description  |
|---|--|
| bool<br>( <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

(AkiraNetwork.VirtualStorageLibrary.html)  
VirtualWildcardMatcher  
(AkiraNetwork.VirtualStorageLibrary.html)  
IEquatable<T>  
(AkiraNetwork.VirtualStorageLibrary.html)  
IComparable<T>  
(AkiraNetwork.VirtualStorageLibrary.html)  
IComparable  
(AkiraNetwork.VirtualStorageLibrary.html)  
PatternMatch  
(AkiraNetwork.VirtualStorageLibrary.html)  
VirtualCycleDetector  
VirtualTextFormatter.GenerateSingleTableDebugText<T>(T)  
(AkiraNetwork.VirtualStorageLibrary.html#AkiraNetwork\_VirtualStorageLibrary\_Utils\_VirtualTextFormatter\_GenerateSingleTableDebugText\_1\_\_0\_)  
VirtualDirectory  
(AkiraNetwork.VirtualStorageLibrary.html)  
VirtualException  
(AkiraNetwork.VirtualStorageLibrary.html)  
VirtualGroupCondition<T, TKey>  
(AkiraNetwork.VirtualStorageLibrary.html)  
VirtualID  
(AkiraNetwork.VirtualStorageLibrary.html)  
VirtualItem  
(AkiraNetwork.VirtualStorageLibrary.html)  
VirtualItem<T>  
(AkiraNetwork.VirtualStorageLibrary.html)  
VirtualNode  
(AkiraNetwork.VirtualStorageLibrary.html)  
VirtualNodeContext  
(AkiraNetwork.VirtualStorageLibrary.html)  
VirtualNodeExtensions  
(AkiraNetwork.VirtualStorageLibrary.html)  
VirtualNodeListConditions  
(AkiraNetwork.VirtualStorageLibrary.html)  
VirtualnodeName  
(AkiraNetwork.VirtualStorageLibrary.html)  
VirtualNodeNotFoundException  
(AkiraNetwork.VirtualStorageLibrary.html)  
VirtualNodeType  
(AkiraNetwork.VirtualStorageLibrary.html)  
VirtualNodeTypeFilter  
(AkiraNetwork.VirtualStorageLibrary.html)  
VirtualPath  
(AkiraNetwork.VirtualStorageLibrary.html)

# Class VirtualNodeNotFoundException

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

Filter by title

## Inheritance

(AkiraNetwork.VirtualStorageLibrary)

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

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

↳ VirtualException (<https://learn.microsoft.com/dotnet/api/AkiraNetwork.VirtualStorageLibrary.VirtualException>)

↳ VirtualNodeNotFoundException

↳ NotifyNodeDelegate

## Implements

PatternMatch

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

(AkiraNetwork.VirtualStorageLibrary)

## Inherited Members

VirtualCycleDetector

Exception (BaseException) (<https://learn.microsoft.com/dotnet/api/system.exception.getbaseexception>)

VirtualException (Gettype) (<https://learn.microsoft.com/dotnet/api/system.exception.gettype>)

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

VirtualException (Data) (<https://learn.microsoft.com/dotnet/api/system.exception.data>)

(AkiraNetwork.VirtualStorageLibrary)

Exception (HelpLink) (<https://learn.microsoft.com/dotnet/api/system.exception.helplink>)

Exception (HRESULT) (<https://learn.microsoft.com/dotnet/api/system.exception.hresult>)

Exception (InnerException) (<https://learn.microsoft.com/dotnet/api/system.exception.innerexception>)

VirtualException (Message) (<https://learn.microsoft.com/dotnet/api/system.exception.message>)

Exception (Source) (<https://learn.microsoft.com/dotnet/api/system.exception.source>)

Exception (StackTrace) (<https://learn.microsoft.com/dotnet/api/system.exception.stacktrace>)

VirtualException (TargetException) (<https://learn.microsoft.com/dotnet/api/system.exception.targetsite>)

Exception (ObjectState) (<https://learn.microsoft.com/dotnet/api/system.exception.serializeobjectstate>)

object (Equals) ([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) ([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)))

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

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

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

VirtualNodeExtensions

**Namespace:** AkiraNetwork (<https://learn.microsoft.com/html/VirtualStorageLibrary>) (<https://learn.microsoft.com/html/AkiraNetwork.VirtualStorageLibrary.html>)

**Assembly:** VirtualStorageLibrary.dll

VirtualNodeListConditions

## Syntax

(AkiraNetwork.VirtualStorageLibrary)

    VirtualNodeName

    p(AkiraNetwork.VirtualStorageLibrary) : VirtualException, ISerializable

    VirtualNodeNotFoundException

(AkiraNetwork.VirtualStorageLibrary)

    VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary)

    VirtualNodeTypeFilter

(AkiraNetwork.VirtualStorageLibrary)

    VirtualPath

(AkiraNetwork.VirtualStorageLibrary)

## Constructors

### VirtualNodeNotFoundException()

(AkiraNetwork.VirtualStorageLibrary)

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

## Declaration

```
public VirtualNodeNotFoundException()  
    ▾
```

(AkiraNetwork.VirtualStorageLibrary)

## VirtualNodeNotFoundException(string)

IVirtualWildcardMatcher

Initializes a new instance of the VirtualNodeNotFoundException

(AkiraNetwork.VirtualStorageLibrary)

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

    NotInNodeDelegate

(AkiraNetwork.VirtualStorageLibrary)

    Declaration

        PatternMatch

(AkiraNetwork.VirtualStorageLibrary)

```
public VirtualNodeNotFoundException(string message)  
    VirtualCycleDetector
```

(AkiraNetwork.VirtualStorageLibrary)

    Parameters

(AkiraNetwork.VirtualStorageLibrary)

Type

VirtualException

string

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

Name

Description

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.

    VirtualItem<T>

    Declaration

(AkiraNetwork.VirtualStorageLibrary)

    1.html)

```
public VirtualNodeNotFoundException(string message, Exception innerException)  
    VirtualNodeContext
```

(AkiraNetwork.VirtualStorageLibrary)

    Parameters

(AkiraNetwork.VirtualStorageLibrary)

Type

VirtualNodeExtensions

(AkiraNetwork.VirtualStorageLibrary)

string

VirtualNodeListConditions

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

(AkiraNetwork.VirtualStorageLibrary)

Exception

(AkiraNetwork.VirtualStorageLibrary)

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

VirtualNodeNotFoundException

(AkiraNetwork.VirtualStorageLibrary)

    VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary)

    VirtualNodeTypeFilter

ISerializable

(AkiraNetwork.VirtualStorageLibrary)(https://learn.microsoft.com/dotnet/api/system.runtime.serialization.iserializable)

    VirtualPath

(AkiraNetwork.VirtualStorageLibrary)

## Implements

# Extension Methods

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

▼

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

# Enum VirtualNodeType

Specifies the types of virtual nodes.

Filter by title

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

**Assembly:** VirtualStorageLibrary.dll  
1.html)

**Syntax**

```
VirtualWildcardMatcher
(AkiraNetwork.VirtualStorageLibrary)
public enum VirtualNodeType
(AkiraNetwork.VirtualStorageLibrary)
PatternMatch
(AkiraNetwork.VirtualStorageLibrary)
VirtualCycleDetector
(AkiraNetwork.VirtualStorageLibrary)
```

## Fields

| Name   | Description   |
|--|---|
| VirtualDirectory<br>(AkiraNetwork.VirtualStorageLibrary)               |   |
| VirtualException<br>(AkiraNetwork.VirtualStorageLibrary)               | Represents a directory node.                        |
| Item<br>(AkiraNetwork.VirtualStorageLibrary)                           | Represents an item node with a user-defined type T. |
| VirtualGroupCondition<T, TKey><br>(AkiraNetwork.VirtualStorageLibrary) |   |
| None<br>(AkiraNetwork.VirtualStorageLibrary)                           | Represents an unspecified or unknown node type.     |
| SymbolicLink<br>VirtualID<br>(AkiraNetwork.VirtualStorageLibrary)      | Represents a symbolic link node.                    |

## Extension Methods

```
VirtualTextFormatter.GenerateSingleTableDebugText<VirtualNodeType>(VirtualNodeType)
(AkiraNetwork.VirtualStorageLibrary)
(AkiraNetwork.VirtualStorageLibrary.Utilities.VirtualTextFormatter.html#AkiraNetwork_VirtualStorageLibrary_Utilitie
1.html)
s_VirtualTextFormatter_GenerateSingleTableDebugText_1__0_
VirtualNode
(AkiraNetwork.VirtualStorageLibrary)
VirtualItem
VirtualItem<T>
VirtualItemContext
VirtualNodeContext
VirtualNodeExtensions
VirtualNodeExtensions
VirtualNodeListConditions
VirtualNodeListConditions
VirtualnodeName
VirtualnodeName
VirtualNodeNotFoundException
VirtualNodeNotFoundException
VirtualNodeType
VirtualNodeType
VirtualNodeTypeFilter
VirtualNodeTypeFilter
VirtualPath
VirtualPath
```

# Enum VirtualNodeTypeFilter

Specifies the filters for virtual node types.

▼ Filter by title

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

**Assembly:** VirtualStorageLibrary.dll  
1.html

**Syntax**

```
VirtualWildcardMatcher
(AkiraNetwork.VirtualStorageLibrary)
public enum VirtualNodeTypeFilter
{
    NoTypeNodeDelegate
(AkiraNetwork.VirtualStorageLibrary)
    PatternMatch
(AkiraNetwork.VirtualStorageLibrary)
    VirtualCycleDetector
(AkiraNetwork.VirtualStorageLibrary)
```

## Fields

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

## Extension Methods

|  |
|--|
| VirtualTextFormatter.GenerateSingleTableDebugText<VirtualNodeTypeFilter>(VirtualNodeTypeFilter)  |
| (AkiraNetwork.VirtualStorageLibrary.Utilities.VirtualTextFormatter.html#AkiraNetwork_VirtualStorageLibrary_Utils<br>s_VirtualTextFormatter_GenerateSingleTableDebugText_1__0_) |
| VirtualNodeContext   |
| (AkiraNetwork.VirtualStorageLibrary)   |
| VirtualNodeExtensions  |
| (AkiraNetwork.VirtualStorageLibrary)   |
| VirtualNodeListConditions  |
| (AkiraNetwork.VirtualStorageLibrary)   |
| VirtualnodeName  |
| (AkiraNetwork.VirtualStorageLibrary)   |
| VirtualNodeNotFoundException   |
| (AkiraNetwork.VirtualStorageLibrary)   |
| VirtualNodeType  |
| (AkiraNetwork.VirtualStorageLibrary)   |
| VirtualNodeTypeFilter  |
| (AkiraNetwork.VirtualStorageLibrary)   |
| VirtualPath  |
| (AkiraNetwork.VirtualStorageLibrary)   |

# Class VirtualPath

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

## Inheritance

1.html

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

## Implements

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

## Inherited Members

(AkiraNetwork.VirtualStorageLibrary)

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

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

VirtualID

**Namespace:** AkiraNetwork ([AkiraNetwork.html](https://learn.microsoft.com/dotnet/api/AkiraNetwork.html)).VirtualStorageLibrary ([AkiraNetwork.VirtualStorageLibrary.html](https://learn.microsoft.com/dotnet/api/AkiraNetwork.VirtualStorageLibrary.html))

**Assembly:** VirtualStorageLibrary.dll

VirtualItem

(AkiraNetwork.VirtualStorageLibrary)

## Syntax

VirtualPath  
VirtualItem<T>  
p(AkiraNetwork.VirtualStorageLibrary.IEnumerable<VirtualPath>, IComparable<VirtualPath>, IComparable 1.html)

VirtualNode

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeContext

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeExtensions

**VirtualPath(IEnumerable<VirtualNodeName>)**

(AkiraNetwork.VirtualStorageLibrary)

Initializes a new instance of the VirtualPath ([AkiraNetwork.VirtualStorageLibrary.VirtualPath.html](https://learn.microsoft.com/dotnet/api/AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)) class with the specified collection of path parts.

VirtualNodeName

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeNotFoundException

p(AkiraNetwork.VirtualStorageLibrary.IEnumerable<VirtualNodeName> parts)

VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary)

## Parameters

VirtualNodeTypeFilter

(AkiraNetwork.VirtualStorageLibrary)

VirtualPath

(AkiraNetwork.VirtualStorageLibrary)

| Type  | Name  | Description                   |
|---|-------|-------------------------------|
| IEnumerable<br>( <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> )<br><VirtualNodeName (AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)> | parts | The collection of path parts. |

▼

## VirtualPath(string)

Initializes a new instance of the VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) class with the specified path string.  
 VirtualWildcardMatcher  
 (AkiraNetwork.VirtualStorageLibrary)

### Declaration

```
(AkiraNetwork.VirtualStorageLibrary)
public VirtualPath(string path)
(AkiraNetwork.VirtualStorageLibrary)
  VirtualCycleDetector
```

### Parameters

| Type   | Name | Description |
|--|------|-------------|
| VirtualDirectory<br>(AkiraNetwork.VirtualStorageLibrary) |      |             |

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

## Properties

### VirtualID

(AkiraNetwork.VirtualStorageLibrary)

### Depth

(AkiraNetwork.VirtualStorageLibrary)

Gets the depth of the path.

### VirtualItem

(AkiraNetwork.VirtualStorageLibrary)

### Declaration

```
1.html)
  VirtualNode
public int Depth { get; }
(AkiraNetwork.VirtualStorageLibrary)
  VirtualNodeContext
```

### PropertyValue

(AkiraNetwork.VirtualStorageLibrary)

### VirtualNodeExtensions

(AkiraNetwork.VirtualStorageLibrary)

### int (VirtualNodeListConditions)

(AkiraNetwork.VirtualStorageLibrary)

### VirtualNodeName

(AkiraNetwork.VirtualStorageLibrary)

### VirtualNodeNotFoundException

(AkiraNetwork.VirtualStorageLibrary)

Gets the directory part of the path.

### VirtualNodeType

### Declaration

```
AkiraNetwork.VirtualStorageLibrary
  VirtualNodeTypeFilter
public VirtualNodeTypeFilter VirtualType { get; }
  VirtualPath
(AkiraNetwork.VirtualStorageLibrary)
```

### VirtualPath

(AkiraNetwork.VirtualStorageLibrary)

| Type                            | Description                                    |
|---------------------------------|--|
| int (VirtualNodeListConditions) | An integer representing the depth of the path. |

## Property Value

| Type   | Description  |
|--|--|
| VirtualPath<br>(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) | A VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) representing the directory part of the path. |

(AkiraNetwork.VirtualStorageLibrary.Dot1.html)

IVirtualWildcardMatcher  
Gets the dot (".") symbol.  
(AkiraNetwork.VirtualStorageLibrary)

Declaration

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

VirtualCycleDetector

Property Value

| Type   | Description                           |
|--|---------------------------------------|
| VirtualDirectory<br>(AkiraNetwork.VirtualStorageLibrary) | A string representing the dot symbol. |

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

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

DotDot  
Gets VirtualDoubleDot ("..") symbol.  
(AkiraNetwork.VirtualStorageLibrary)

Declaration

VirtualItem  
(AkiraNetwork.VirtualStorageLibrary)  
public static string DotDot { get; }  
(AkiraNetwork.VirtualStorageLibrary)

1.html)

Property Value

| Type  | Description                                  |
|---|--|
| VirtualNode<br>(AkiraNetwork.VirtualStorageLibrary) | A string representing the double dot symbol. |

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

VirtualNodeExtensions  
(AkiraNetwork.VirtualStorageLibrary)

FixedDepth  
Gets the depth of the fixed path.  
VirtualNodeName  
(AkiraNetwork.VirtualStorageLibrary)

Declaration

VirtualNodeNotFoundException  
(AkiraNetwork.VirtualStorageLibrary)  
public int FixedDepth { get; }  
VirtualNodeType  
(AkiraNetwork.VirtualStorageLibrary)

Property Value

TypeFilter  
(AkiraNetwork.VirtualStorageLibrary)  
VirtualPath  
(AkiraNetwork.VirtualStorageLibrary)

| 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 depth of the fixed path. |

## Remarks

The fixed depth indicates the number of parts in the path, excluding those with wildcards.

## FixedPath

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

Gets the fixed part of the path applying wildcard matching.

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

### Declaration

NotifyNodeDelegate

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

```
public VirtualPath FixedPath { get; }
```

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

VirtualCycleDetector

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

## Property Value

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

### Type

VirtualDirectory

### Type

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

VirtualException

### Type

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

VirtualGroupCondition<T, TKey>

### Type

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

## Remarks

(2.html)

The fixed path excludes parts containing wildcards.

VirtualID

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

VirtualItem

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

## IsAbsolute

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

### Value

VirtualItem

### Declaration

(1.html)

VirtualNode

### Type

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

VirtualNodeContext

### Type

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

## Property Value

VirtualNodeExtensions

### Type

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

VirtualNodeListConditions

### Type

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

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

### Description

true if the instance's path is absolute; otherwise, false .

## IsDot

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

VirtualNodeType

### Value

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

### Declaration

VirtualNodeTypeFilter

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

VirtualPath

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

```
public bool IsDot { get; }
```

## Property Value

| Type  | Description  |
|---|--|
| bool<br>( <a href="https://learn.microsoft.com/dotnet/api/system.boolean">https://learn.microsoft.com/dotnet/api/system.boolean</a> )<br>1.html)<br>IVirtualWildcardMatcher<br>(AkiraNetwork.VirtualStorageLibrary<br>NotifyNodeDelegate<br>(AkiraNetwork.VirtualStorageLibrary<br>Gets a value indicating whether the instance's path is a double dot.<br>PatternMatch<br>Declaration<br>(AkiraNetwork.VirtualStorageLibrary<br>VirtualCycleDetector<br>(AkiraNetwork.VirtualStorageLibrary<br>VirtualDirectory<br>(AkiraNetwork.VirtualStorageLibrary<br>Property Value<br>VirtualException | true if the instance's path is a dot; otherwise, false . |

## IsDotDot

Gets a value indicating whether the instance's path is a double dot.

PatternMatch

Declaration  
(AkiraNetwork.VirtualStorageLibrary  
VirtualCycleDetector  
(AkiraNetwork.VirtualStorageLibrary  
VirtualDirectory  
(AkiraNetwork.VirtualStorageLibrary  
Property Value  
VirtualException

| Type  | Description   |
|---|---|
| AkiraNetwork.VirtualStorageLibrary<br>VirtualGroupCondition<T, TKey><br>bool<br>(AkiraNetwork.VirtualStorageLibrary<br>( <a href="https://learn.microsoft.com/dotnet/api/system.boolean">https://learn.microsoft.com/dotnet/api/system.boolean</a> )<br>n)<br>VirtualID<br>(AkiraNetwork.VirtualStorageLibrary<br>VirtualItem<br>(AkiraNetwork.VirtualStorageLibrary<br>IsEmpty<br>VirtualItem<T><br>Gets a value indicating whether the instance's path is empty.<br>1.html)<br>Declaration<br>VirtualNode<br>(AkiraNetwork.VirtualStorageLibrary<br>public bool IsEmpty { get; }<br>VirtualNodeContext<br>(AkiraNetwork.VirtualStorageLibrary<br>VirtualNode<br>VirtualNodeExtensions<br>Type<br>VirtualNodeListConditions<br>bool<br>(AkiraNetwork.VirtualStorageLibrary<br>( <a href="https://learn.microsoft.com/dotnet/api/system.boolean">https://learn.microsoft.com/dotnet/api/system.boolean</a> )<br>(AkiraNetwork.VirtualStorageLibrary<br>VirtualNodeNotFoundException<br>(AkiraNetwork.VirtualStorageLibrary<br>VirtualNodeType<br>VirtualNodeTypeFilter<br>Gets a value indicating whether the instance's path ends with a slash.<br>VirtualNodeTypeFilter<br>Declaration<br>(AkiraNetwork.VirtualStorageLibrary<br>VirtualPath<br>(AkiraNetwork.VirtualStorageLibrary | true if the instance's path is a double dot; otherwise, false . |

## IsEmpty

Gets a value indicating whether the instance's path is empty.

1.html)  
Declaration  
VirtualNode

(AkiraNetwork.VirtualStorageLibrary  
VirtualNodeContext  
(AkiraNetwork.VirtualStorageLibrary  
VirtualNode  
VirtualNodeExtensions  
Type  
VirtualNodeListConditions  
bool  
(AkiraNetwork.VirtualStorageLibrary  
(<https://learn.microsoft.com/dotnet/api/system.boolean>)  
(AkiraNetwork.VirtualStorageLibrary  
VirtualNodeNotFoundException  
(AkiraNetwork.VirtualStorageLibrary  
VirtualNodeType  
VirtualNodeTypeFilter  
Gets a value indicating whether the instance's path ends with a slash.  
VirtualNodeTypeFilter  
Declaration  
(AkiraNetwork.VirtualStorageLibrary  
VirtualPath  
(AkiraNetwork.VirtualStorageLibrary

| Type   | Description  |
|--|--|
| AkiraNetwork.VirtualStorageLibrary<br>VirtualNodeListConditions<br>bool<br>(AkiraNetwork.VirtualStorageLibrary<br>( <a href="https://learn.microsoft.com/dotnet/api/system.boolean">https://learn.microsoft.com/dotnet/api/system.boolean</a> )<br>(AkiraNetwork.VirtualStorageLibrary<br>VirtualNodeNotFoundException<br>(AkiraNetwork.VirtualStorageLibrary<br>VirtualNodeType<br>VirtualNodeTypeFilter<br>Gets a value indicating whether the instance's path ends with a slash.<br>VirtualNodeTypeFilter<br>Declaration<br>(AkiraNetwork.VirtualStorageLibrary<br>VirtualPath<br>(AkiraNetwork.VirtualStorageLibrary | true if the instance's path is empty; otherwise, false . |

## IsEndsWithSlash

Gets a value indicating whether the instance's path ends with a slash.

VirtualNodeTypeFilter

Declaration  
(AkiraNetwork.VirtualStorageLibrary  
VirtualPath  
(AkiraNetwork.VirtualStorageLibrary

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

## Property Value

| Type  | Description   |
|---|---|
| bool<br>( <a href="https://learn.microsoft.com/dotnet/api/system.boolean">https://learn.microsoft.com/dotnet/api/system.boolean</a> ) 1.html) | true if the instance's path ends with a slash; otherwise, false . |

IVirtualWildcardMatcher  
(AkiraNetwork.VirtualStorageLibrary)

NotifyNodeDelegate  
(AkiraNetwork.VirtualStorageLibrary)

**IsRoot**  
Gets a value indicating whether the instance's path is the root path.  
(AkiraNetwork.VirtualStorageLibrary)

Declaration  
VirtualCycleDetector  
(AkiraNetwork.VirtualStorageLibrary)

**VirtualDirectory**  
IsRoot { get; }  
(AkiraNetwork.VirtualStorageLibrary)

VirtualException  
**Property Value**  
(AkiraNetwork.VirtualStorageLibrary)

| Type   | Description  |
|--|--|
| VirtualGroupCondition<T, TKey><br>(AkiraNetwork.VirtualStorageLibrary) | true if the instance's path is the root path; otherwise, false . |

VirtualItem  
(AkiraNetwork.VirtualStorageLibrary)

**NodeName**  
Value<T>  
(AkiraNetwork.VirtualStorageLibrary)

Gets the node name of the path.  
1.html)

VirtualNode  
Declaration  
(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeContext  
VirtualNodeName NodeName { get; }  
(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeExtensions  
(AkiraNetwork.VirtualStorageLibrary)

**Property Value**  
VirtualNodeListConditions  
(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeName  
(AkiraNetwork.VirtualStorageLibrary)

INodeNotFoundException  
(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeType  
(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeTypeFilter  
(AkiraNetwork.VirtualStorageLibrary)

**PartsList**  
VirtualPath  
Gets the list of parts that make up the path.  
(AkiraNetwork.VirtualStorageLibrary)

VirtualPath  
(AkiraNetwork.VirtualStorageLibrary)

| Type  | Description  |
|---|--|
| VirtualNodeName<br>(AkiraNetwork.VirtualStorageLibrary) | A<br>VirtualNodeName<br>(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)<br>representing the node name of the path. |



Gets the path separator character.

## Declaration

```
public static char Separator { get; }
```

## Property Value

| Type   | Description                                  |
|--|--|
| AkiraNetwork.VirtualStorageLibrary.<br>1.html)<br>char ( <a href="https://learn.microsoft.com/dotnet/api/system.char">https://learn.microsoft.com/dotnet/api/system.char</a> )<br>VirtualWildcardMatcher | A character representing the path separator. |

# Methods

(AkiraNetwork.VirtualStorageLibrary  
VirtualCycleDetector

### AddEndSlash()

**VirtualDirectory**  
Adds a trailing slash to the path if it doesn't already have one.

VirtualE

```
VirtualException  
Declaration  
(AkiraNetwork.VirtualStorageLibrary)  
  
VirtualGroupCondition<T, TKey>  
public VirtualPath AddEndSlash()  
(AkiraNetwork.VirtualStorageLibrary)  
2.html)
```

**ReturnVirtualID**

| Type   | Description  |
|--|--|
| (AkiraNetwork.VirtualStorageLibrary.VirtualItem)             |  |
| VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath) | A new VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) with a trailing slash. |
| (AkiraNetwork.VirtualStorageLibrary.VirtualPath)             |  |

**AddStartSlash()** (AkiraNetwork.Vi

## Addstart (slash) VirtualNodeContext

Adds a leading slash to the path if it doesn't already have one.

## **Virtual Declaration**

```
VirtualNodeListConditions  
p(AkiraNetworkVirtualStorageLibrary)ash()
```

VirtualM

**Returns** VirtualNodeNotFoundException

Virt

Type AkiraNetwork.VirtualStorageLibrary

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

### **ArePathsSubdirectories(VirtualPath, VirtualPath)**

Determines whether one path is a subdirectory of another path.

## Declaration

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

| <b>Parameters</b> | (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)                         |              |   |
|-------------------|---|--------------|---|
| <b>Type</b>       | VIRTUALWildcardMatcher  | <b>Name</b>  | <b>Description</b>  |
|                   | (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)                         | <i>path1</i> | The first VirtualPath<br>(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)  |
|                   | NotifyNodeDelegate<br>(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)   |              | PatternMatch<br>.   |
|                   | (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)                         | <i>path2</i> | The second VirtualPath<br>(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) |
|                   | VirtualCycleDetector<br>(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) |              | VirtualDirectory<br>.   |
|                   | (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)                         |              |   |

Retuřák - Akira Network Virtual Storage Library

| Type   | Description   |
|--|---|
| VirtualGroupCondition<T, TKey><br>(AkiraNetwork.VirtualStorageLibrary) | true if one path is a subdirectory of the other;<br>otherwise, false. |

`(AkiraNetwork.VirtualStorageLibrary.Exceptions)`

| Exceptions  | Type   | Condition                       |
|---|--|---------------------------------|
| <code>VirtualItem&lt;T&gt;</code>   | ( <a href="#">AkiraNetwork.VirtualStorageLibrary</a> ) |                                 |
| <code>ArgumentException</code><br>( <a href="https://learn.microsoft.com/dotnet/api/system.argumentexception">https://learn.microsoft.com/dotnet/api/system.argumentexception</a> ) |  | Thrown if either path is empty. |

**Combine(params VirtualPath[1])**

(AkiraNetwork.VirtualStorageLibrary)  
Combines the instance's path with the specified paths.  
VirtualNodeListConditions

### Declaration

VirtualNodeName

```
(AkiraNetwork.VirtualStorageLibrary)
public VirtualPath Combine(params VirtualPath[] paths)
    VirtualNodeNotFoundException
```

(AkiraNetwork VirtualStorage) library

## Parameters

## VirtGAN Node Type

## **VirtualNodeTypeFilter**

## virtualNode (AkiraNetwork)

(AkriaNetwork.VirtualStorageLibrary)  
VirtualPath

| Type   | Name  | Description   |
|--|-------|---|
| VirtualPath<br>(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)<br>[] | paths | An array of VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) objects to combine. |

#### Returns

(AkiraNetwork.VirtualStorageLibrary.

#### Type

.html)

#### Description

IVirtualWildcardMatcher  
VirtualPath  
(AkiraNetwork.VirtualStorageLibrary.  
(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)  
NotifyNodeDelegate  
alPath.html)  
(AkiraNetwork.VirtualStorageLibrary.

A new VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) representing the combined path.

PatternMatch

(AkiraNetwork.VirtualStorageLibrary.

## Combine(params string[])

(AkiraNetwork.VirtualStorageLibrary.

Combines the specified paths into a single path.

VirtualDirectory  
(AkiraNetwork.VirtualStorageLibrary.

VirtualException

(AkiraNetwork.VirtualStorageLibrary.  
public static string Combine(params string[] paths)  
VirtualGroupCondition<T, TKey>

(AkiraNetwork.VirtualStorageLibrary.

#### Parameters

| Type  | Name  | Description                          |
|---|-------|--------------------------------------|
| VirtualID<br>(AkiraNetwork.VirtualStorageLibrary. | paths | An array of path strings to combine. |

#### Returns

VirtualItem<T>  
(AkiraNetwork.VirtualStorageLibrary.

#### Type

.html)

#### Description

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

A combined path string.

## CombineFromIndex(VirtualPath, int)

(AkiraNetwork.VirtualStorageLibrary.

Combines the instance's path with the parts of another path starting from a specified index.

VirtualNodeListConditions

Declaration  
(AkiraNetwork.VirtualStorageLibrary.

VirtualNodeName

(AkiraNetwork.VirtualStorageLibrary.  
public VirtualPath CombineFromIndex(VirtualPath path, int index)

VirtualNodeNotFoundException

(AkiraNetwork.VirtualStorageLibrary.

#### Parameters

VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary.

VirtualNodeTypeFilter

(AkiraNetwork.VirtualStorageLibrary.

VirtualPath

(AkiraNetwork.VirtualStorageLibrary.

| Type   | Name         | Description   |
|--|--------------|---|
| VirtualPath<br>(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)                                 | <i>path</i>  | The VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) whose parts to combine. |
| int<br>(https://learn.microsoft.com/dotnet/api/AkiraNetwork.VirtualStorageLibrary.system.int32.html) | <i>index</i> | The starting index from which to combine the parts.   |

| Type  | Description   |
|---|---|
| NotifyNodeDelegate<br>(AkiraNetwork.VirtualStorageLibrary)                    |   |
| VirtualPathMatch<br>(AkiraNetwork.VirtualStorageLibrary.Virtu<br>alPathMatch) | A new<br>VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)<br>representing the combined path. |
| VirtualCycleDetector  |   |

# CompareTo(VirtualPath?)

Compares the instance's path with another VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html).  
VirtualGroupCondition<T, TKey>

## Declaration

```
    pVirtualPath CompareTo(VirtualPath? other)
```

(AkiraNetwork.VirtualStorageLibrary  
VirtualItem

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

| Type   | Name | Description  |
|--|------|--|
| (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)            |      | The other VirtualPath  |
| (AkiraNetwork.VirtualStorageLibrary.VirtualNodeVirtualPath.html) |      | (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) to compare with. |
| (AkiraNetwork.VirtualStorageLibrary.VirtualNodeContext)          |      |  |

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeExtensions)

**Type** `(AkiraNetwork.VirtualStorageLibrary)`

## **VirtualNodeName**

**CompareTo<T>()**  
(AkiraNetwork.VirtualStorageLibrary  
VirtualNodeNotFoundException

**CompareTo(Obj):**  
(AkiraNetwork.VirtualStorageLibrary)

Computes the storage's path with another  
(AkiraNetwork.VirtualStorageLibrary)

## Declaration

### VirtualNodeTypeFilter

```
(AkiraNetwork.VirtualStorageLibrary  
public IntPtr CompareTo(object? ob  
(AkiraNetwork.VirtualStorageLibrary
```

(AkiraNetwork.virtualStorageLibrary)

| Type  | Description  |
|---|--|
| AkiraNetwork.VirtualStorageLibrary.VirtualNodeListConditions<br>int ( <a href="https://learn.microsoft.com/dotnet/api/system.int32">https://learn.microsoft.com/dotnet/api/system.int32</a> )<br>(AkiraNetwork.VirtualStorageLibrary) | An integer indicating the relative order of the paths. |

## CompareTo(object?)

Compare object  
(AkiraNetwork.VirtualStorageLibrary)

(AkiraNetwork.VirtualStorageLibrary)

## Declaration

### VirtualNodeTypeFilter

(AkiraNetwork.VirtualStorageLibrary  
public virtual CompareTo(object? ob

(AkiraNetwork.virtualStorageLibrary)

## Parameters

| 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> | The object to compare with. |

## Returns

| Type  | Description  |
|---|--|
| int ( <a href="https://learn.microsoft.com/dotnet/api/system.int32">https://learn.microsoft.com/dotnet/api/system.int32</a> 1.html) | An integer indicating the relative order of the paths. |

IVirtualWildcardMatcher  
Exceptions  
(AkiraNetwork.VirtualStorageLibrary.VirtualPathException)

| Type  | Condition   |
|---|---|
| NotifyNodeDelegate  |   |
| (AkiraNetwork.VirtualStorageLibrary.ArgumentException)  | Thrown if the object is not a VirtualPath             |
| PatternMatch<br>( <a href="https://learn.microsoft.com/dotnet/api/system.argumentexception">https://learn.microsoft.com/dotnet/api/system.argumentexception</a> ) | (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) |
| VirtualCycleDetector  |   |
| (AkiraNetwork.VirtualStorageLibrary.VirtualDirectory)   |   |
| (AkiraNetwork.VirtualStorageLibrary.VirtualEvent)   |   |
| (AkiraNetwork.VirtualStorageLibrary.VirtualItem)  |   |
| VIRTUALITEM<T>  |   |

## Equals(VirtualPath?)

(AkiraNetwork.VirtualStorageLibrary.VirtualPath)  
Indicates whether this path is equal to the specified VirtualPath  
(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) object.

Declaration  
VirtualID

(AkiraNetwork.VirtualStorageLibrary.VirtualItem)  
public bool Equals(VirtualPath? other)

## Parameters

| Type        | Name         | Description   |
|-------------|--------------|---|
| VirtualPath | <i>other</i> | The VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) object to compare with the instance's path. |

## Returns

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeExtensions)

| Type                      | Description   |
|---------------------------|---|
| VirtualNodeListConditions | bool ( <a href="https://learn.microsoft.com/dotnet/api/system.boolean">https://learn.microsoft.com/dotnet/api/system.boolean</a> ) true if the specified VirtualnodeName ( <a href="https://learn.microsoft.com/dotnet/api/system.string">https://learn.microsoft.com/dotnet/api/system.string</a> ) is equal to the instance's path; otherwise, false. |

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeNotFoundException)

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeType)

**Equals(object?)** (AkiraNetwork.VirtualStorageLibrary.VirtualStorageLibrary)

VirtualNodeTypeFilter

Indicates whether this path is equal to the specified object.

(AkiraNetwork.VirtualStorageLibrary.VirtualPath)

Declaration  
(AkiraNetwork.VirtualStorageLibrary)

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

#### Parameters

| Type   | Name | Description                                     |
|--|------|---|
| object<br>( <a href="https://learn.microsoft.com/dotnet/api/system.object">https://learn.microsoft.com/dotnet/api/system.object</a> )<br>1.html) | obj  | The object to compare with the instance's path. |

#### Returns

| Type   | Description   |
|--|---|
| bool<br>( <a href="https://learn.microsoft.com/dotnet/api/system.boolean">https://learn.microsoft.com/dotnet/api/system.boolean</a> )<br>( <a href="https://learn.microsoft.com/dotnet/api/system.boolean">https://learn.microsoft.com/dotnet/api/system.boolean</a> )<br>VirtualCycleDetector | true if the specified object is equal to the instance's path; otherwise, false. |

#### Overrides

|   |                                      |
|---|--------------------------------------|
| VirtualDirectory<br>object.Equals(object) ( <a href="https://learn.microsoft.com/dotnet/api/system.object.equals(system.object)">https://learn.microsoft.com/dotnet/api/system.object.equals(system.object)</a> ) | (AkiraNetwork.VirtualStorageLibrary) |
| VirtualException  | (AkiraNetwork.VirtualStorageLibrary) |

### GetHashCode()

| Declaration                          |
|--------------------------------------|
| (AkiraNetwork.VirtualStorageLibrary) |

Retrieves the hash code for this path.

#### Declaration

|                                      |
|--------------------------------------|
| VirtualID                            |
| (AkiraNetwork.VirtualStorageLibrary) |

#### Overrides

|                                      |
|--------------------------------------|
| VirtualItem                          |
| (AkiraNetwork.VirtualStorageLibrary) |

#### Returns

|               |
|---------------|
| 1.html)       |
| (VirtualNode) |

#### Type

| Type   | Description                  |
|--|------------------------------|
| VirtualNode<br>( <a href="https://learn.microsoft.com/dotnet/api/system.int32">https://learn.microsoft.com/dotnet/api/system.int32</a> )<br>VirtualNodeContext | The hash code for this path. |

#### Overrides

|                                      |
|--------------------------------------|
| VirtualNodeExtensions                |
| (AkiraNetwork.VirtualStorageLibrary) |

#### VirtualNodeListConditions

|                                      |
|--------------------------------------|
| (AkiraNetwork.VirtualStorageLibrary) |
| (VirtualNodeName)                    |

#### VirtualNodeName

Retrieves the name of the instance's path.

#### Declaration

|                                      |
|--------------------------------------|
| (AkiraNetwork.VirtualStorageLibrary) |
| (VirtualNodeType)                    |

#### public VirtualNodeName GetnodeName()

|                                      |
|--------------------------------------|
| VirtualNodeTypeFilter                |
| (AkiraNetwork.VirtualStorageLibrary) |

|                                      |
|--------------------------------------|
| VirtualPath                          |
| (AkiraNetwork.VirtualStorageLibrary) |

| Type   | Description   |
|--|---|
| VirtualNodeName<br>(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html) | A VirtualNodeName<br>(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html) object representing the node name. |

## (AkiraNetwork.VirtualStorageLibrary) **GetParentPath()**

IVirtualWildcardMatcher

Gets the parent path of the instance's path.  
(AkiraNetwork.VirtualStorageLibrary)

Declaration

(AkiraNetwork.VirtualStorageLibrary)

PatternMatch

public VirtualPath GetParentPath()  
(AkiraNetwork.VirtualStorageLibrary)

VirtualCycleDetector

Returns (AkiraNetwork.VirtualStorageLibrary)

| Type   | Description |
|--|-------------|
| VirtualDirectory<br>(AkiraNetwork.VirtualStorageLibrary) |             |

VirtualPath

VirtualException  
(AkiraNetwork.VirtualStorageLibrary)

VirtualPath  
(AkiraNetwork.VirtualStorageLibrary)

VirtualGroupCondition<T, TKey>  
(AkiraNetwork.VirtualStorageLibrary)

VirtualPath  
2.html)

VirtualID

(AkiraNetwork.VirtualStorageLibrary)

VirtualItem

VirtualItem  
(AkiraNetwork.VirtualStorageLibrary)

Declaration

VirtualItem<T>

(AkiraNetwork.VirtualStorageLibrary)

public LinkedList<VirtualNodeName> GetPartsLinkedList()

VirtualNode

(AkiraNetwork.VirtualStorageLibrary)

Returns VirtualNodeContext

| Type                               | Description |
|------------------------------------|-------------|
| AkiraNetwork.VirtualStorageLibrary |             |

VirtualNodeExtensions

LinkedList  
(AkiraNetwork.VirtualStorageLibrary)  
(https://learn.microsoft.com/dotnet/api/system.collections.generic.linkedlist-1)  
m.collections.generic.linkedlist-1)

< (AkiraNetwork.VirtualStorageLibrary)

VirtualNodeName

VirtualNodeName  
(AkiraNetwork.VirtualStorageLibrary)

(AkiraNetwork.VirtualStorageLibrary.VirtualNo

VirtualNodeNotFoundException  
deName.html)

> (AkiraNetwork.VirtualStorageLibrary)

VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeTypeFilter

(AkiraNetwork.VirtualStorageLibrary)

VirtualPath

(AkiraNetwork.VirtualStorageLibrary)

## (AkiraNetwork.VirtualStorageLibrary) **GetPartsList()**

VirtualPath

(AkiraNetwork.VirtualStorageLibrary)

Retrieves the node parts of the path as a list. This method converts the node parts stored in a linked list structure into a list structure, making it more convenient for sequential access and operations that require indexing.

## Declaration

```
public List<VirtualnodeName> GetPartsList()
▼
```

**Returns**  
 (AkiraNetwork.VirtualStorageLibrary)

| Type  | Description  |
|---|--|
| 1.html<br>IVirtualWildcardMatcher<br>List<VirtualnodeName><br>(AkiraNetwork.VirtualStorageLibrary)<br>(https://learn.microsoft.com/dotnet/api/system.collections.generic.list-1)<br>NotifyNodeDelegate<br>m.collections.generic.list-1<br>< PatternMatch<br>VirtualnodeName<br>(AkiraNetwork.VirtualStorageLibrary)<br>(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName)<br>VirtualCycleDetector<br>odeName.html<br>(AkiraNetwork.VirtualStorageLibrary)<br>> VirtualDirectory<br>(AkiraNetwork.VirtualStorageLibrary)<br>VirtualException | A<br>List<T><br>(https://learn.microsoft.com/dotnet/api/system.collections.generic.list-1)<br>of<br>VirtualnodeName<br>(AkiraNetwork.VirtualStorageLibrary.VirtualnodeName.html)<br>objects representing the node parts of the path. |

## GetRelativePath(VirtualPath)

VirtualGroupCondition<T, TKey>

Gets the relative path from the specified base path.

2.html  
Declaration  
VirtualID

(AkiraNetwork.VirtualStorageLibrary)  
public VirtualPath GetRelativePath(VirtualPath basePath)

VirtualItem

Parameters

(AkiraNetwork.VirtualStorageLibrary)

Type

1.html

VirtualNode

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeContext

(AkiraNetwork.VirtualStorageLibrary)

Type

virtualNodeExtensions

(AkiraNetwork.VirtualStorageLibrary)

VirtualPath

VirtualNodeListConditions

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

(AkiraNetwork.VirtualStorageLibrary)

VirtualnodeName

(AkiraNetwork.VirtualStorageLibrary)

Exceptions

VirtualNodeNotFoundException

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeTypeFilter

(AkiraNetwork.VirtualStorageLibrary)

VirtualPath

(AkiraNetwork.VirtualStorageLibrary)

| Type        | Name     | Description    |
|-------------|----------|----------------|
| VirtualPath | basePath | The base path. |

| Type        | Description   |
|-------------|---|
| VirtualPath | A VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) representing the relative path. |

| Type  | Condition   |
|---|---|
| InvalidOperationException<br>( <a href="https://learn.microsoft.com/dotnet/api/system.invalidoperationexception">https://learn.microsoft.com/dotnet/api/system.invalidoperationexception</a> )<br>(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) | Thrown if the instance's path is not absolute: This exception occurs when the method is called on a VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) instance that does not start with the root path separator, indicating that it is not an absolute path.<br>Thrown if the base path is not absolute: This exception occurs when the <code>basePath</code> argument does not start with the root path separator, indicating that it is not an absolute path. Both the instance's path and the base path must be absolute paths to compute a relative path. |

## IVirtualWildcardMatcher

### (AkiraNetwork.VirtualStorageLibrary.VirtualPath)

Determines whether this instance's path is a subdirectory of the specified parent path.

#### Declaration

```
VirtualCycleDetector
(AkiraNetwork.VirtualStorageLibrary.VirtualPath parentPath)
VirtualDirectory
(AkiraNetwork.VirtualStorageLibrary.VirtualPath parentPath)
```

#### Parameters

VirtualException

| Type  | Name              | Description   |
|---|-------------------|---|
| (AkiraNetwork.VirtualStorageLibrary.VirtualGroupCondition<T, TKey><br>VirtualPath<br>(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)<br>VirtualID<br>(AkiraNetwork.VirtualStorageLibrary.VirtualItem<br>VirtualItem | <i>parentPath</i> | The parent<br>VirtualPath<br>(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)<br>. . . |

#### Return

(AkiraNetwork.VirtualStorageLibrary.VirtualItem<T>

| Type  | Description   |
|---|---|
| (AkiraNetwork.VirtualStorageLibrary.VirtualNodeContext<br>bool.html)<br>( <a href="https://learn.microsoft.com/dotnet/api/system.argumentexception">https://learn.microsoft.com/dotnet/api/system.argumentexception</a> )<br>bool(AkiraNetwork.VirtualStorageLibrary.VirtualNodeContext<br>VirtualNodeContext | true if this instance's path is a subdirectory of the parent path;<br>otherwise, false. |

#### Exceptions

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeExtensions  
VirtualNodeException

| Type  | Condition   |
|---|---|
| (AkiraNetwork.VirtualStorageLibrary.VirtualNodeListConditions<br>ArgumentOutOfRangeException<br>( <a href="https://learn.microsoft.com/dotnet/api/system.argumentexception">https://learn.microsoft.com/dotnet/api/system.argumentexception</a> )<br>VirtualNodeName<br>(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName<br>VirtualNodeNotFoundException<br>(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName<br>VirtualNodeTypeFilter | Thrown if this instance's path or the parent path is empty. |

## NormalizePath()

(AkiraNetwork.VirtualStorageLibrary.VirtualPath  
Normalizes the instance's path.

| Type   | Condition |
|--|-----------|
| (AkiraNetwork.VirtualStorageLibrary.VirtualPath<br>(AkiraNetwork.VirtualStorageLibrary.VirtualPath<br>VirtualNodeTypeFilter<br>VirtualPath<br>(AkiraNetwork.VirtualStorageLibrary.VirtualPath<br>VirtualPath |           |

```
public VirtualPath NormalizePath()
```

## Returns

| Type   | Description  |
|--|--|
| VirtualPath<br>(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) | A new VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) instance with the normalized path. |

IVirtualWildcardMatcher  
(AkiraNetwork.VirtualStorageLibrary)

NotifyNodeDelegate  
(AkiraNetwork.VirtualStorageLibrary)

PatternMatch  
Normalizes the specified path string.  
(AkiraNetwork.VirtualStorageLibrary)

Declaration  
VirtualCycleDetector  
(AkiraNetwork.VirtualStorageLibrary)

```
public VirtualDirectory NormalizePath(string path)
```

VirtualException  
Parameters  
(AkiraNetwork.VirtualStorageLibrary)

| Type   | Name | Description                   |
|--|------|-------------------------------|
| VirtualGroupCondition<T, TKey><br>(AkiraNetwork.VirtualStorageLibrary) | path | The path string to normalize. |

VirtualID  
Returns  
(AkiraNetwork.VirtualStorageLibrary)

| Type  | Description               |
|---|---------------------------|
| VirtualItem<br>(AkiraNetwork.VirtualStorageLibrary) | A normalized path string. |

VirtualItem<T>  
(AkiraNetwork.VirtualStorageLibrary)

Exceptions  
(AkiraNetwork.VirtualStorageLibrary)

| Type  | Condition  |
|---|--|
| InvalidOperationContext<br>(AkiraNetwork.VirtualStorageLibrary) | Thrown when attempting to navigate above the root directory. |

VirtualNode  
(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeExtensions  
(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeListConditions  
(AkiraNetwork.VirtualStorageLibrary)

VirtualnodeName  
(AkiraNetwork.VirtualStorageLibrary)

Determines whether the instance's path starts with the specified path.  
VirtualNodeNotFoundException  
(AkiraNetwork.VirtualStorageLibrary)

Declaration  
(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeType  
(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeTypeFilter  
(AkiraNetwork.VirtualStorageLibrary)

Parameters  
VirtualPath  
(AkiraNetwork.VirtualStorageLibrary)

| Type   | Name | Description  |
|--|------|--|
| VirtualPath<br>(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) | path | The VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) to compare with. |

## Returns

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

| Type  | Description   |
|---|---|
| bool<br>VirtualWildcardMatcher<br>(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) | true if the instance's path starts with the specified path; otherwise, false. |

NotifyNodeDelegate  
olean  
(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

PatternMatch

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

## ToString()

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

Returns a string that represents the instance's path.

## Declaration

VirtualException

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

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

## Returns

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

| Type   | Description                                   |
|--|---|
| VirtualID<br>(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) | A string that represents the instance's path. |

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

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

## Overrides

VirtualItem<T>

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

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

1.html

VirtualNode

## TrimEndSlash()

VirtualNodeContext

Removes the trailing slash from the path.

## Declaration

VirtualNodeExtensions

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

VirtualNodeListConditions

public VirtualPath TrimEndSlash()  
(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

VirtualnodeName

## Returns

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

| Type  | Description   |
|---|---|
| VirtualNodeNotFoundException<br>(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) | A new VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) without the trailing slash. |

VirtualNodeType  
(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

VirtualNodeFilter  
(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

VirtualPath

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

# Operators

## operator +(VirtualPath, VirtualNodeName)

Combines a VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) with a VirtualNodeName (AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html).

### Declaration

```
(AkiraNetwork.VirtualStorageLibrary.  
    1.html)  
public static VirtualPath operator +(VirtualPath path, VirtualNodeName nodeName)
```

(AkiraNetwork.VirtualStorageLibrary.

NotifyNodeDelegate

### Parameters

(AkiraNetwork.VirtualStorageLibrary.

| Type   | Description  |
|--|--|
| PatternMatch<br>(AkiraNetwork.VirtualStorageLibrary.<br>VirtualPath<br>VirtualCycleDetector<br>(AkiraNetwork.VirtualStorageLibrary.VirtualPat<br>h.html)<br>VirtualDirectory<br>(AkiraNetwork.VirtualStorageLibrary.<br>VirtualException | <i>path</i><br>The<br>VirtualPath<br>(AkiraNetwork.VirtualStorageLibrary.VirtualPath.htm<br>l)             |
| VirtualItem<br>(AkiraNetwork.VirtualStorageLibrary.<br>VirtualNodeDelegate<br>(AkiraNetwork.VirtualStorageLibrary.VirtualNo<br>de.html)<br>VirtualID<br>(AkiraNetwork.VirtualStorageLibrary.<br>VirtualItem                              | <i>nodeName</i><br>The<br>VirtualNodeName<br>(AkiraNetwork.VirtualStorageLibrary.VirtualNodeNa<br>me.html) |

### Returns

| Type  | Description   |
|---|---|
| VirtualPath<br>(AkiraNetwork.VirtualStorageLibrary.<br>VirtualPath<br>VirtualPath.html)<br>(AkiraNetwork.VirtualStorageLibrary.Virtu<br>alPath.html)<br>(AkiraNetwork.VirtualStorageLibrary.<br>VirtualNodeContext<br>(AkiraNetwork.VirtualStorageLibrary.<br>VirtualNodeExtensions | A new<br>VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)<br>representing the combined path. |

## operator +(VirtualPath, VirtualPath)

Combines two VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) instances.

VirtualNodeListConditions

### Declaration

```
(AkiraNetwork.VirtualStorageLibrary.  
VirtualNodeName  
public static VirtualPath operator +(VirtualPath path1, VirtualPath path2)  
    VirtualNodeNotFoundException  
    (AkiraNetwork.VirtualStorageLibrary.
```

### Parameters

VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary.

VirtualNodeTypeFilter

(AkiraNetwork.VirtualStorageLibrary.

VirtualPath

(AkiraNetwork.VirtualStorageLibrary.



Combines a VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) with a string.

## Declaration

```
public static VirtualPath operator +(VirtualPath path, string str)
```



### Parameters

| Type  | Name | Description  |
|---|------|--|
| (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)<br>VirtualPath<br>VirtualWildcardMatcher<br>(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)<br>NotifyNodeDelegate<br>(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)<br>PatternMatch<br>(https://learn.microsoft.com/en-us/dotnet/api/system.string)<br>VirtualCycleDetector<br>(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)<br>VirtualDirectory | path | The VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html). |
| string  | str  | The string to combine with the path.                                   |

### Returns

(AkiraNetwork.VirtualStorageLibrary.VirtualPath)

| Type             | Description   |
|------------------|---|
| VirtualException | A new VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) representing the combined path. |

VirtualID

(AkiraNetwork.VirtualStorageLibrary.VirtualPath)

## operator +(char, VirtualPath)

(AkiraNetwork.VirtualStorageLibrary.VirtualPath)

Combines a character with a VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html).

VirtualItem<T>

(AkiraNetwork.VirtualStorageLibrary.VirtualPath)

### Declaration

1.html)

VirtualNode

```
public static VirtualPath operator +(char chr, VirtualPath path)
```

(AkiraNetwork.VirtualStorageLibrary.VirtualPath)

VirtualNodeContext

### Parameters

(AkiraNetwork.VirtualStorageLibrary.VirtualPath)

| Type   | Name | Description  |
|--|------|--|
| VirtualNodeExtensions<br>(AkiraNetwork.VirtualStorageLibrary.VirtualPath)                                | chr  | The character to combine with the path.                                |
| VirtualNodeListConditions<br>(https://learn.microsoft.com/en-us/dotnet/api/system.virtualnodeconditions) | path | The VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html). |

VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary.VirtualPath)

VirtualNodeTypeFilter

(AkiraNetwork.VirtualStorageLibrary.VirtualPath)

VirtualPath

(AkiraNetwork.VirtualStorageLibrary.VirtualPath)

### Returns

(AkiraNetwork.VirtualStorageLibrary.VirtualPath)

| Type   | Description   |
|--|---|
| VirtualPath<br>(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) | A new<br>VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)<br>representing the combined path. |

二

**operator + (string, VirtualPath)**

1.html)  
Combines a string with a VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html).  
VirtualWildcardMatcher

## DeclarationNetwork.VirtualStorageLibrary

## NotifyNodeDelegate

```
(AkiraNetwork.VirtualStorageLibrary) public static VirtualPath operator +(string str, VirtualPath path) { PatternMatch
```

## Patternlivatem (Alpine Nature Park)

(AkiraNetwork.virtualStorageLibraryParameters) [Root]

## Parameters

-(AkiraNetwork Virtual)

| Type   | Name        | Description  |
|--|-------------|--|
| VirtualDirectory<br>string<br>(AkiraNetwork.VirtualStorageLibrary)<br>( <a href="http://learn.microsoft.com/dotnet/api/system">http://learn.microsoft.com/dotnet/api/system</a> .<br>VirtualException<br>string)<br>(AkiraNetwork.VirtualStorageLibrary)           | <i>str</i>  | The string to combine with the path.   |
| VirtualGroupCondition<T, TKey><br>VirtualPath<br>(AkiraNetwork.VirtualStorageLibrary)<br>( <a href="http://learn.microsoft.com/dotnet/api/system">http://learn.microsoft.com/dotnet/api/system</a> .<br>html)<br>VirtualID<br>(AkiraNetwork.VirtualStorageLibrary) | <i>path</i> | The<br>VirtualPath<br>(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)<br>. . |

## Virt Returns

(AkiraNetwork.VirtualStorageLibrary)

| Type  | Description   |
|---|---|
| <code>virtualItem&lt;T&gt;</code>   |   |
| <code>(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)</code>        | A new <code>VirtualPath</code> ( <code>AkiraNetwork.VirtualStorageLibrary.VirtualPath.html</code> ) representing the combined path. |
| <code>(AkiraNetwork.VirtualStorageLibrary.VirtualNode.alPath.html)</code> |   |
| <code>(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)</code>        |   |

## VirtualNodeContext

(AkiraNetwork.VirtualStorageLibrary)

## **operator (VirtualPath?, VirtualPath?)**

(AkiraNetwork.VirtualStorageLibrary)

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

→ (AkiraN)

## **VirtualNetwork.Virt laration**

**VirtualNodeName**  
(AlibiNetworks VirtualStorage Library)

```
(AkkaNetwork.VirtualStorageLibrary  
public static bool operator ==(VirtualPath? left, VirtualPath? right)  
VirtualNodeNotFoundException
```

(AkiraNetwork.VirtualStorageLibrary)

## Parameters

(AkiraNetwork.VirtualStorageLibrary)

## VirtualNodeTypeFilter

(AkiraNetwork.VirtualStorageLibrary)

## VirtualPath

(AkiraNetwork VirtualStorage) library

| Type   | Name         | Description  |
|--|--------------|--|
| VirtualPath<br>(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) | <i>left</i>  | The left<br>VirtualPath<br>(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)<br>.  |
| VirtualPath<br>(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) | <i>right</i> | The right<br>VirtualPath<br>(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)<br>. |

Returns [NotifyNodeDelegate](#)

| Type  | Description                                    |
|---|--|
| PatternMatch<br>bool(AkiraNetwork.VirtualStorageLibrary.net/api/system.boolean) | true if the paths are equal; otherwise, false. |

VirtualCycleDetector

(AkiraNetwork.VirtualStorageLibrary)

VirtualDirectory

(AkiraNetwork.VirtualStorageLibrary)

## implicit operator string(VirtualPath?)

Implicitly converts a VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) to a string  
(AkiraNetwork.VirtualStorageLibrary.net/api/system.string).  
(<https://learn.microsoft.com/dotnet/api/system.string>).

VirtualGroupCondition<T, TKey>

(AkiraNetwork.VirtualStorageLibrary)

2.html)

public static implicit operator string(VirtualPath? virtualPath)

(AkiraNetwork.VirtualStorageLibrary)

VirtualItem

(AkiraNetwork.VirtualStorageLibrary)

Type virtualItem<T>

(AkiraNetwork.VirtualStorageLibrary)

VirtualPath

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

VirtualNode

(AkiraNetwork.VirtualStorageLibrary.VirtualNode.html)

VirtualNodeContext

(AkiraNetwork.VirtualStorageLibrary)

Returns [VirtualNodeExtensions](#)

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeListConditions

string (AkiraNetwork.VirtualStorageLibrary)

(<https://learn.microsoft.com/dotnet/api/system.string>) The string representation of the

VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html), or an

empty string if the path is null.  
(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeNotFoundException

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary)

## implicit operator VirtualPath(string)

(AkiraNetwork.VirtualStorageLibrary)

Implicitly converts a string (<https://learn.microsoft.com/dotnet/api/system.string>) to a VirtualPath

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html).

VirtualPath

(AkiraNetwork.VirtualStorageLibrary)

```
public static implicit operator 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 | The string representation of the path. |

## Returns

| Type  | Description   |
|---|---|
| IVirtualWildcardMatcher<br>(AkiraNetwork.VirtualStorageLibrary) | A VirtualPath ( <a href="https://learn.microsoft.com/dotnet/api/AkiraNetwork.VirtualStorageLibrary.VirtualPath.html">AkiraNetwork.VirtualStorageLibrary.VirtualPath.html</a> ) object representing the given string path. |

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

Determines whether two VirtualPath ([AkiraNetwork.VirtualStorageLibrary.VirtualPath.html](https://learn.microsoft.com/dotnet/api/AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)) instances are not equal.

## Returns

Declared by [AkiraNetwork.VirtualStorageLibrary](https://learn.microsoft.com/dotnet/api/AkiraNetwork.VirtualStorageLibrary)  
VirtualGroupCondition<T, TKey>  
([AkiraNetwork.VirtualStorageLibrary](https://learn.microsoft.com/dotnet/api/AkiraNetwork.VirtualStorageLibrary))

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

2.html)

## VirtualID

### Parameters

| Type  | Name  | Description  |
|---|-------|--|
| VirtualItem<br>(AkiraNetwork.VirtualStorageLibrary) | left  | The left<br>VirtualPath<br>( <a href="https://learn.microsoft.com/dotnet/api/AkiraNetwork.VirtualStorageLibrary.VirtualPath.html">AkiraNetwork.VirtualStorageLibrary.VirtualPath.html</a> )<br>1.html) |
| VirtualNode<br>(AkiraNetwork.VirtualStorageLibrary) | right | The right<br>VirtualPath<br>( <a href="https://learn.microsoft.com/dotnet/api/AkiraNetwork.VirtualStorageLibrary.VirtualPath.html">AkiraNetwork.VirtualStorageLibrary.VirtualPath.html</a> )<br>1.     |

## VirtualNodeListConditions

### Returns

| Type   | Description  |
|--|--|
| VirtualNodeName<br>( <a href="https://learn.microsoft.com/dotnet/api/system.boolean">https://learn.microsoft.com/dotnet/api/system.boolean</a> ) | true if the paths are not equal; otherwise, false. |

## VirtualNodeType

### Implements

|                |   |
|----------------|---|
| IComparable<T> | <a href="https://learn.microsoft.com/dotnet/api/system.ienumerable-1">https://learn.microsoft.com/dotnet/api/system.ienumerable-1</a> |
| IComparable<T> | <a href="https://learn.microsoft.com/dotnet/api/system.icomparable-1">https://learn.microsoft.com/dotnet/api/system.icomparable-1</a> |
| IComparable    | <a href="https://learn.microsoft.com/dotnet/api/system.icomparable">https://learn.microsoft.com/dotnet/api/system.icomparable</a>     |

# Extension Methods

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

▼

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

# Class VirtualSortCondition<T>

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

**Inheritance**  
[\(AkiraNetwork.VirtualStorageLibrary-1.html\)](#)

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

**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)))  
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)))  
VirtualCycleDetector  
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>)  
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>

**Namespace** [\(AkiraNetwork.VirtualStorageLibrary\)](#) [\(AkiraNetwork.VirtualStorageLibrary-1.html\)](#) [\(AkiraNetwork.VirtualStorageLibrary\)](#)

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

**Syntax**

```
(AkiraNetwork.VirtualStorageLibrary)
VirtualItem
public class VirtualSortCondition<T>
(AkiraNetwork.VirtualStorageLibrary)
VirtualItem<T>
```

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

| Name | Description                          |
|------|--------------------------------------|
| T    | The type of the entity to be sorted. |

**Constructors**

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

Initializes a new instance of the VirtualSortCondition<T> ([\(AkiraNetwork.VirtualStorageLibrary\)](#)) class with the specified sorting property and order ([\(AkiraNetwork.VirtualStorageLibrary\)](#)).

**Declaration**

```
(AkiraNetwork.VirtualStorageLibrary)
VirtualNodeTypeFilter
public VirtualSortCondition(Expression<Func<T, object>> sortBy, bool ascending = true)
(AkiraNetwork.VirtualStorageLibrary)
```

**VirtualPath**

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



# Extension Methods

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

▼

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

# 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)

↳ VirtualStorageExtensions

(AkiraNetwork.VirtualStorageLibrary)

## Inherited Members

NotifyNodeDelegate

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

PatternMatch

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

VirtualCycleDetector

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

(AkiraNetwork.VirtualStorageLibrary)

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

VirtualDirectory

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)

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)

2.html)

## Syntax

VirtualID

(AkiraNetwork.VirtualStorageLibrary)  
public static class VirtualStorageExtensions

VirtualItem

(AkiraNetwork.VirtualStorageLibrary)

VirtualItem<T>

(AkiraNetwork.VirtualStorageLibrary)

## Methods

VirtualNode

(AkiraNetwork.VirtualStorageLibrary)

### ApplySortConditions<T>(IEnumerable<T>, List<VirtualSortCondition<T>>?)

(AkiraNetwork.VirtualStorageLibrary)

Applies the specified sort conditions to the elements of the source sequence.

(AkiraNetwork.VirtualStorageLibrary)

## Declaration

VirtualNodeListConditions

(AkiraNetwork.VirtualStorageLibrary)  
public static IEnumerable<T> ApplySortConditions<T>(this IEnumerable<T> source, List<VirtualNodeName> sortConditions = null)

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeNotFoundException

## Parameters

VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeTypeFilter

(AkiraNetwork.VirtualStorageLibrary)

VirtualPath

(AkiraNetwork.VirtualStorageLibrary)

| Type  | Name   | Description                       |
|---|--------|-----------------------------------|
| IEnumerable<br>( <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> )<br><br><T><br>▼<br>List<br>( <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> )<br>< IVirtualWildcardMatcher<br>VirtualSortCondition<br>(AkiraNetwork.VirtualStorageLibrary<br>(AkiraNetwork.VirtualStorageLibrary.VirtualSortCondition-1.html))<br>(AkiraNetwork.VirtualStorageLibrary<br><T> PatternMatch<br>(AkiraNetwork.VirtualStorageLibrary) | source | The sequence of elements to sort. |

Return Value

| Type   | Description  |
|--|--|
| VirtualDirectory<br>(AkiraNetwork.VirtualStorageLibrary) | An<br>IEnumerable<T><br>( <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> )<br>containing the sorted elements. |

Type Parameters

| Name | Description   |
|------|---|
| T    | (AkiraNetwork.VirtualStorageLibrary) The type of elements in the source sequence.<br>VirtualItem<T><br>(AkiraNetwork.VirtualStorageLibrary<br>1.html) |

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

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeContext

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

(AkiraNetwork.VirtualStorageLibrary)

Declaration

(AkiraNetwork.VirtualStorageLibrary)

```
public static IEnumerable<T> GroupAndSort<T>(this IEnumerable<T> source, VirtualGroupCondition<T, object>? groupCondition, List<VirtualSortCondition<T>>? sortConditions = null)
```

VirtualNodeName

(AkiraNetwork.VirtualStorageLibrary)

Parameters

VirtualNodeNotFoundException

| Type  | Name   | Description                                 |
|---|--------|---|
| IEnumerable<br>(AkiraNetwork.VirtualStorageLibrary<br>( <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> )<br><T><br>VirtualPath<br>(AkiraNetwork.VirtualStorageLibrary<br><T> VirtualPath<br>(AkiraNetwork.VirtualStorageLibrary) | source | The sequence of elements to group and sort. |

| Type   | Name                  | Description   |
|--|-----------------------|---|
| <p>VirtualGroupCondition<br/>(AkiraNetwork.VirtualStorageLibrary.VirtualGroupCondition-<br/>n-2.html)</p> <p>&lt;T,<br/>object<br/>(<a href="https://learn.microsoft.com/dotnet/api/system.object">https://learn.microsoft.com/dotnet/api/system.object</a>)<br/>&gt; (AkiraNetwork.VirtualStorageLibrary-<br/>1.html)</p>   | <i>groupCondition</i> | The condition that defines how to group the elements. If null, no grouping is applied.    |
| <p>List&lt;IVirtualWildcardMatcher<br/>(<a href="https://learn.microsoft.com/dotnet/api/system.collection.sgenomic.list-1">https://learn.microsoft.com/dotnet/api/system.collection.sgenomic.list-1</a>)<br/>NotifyNodeDelegate<br/>&lt; (AkiraNetwork.VirtualStorageLibrary-<br/>VirtualSortCondition<br/>PatternMatcher<br/>(AkiraNetwork.VirtualStorageLibrary.VirtualSortCondition-<br/>1.html)<br/>VirtualCycleDetector<br/>&lt;T&gt;(AkiraNetwork.VirtualStorageLibrary-</p> | <i>sortConditions</i> | A list of conditions that define the sort order. If null or empty, no sorting is applied. |

**Befüllen der AkiraNetwork VirtualStorage Library**

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

Type(AkiraNetwork VirtualStorage) iibrar

| Name     | Description                                  |
|----------|--|
| <i>T</i> | The type of elements in the source sequence. |

# Class VirtualStorageSettings

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

**Inheritance**  
[\(AkiraNetwork.VirtualStorageLibrary 1.html\)](#)

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

**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)))  
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)))  
VirtualCycleDetector  
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>)  
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>

**Namespace** [\(AkiraNetwork.VirtualStorageLibrary\)](#) [\(AkiraNetwork.VirtualStorageLibrary 1.html\)](#) [\(AkiraNetwork.VirtualStorageLibrary\)](#)

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

**Syntax**

```
(AkiraNetwork.VirtualStorageLibrary)
VirtualItem
public class VirtualStorageSettings
(AkiraNetwork.VirtualStorageLibrary)
VirtualItem<T>
(AkiraNetwork.VirtualStorageLibrary)
1.html)
```

## Properties

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

VirtualNodeContext

**InvalidNodeNameCharacters**

VirtualNodeExtensions

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

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

VirtualNodeListConditions

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

VirtualnodeName

```
public char[] InvalidNodeNameCharacters { get; set; }
```

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

VirtualNodeNotFoundException

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

VirtualNodeType

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

VirtualNodeTypeFilter

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

[] VirtualPath

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

| Type   | Description   |
|--------|---|
| char[] | 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 | Value  |
|----------|--|
| Type     | VirtualWildcardMatcher   |
| string   | (AkiraNetwork.VirtualStorageLibrary.NotifyNodeDelegate (https://learn.microsoft.com/dotnet/api/system.string)) |
| []       | (AkiraNetwork.VirtualStorageLibrary.PatternMatch)  |
|          | (AkiraNetwork.VirtualStorageLibrary.IVirtualStorage)   |

# NodeListConditions

(AkiraNetwork.VirtualStorageLibrary)  
Gets or sets the conditions for listing nodes.  
**VirtualException**

## Declaration (Akira N)

```
VirtualGroupCondition<T, TKey>
(AkiraNetwork.VirtualStorageLibrary
public VirtualNodeListConditions NodeListConditions { get; set; }  
2.html)
```

## VirtualID

| Property                  | Type   | Value   |
|---------------------------|--|---|
| Type                      | VirtualItem  | (AkiraNetwork.VirtualStorageLibrary)                  |
| VirtualNodeListConditions | (AkiraNetwork.VirtualStorageLibrary.Virtual NodeListConditions.html) | The conditions used to filter, group, and sort nodes. |

(Akira)  
PathDot.

## `VirtualNodeContext`

sets or gets the path represented by a virtual network interface.

VirtualN

# Laration (AkhanNetwork.Virtua

## Virtual NodeList Conditions

publ. by N

## VirtualNodeName

**VirtualNodeName**  
(Ali, Net, and Mist [IC])

**Property Value**

## VirtualNodeNotFoundException

AkiraNetwork.VirtualStorageLib

## VirtualNodeType

<https://learn.microsoft.com/en-us/absa/network/virtualstorage/>

(AkiraNetwork.VirtualStorageLibrary) Library

## VirtualNodeTypeFilter

The representation of the current directory.

# PathDotDot

## (AkiraNetwork.VirtualStorageLibrary)



## Property Value

| Type  | Description   |
|---|---|
| string<br>( <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

Gets or sets the prefix used for item names. The default value is "item". Used for auto-generating node names.  
VirtualWildcardMatcher

### Declaration

```
NotifyNodeDelegate  
(AkiraNetwork.VirtualStorageLibrary  
public string PrefixItem { get; set; }  
PatternMatch  
(AkiraNetwork.VirtualStorageLibrary)
```

### Property Value

| Type  | Description                                    |
|---|--|
| VirtualDirectory<br>string<br>( <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 item names. |

## PrefixSymbolicLink

Gets or sets the prefix used for symbolic link names. The default value is "link". Used for auto-generating node names.  
VirtualID  
VirtualException  
(AkiraNetwork.VirtualStorageLibrary  
VirtualGroupCondition<T, TKey>  
(AkiraNetwork.VirtualStorageLibrary)

### Declaration

```
VirtualItem  
(AkiraNetwork.VirtualStorageLibrary  
public string> PrefixSymbolicLink { get; set; }  
(AkiraNetwork.VirtualStorageLibrary  
1.html)
```

### Property Value

| Type   | Description   |
|--|---|
| VirtualNodeContext<br>string<br>( <a href="https://learn.microsoft.com/dotnet/api/system.string">https://learn.microsoft.com/dotnet/api/system.string</a> )<br>VirtualNodeExtensions | The prefix added to auto-generated symbolic link names. |

## Settings

Gets the current instance of the settings.  
VirtualNodeName  
(AkiraNetwork.VirtualStorageLibrary)

### Declaration

```
(AkiraNetwork.VirtualStorageLibrary  
VirtualNodeType  
public static VirtualStorageSettings Settings { get; }  
(AkiraNetwork.VirtualStorageLibrary)
```

### VirtualNodeTypeFilter

### Property Value

VirtualPath  
(AkiraNetwork.VirtualStorageLibrary)

| Type   | Description   |
|--|---|
| VirtualStorageSettings<br>(AkiraNetwork.VirtualStorageLibrary.VirtualStorageSettings.html) | The current instance of VirtualStorageSettings<br>(AkiraNetwork.VirtualStorageLibrary.VirtualStorageSettings.html)<br>. |

(AkiraNetwork.VirtualStorageLibrary  
1.html)

## WildcardMatcher

(AkiraNetwork.VirtualStorageLibrary  
Gets or sets the wildcard matcher.  
NotifyNodeDelegate

Declaration  
(AkiraNetwork.VirtualStorageLibrary

PatternMatch

```
public IVirtualWildcardMatcher? WildcardMatcher { get; set; }
```

VirtualCycleDetector

(AkiraNetwork.VirtualStorageLibrary

Property Value  
Directory

| Type   | Description  |
|--|--|
| (AkiraNetwork.VirtualStorageLibrary<br>VirtualException<br>IVirtualWildcardMatcher<br>(AkiraNetwork.VirtualStorageLibrary<br>(AkiraNetwork.VirtualStorageLibrary<br>VirtualGroupCondition, Key<br>ml) (AkiraNetwork.VirtualStorageLibrary<br>2.html) | The implementation of wildcard matching used in the storage. |

VirtualID

(AkiraNetwork.VirtualStorageLibrary  
VirtualItem  
(AkiraNetwork.VirtualStorageLibrary

## Methods

(AkiraNetwork.VirtualStorageLibrary

VirtualItem<T>

Initialized  
1.html)

Initializes the settings. Existing settings are reset, and default settings are applied.

VirtualNode  
(AkiraNetwork.VirtualStorageLibrary

VirtualNodeContext

(AkiraNetwork.VirtualStorageLibrary

```
public static void Initialize()  
VirtualNodeExtensions
```

(AkiraNetwork.VirtualStorageLibrary

VirtualNodeListConditions

(AkiraNetwork.VirtualStorageLibrary

VirtualNodeName

VirtualTextFormatter  
(AkiraNetwork.VirtualStorageLibrary

VirtualNodeNotFoundException  
(AkiraNetwork.VirtualStorageLibrary

VirtualTextFormatter  
(AkiraNetwork.VirtualStorageLibrary

VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary

VirtualNodeTypeFilter

(AkiraNetwork.VirtualStorageLibrary

VirtualPath

(AkiraNetwork.VirtualStorageLibrary

## Extension Methods

(AkiraNetwork.VirtualStorageLibrary

GenerateSingleTableDebugText<T>(T)

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

(AkiraNetwork.VirtualStorageLibrary

VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary

VirtualNodeTypeFilter

(AkiraNetwork.VirtualStorageLibrary

VirtualPath

(AkiraNetwork.VirtualStorageLibrary

# Class VirtualStorageState

Manages the state of the virtual storage. This class implements the singleton pattern and maintains the current setting of [FilterAndState](#).

**Inheritance**  
[\(AkiraNetwork.VirtualStorageLibrary 1.html\)](#)

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

**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)))  
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)))  
VirtualCycleDetector  
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>)  
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>

**Namespace** [\(AkiraNetwork.VirtualStorageLibrary\)](#) [\(AkiraNetwork.VirtualStorageLibrary 1.html\)](#) [\(AkiraNetwork.VirtualStorageLibrary.html\)](#)

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

**Syntax**

```
(AkiraNetwork.VirtualStorageLibrary)
VirtualItem
public class VirtualStorageState
(AkiraNetwork.VirtualStorageLibrary)
VirtualItem<T>
(AkiraNetwork.VirtualStorageLibrary
1.html)
```

## Properties

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

Gets or sets an array of characters that are invalid in node names.  
**Declaration**  
[\(AkiraNetwork.VirtualStorageLibrary\)](#)

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

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

| Type   | Description  |
|--|--|
| char[] ( <a href="https://learn.microsoft.com/dotnet/api/system.char">https://learn.microsoft.com/dotnet/api/system.char</a> ) | An array of characters that are invalid in node names. |
| VirtualPath<br>(AkiraNetwork.VirtualStorageLibrary)  |  |

## InvalidNodeNames

Gets or sets an array of invalid node names.

### Declaration

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

**Property Value**  
(AkiraNetwork.VirtualStorageLibrary  
1.html)

**Type**VirtualWildcardMatcher

(AkiraNetwork.VirtualStorageLibrary  
string (<https://learn.microsoft.com/dotnet/api/system.string>)[]  
NotifyNodeDelegate

### Description

An array of invalid node names.

## NodeListConditions

Gets or sets the conditions for listing nodes.  
(AkiraNetwork.VirtualStorageLibrary  
VirtualCycleDetector

### Declaration

(AkiraNetwork.VirtualStorageLibrary  
VirtualException

```
public VirtualNodeListConditions NodeListConditions { get; set; }
```

(AkiraNetwork.VirtualStorageLibrary  
VirtualGroupCondition<T, TKey>

**Property Value**  
(AkiraNetwork.VirtualStorageLibrary  
2.html)

**Type**VirtualID

(AkiraNetwork.VirtualStorageLibrary  
VirtualNodeListConditions

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeListConditions.html)

### Description

The conditions for listing nodes.

## PathDot

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

(AkiraNetwork.VirtualStorageLibrary  
VirtualNodeContext

(AkiraNetwork.VirtualStorageLibrary  
pVirtualNodeExtension { get; set; }

(AkiraNetwork.VirtualStorageLibrary  
VirtualNodeListConditions

**Property Value**  
(AkiraNetwork.VirtualStorageLibrary  
VirtualNodeName

**Type**virtualNodeName

### Description

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

The path representing the current directory.

(AkiraNetwork.VirtualStorageLibrary  
VirtualNodeType

VirtualNodeType

**PathDotDot**  
(AkiraNetwork.VirtualStorageLibrary  
VirtualNodeTypeFilter

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

(AkiraNetwork.VirtualStorageLibrary  
VirtualPath

**Declaration**  
(AkiraNetwork.VirtualStorageLibrary  
VirtualPath)

```
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> )<br>(AkiraNetwork.VirtualStorageLibrary<br>1.html) | The path representing the parent directory. |

## PathRoot

Gets or sets the root path. The default value is "/".  
(AkiraNetwork.VirtualStorageLibrary  
VirtualNodeDelegate.cs)

### Declaration

```
public string PathRoot { get; set; }  
(AkiraNetwork.VirtualStorageLibrary  
VirtualNode.cs)
```

## Property Value

| Type  | Description    |
|---|----------------|
| string ( <a href="https://learn.microsoft.com/dotnet/api/system.string">https://learn.microsoft.com/dotnet/api/system.string</a> )<br>VirtualGroupCondition<T, TKey><br>(AkiraNetwork.VirtualStorageLibrary<br>VirtualGroup.cs) | The root path. |

## PathSeparator

Gets or sets the path separator character. The default value is '/'.  
VirtualItem  
(AkiraNetwork.VirtualStorageLibrary  
VirtualItem.cs)

### Declaration

```
public char PathSeparator { get; set; }  
1.html  
(AkiraNetwork.VirtualStorageLibrary  
VirtualItem.cs)
```

## Property Value

| Type   | Description                   |
|--|-------------------------------|
| char ( <a href="https://learn.microsoft.com/dotnet/api/system.char">https://learn.microsoft.com/dotnet/api/system.char</a> )<br>(AkiraNetwork.VirtualStorageLibrary<br>VirtualItem.cs) | The path separator character. |

## PrefixDirectory

Gets or sets the prefix used for directory names. The default value is "dir". Used for auto-generating node names.  
VirtualNodeNotFoundException  
(AkiraNetwork.VirtualStorageLibrary  
VirtualNode.cs)

### Declaration

```
public string PrefixDirectory { get; set; }  
VirtualNodeTypeFilter  
(AkiraNetwork.VirtualStorageLibrary  
VirtualNodeType.cs)
```

## Property Value

VirtualPath  
(AkiraNetwork.VirtualStorageLibrary  
VirtualPath.cs)

| 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

```
IVirtualWildcardMatcher
(AkiraNetwork.VirtualStorageLibrary)
public string PrefixItem { get; set; }
NotifyNodeDelegate
(AkiraNetwork.VirtualStorageLibrary)
```

### Property Value

| Type                                 | Description                     |
|--------------------------------------|---------------------------------|
| (AkiraNetwork.VirtualStorageLibrary) | The prefix used for item names. |

### VirtualDirectory

(AkiraNetwork.VirtualStorageLibrary)

### VirtualException

(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; }
```

### Property Value

| Type  | Description                              |
|-------|--|
| .html | The prefix used for symbolic link names. |

### VirtualNodeContext

(AkiraNetwork.VirtualStorageLibrary)

### State

#### VirtualNodeExtensions

(AkiraNetwork.VirtualStorageLibrary)

Gets the current instance of the state.

VirtualNodeListConditions

### Declaration

```
VirtualNodeName
(AkiraNetwork.VirtualStorageLibrary)
public static VirtualStorageState State { get; }
VirtualNodeNotFoundException
(AkiraNetwork.VirtualStorageLibrary)
```

### Property Value

| Type                                 | Description |
|--------------------------------------|-------------|
| (AkiraNetwork.VirtualStorageLibrary) |             |

VirtualStorageState (AkiraNetwork.VirtualStorageLibrary.VirtualStorageState.html)

### VirtualPath

(AkiraNetwork.VirtualStorageLibrary)



| Type   | Name                  | Description                       |
|--|-----------------------|-----------------------------------|
| VirtualNodeTypeFilter<br>(AkiraNetwork.VirtualStorageLibrary.VirtualNodeTypeFilter.html)   | <i>filter</i>         | The filter for node types.        |
| VirtualGroupCondition<br>(AkiraNetwork.VirtualStorageLibrary.VirtualGroupCondition-2.html)<br><VirtualNode (AkiraNetwork.VirtualStorageLibrary.VirtualNode.html),<br>object (https://learn.microsoft.com/dotnet/api/system.object)><br>(AkiraNetwork.VirtualStorageLibrary.VirtualGroupCondition-2.html)   | <i>groupCondition</i> | The condition for grouping nodes. |
| List (https://learn.microsoft.com/dotnet/api/system.collections.generic.list-1)<br>IVirtualWildcardMatcher<br>< (AkiraNetwork.VirtualStorageLibrary.VirtualWildcardMatcher.html)<br>NotifyNodeDelegate<br>< (AkiraNetwork.VirtualStorageLibrary.NotifyNodeDelegate.html)<br>VirtualSortCondition<br>(AkiraNetwork.VirtualStorageLibrary.VirtualSortCondition-1.html)<br>PatternMatch<br><VirtualNode (AkiraNetwork.VirtualStorageLibrary.VirtualNode.html)>><br>(AkiraNetwork.VirtualStorageLibrary.VirtualSortCondition-1.html) | <i>sortConditions</i> | The conditions for sorting nodes. |

## Extension Methods

|  |
|--|
| VirtualTextFormatter.GenerateSingleTableDebugText<T>(T)  |
| (AkiraNetwork.VirtualStorageLibrary.Utilities.VirtualTextFormatter.html#AkiraNetwork_VirtualStorageLibrary_Utilsitie |
| s_VirtualTextFormatter_GenerateSingleTableDebugText_1__0_)   |
| VirtualGroupCondition<T, TKey>   |
| (AkiraNetwork.VirtualStorageLibrary.VirtualGroupCondition-2.html)  |
| VirtualID  |
| (AkiraNetwork.VirtualStorageLibrary.VirtualID.html)  |
| VirtualItem  |
| (AkiraNetwork.VirtualStorageLibrary.VirtualItem.html)  |
| VirtualItem<T>   |
| (AkiraNetwork.VirtualStorageLibrary.VirtualItemT.html)   |
| VirtualNode  |
| (AkiraNetwork.VirtualStorageLibrary.VirtualNode.html)  |
| VirtualNodeContext   |
| (AkiraNetwork.VirtualStorageLibrary.VirtualNodeContext.html)   |
| VirtualNodeExtensions  |
| (AkiraNetwork.VirtualStorageLibrary.VirtualNodeExtensions.html)  |
| VirtualNodeListConditions  |
| (AkiraNetwork.VirtualStorageLibrary.VirtualNodeListConditions.html)  |
| VirtualnodeName  |
| (AkiraNetwork.VirtualStorageLibrary.VirtualnodeName.html)  |
| VirtualNodeNotFoundException   |
| (AkiraNetwork.VirtualStorageLibrary.VirtualNodeNotFoundException.html)   |
| VirtualNodeType  |
| (AkiraNetwork.VirtualStorageLibrary.VirtualNodeType.html)  |
| VirtualNodeTypeFilter  |
| (AkiraNetwork.VirtualStorageLibrary.VirtualNodeTypeFilter.html)  |
| VirtualPath  |
| (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)  |

# Class VirtualStorage<T>

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

Filter by title

## Inheritance

(AkiraNetwork.VirtualStorageLibrary

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

↳ VirtualStorage<T>

↳ VirtualWildcardMatcher

(AkiraNetwork.VirtualStorageLibrary

↳ NotifyNodeDelegate

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

↳ PatternMatch

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

↳ VirtualCycleDetector

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

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

↳ VirtualDirectory

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)

↳ VirtualException

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.html)

## Syntax

VirtualID

(AkiraNetwork.VirtualStorageLibrary  
public class VirtualStorage<T>  
↳ VirtualItem

(AkiraNetwork.VirtualStorageLibrary

## Type Parameters T

| Name | Description  |
|------|--|
| T    | VirtualNode The user-defined type within the virtual storage |

(AkiraNetwork.VirtualStorageLibrary

VirtualNodeContext

(AkiraNetwork.VirtualStorageLibrary

VirtualNodeExtensions

(AkiraNetwork.VirtualStorageLibrary

## VirtualStorage()

(AkiraNetwork.VirtualStorageLibrary

Initializes a new instance of the VirtualStorage class.  
VirtualNodeName

(AkiraNetwork.VirtualStorageLibrary

↳ VirtualNodeNotFoundException

(AkiraNetwork.VirtualStorageLibrary  
public VirtualStorage()  
↳ VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary

↳ VirtualNodeTypeFilter

(AkiraNetwork.VirtualStorageLibrary

↳ VirtualPath

(AkiraNetwork.VirtualStorageLibrary

## Properties

## CurrentPath

Gets or sets the current virtual path.

## Declaration

```
    public VirtualPath CurrentPath { get; }
```

| Property  | Type                   | Value   | Description              |
|---|------------------------|---|--------------------------|
| (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) | VirtualWildcardMatcher |   |                          |
| (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) | VirtualPath            | (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) | The current virtual path |
|   | NotifyNodeDelegate     |   |                          |

NotifyNodeDelegate  
(AkiraNetwork.VirtualStorage  
PatternMatch  
**CycleDetectorForTarget**

**VirtualCycleDetector**  
Gets the cycle detector class used for the WalkPathToTarget method.  
(AKIRA.NETWORK.VirtualStorage library)

## Virtual Directory Declaration

(AkiraNetwork.VirtualStorageLibrary)

```
VirtualException public VirtualCycleDetector CycleDetectorForTarget { get; }  
(AkiraNetwork.VirtualStorageLibrary)
```

## VirtualGroupCondition<T, TKey>

| Property                          | Type   | Description                             |
|-----------------------------------|--|---|
| <code>VirtualCycleDetector</code> | <code>VirtualNetwork.VirtualStorageLibrary</code><br>(Akiri.Network.VirtualStorageLibrary.VirtualCycleDetector.html) | An instance of the cycle detector class |

# CycleDetectorForTree (AkiraNetwork VirtualStorage) [html]

`Gets VirtualNode detector class used for the WalkPathTree method`

Gets the cycle detector class used for the  
(AkiraNetwork.VirtualStorage) library.

### **Declaration**

(AkiraNetwork VirtualStorage) library

```
public void NorthExitCycleDetector_CycleDetectorForTree { get; }
```

(AliNet and Vint) also make them

(AkiraNetwork.VirtualStorageLibrary)

**VirtualNodeListConditions**  
**Property Value** (AkiraNetwork.VirtualStorageLibrary)

| Type  | Description                             |
|---|---|
| <code>virtualNodeName</code>  |   |
| <code>(AkiraNetwork.VirtualStorageLibrary.VirtualCycleDetector)</code>      | An instance of the cycle detector class |
| <code>VirtualNodeNotFoundException</code>                                   |   |
| <code>(AkiraNetwork.VirtualStorageLibrary.VirtualCycleDetector.html)</code> |   |
| <code>(AkiraNetwork.VirtualStorageLibrary)</code>                           |   |

## VirtualNodeType

**VirtualDiskType**  
(AkiraNetwork VirtualStorage library)

**this[VirtualPath, bool]**  
(AkiraNetwork VirtualStorageLibrary)

s or sets the virtual node corresponding to the virtual path.

## Declaration

```
public VirtualDirectory this[VirtualPath path, bool followLinks = true] { get; set; }
```

### Parameters

| Type   | Name        | Description                                      |
|--|-------------|--|
| VirtualPath<br>(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)   | path        | The virtual path                                 |
| bool<br>IWildcardMatcher<br>(AkiraNetwork.VirtualStorageLibrary. <a href="https://learn.microsoft.com/dotnet/api/system.boolean">https://learn.microsoft.com/dotnet/api/system.boolean</a> )<br>n<br>(AkiraNetwork.VirtualStorageLibrary.PatternMatch) | followLinks | Flag indicating whether to follow symbolic links |

**Property Value**  
(AkiraNetwork.VirtualStorageLibrary)

| Type  | Description  |
|---|--|
| VirtualCycleDetector<br>(AkiraNetwork.VirtualStorageLibrary.VirtualDirectory)<br>(AkiraNetwork.VirtualStorageLibrary.VirtualDirectory.html) | The virtual node corresponding to the specified virtual path |

VirtualException  
(AkiraNetwork.VirtualStorageLibrary)

**LinkDictionary**  
VirtualGroupCondition<T, TKey>  
(AkiraNetwork.VirtualStorageLibrary)

Gets the link dictionary.

VirtualID  
**Declaration**  
(AkiraNetwork.VirtualStorageLibrary)

```
public Dictionary<VirtualPath, HashSet<VirtualPath>> LinkDictionary { get; }
```

VirtualItem<T>

**Property Value**  
(AkiraNetwork.VirtualStorageLibrary)

| Type   | Description                                  |
|--|--|
| VirtualNode<br>(AkiraNetwork.VirtualStorageLibrary.Dictionary<br>2) VirtualNodeContext<br><VirtualPath>(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html),<br>VirtualNodeExtensions<br>HashSet<(https://learn.microsoft.com/dotnet/api/system.collections.generic.hashset-1)<VirtualPath>(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)>><br>VirtualNodeListConditions<br>(AkiraNetwork.VirtualStorageLibrary) | A dictionary representing the link mappings. |

VirtualNodeName

**Root**  
(AkiraNetwork.VirtualStorageLibrary)

VirtualNotFoundException

Gets the root directory of the virtual storage.  
(AkiraNetwork.VirtualStorageLibrary)

**Declaration**  
VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeTypeFilter

```
public VirtualDirectory Root { get; }
```

(AkiraNetwork.VirtualStorageLibrary)

VirtualPath

**Property Value**  
(AkiraNetwork.VirtualStorageLibrary)

| Type   | Description                               |
|--|---|
| VirtualDirectory<br>(AkiraNetwork.VirtualStorageLibrary.VirtualDirectory.html) | The root directory of the virtual storage |

▼

## Methods

(AkiraNetwork.VirtualStorageLibrary.VirtualDirectory.AddDirectory.html)

### AddDirectory(VirtualPath, VirtualDirectory, bool)

NotifyNodeDelegate

(AkiraNetwork.VirtualStorageLibrary.VirtualDirectory.NotifyNodeDelegate.html)

Declaration

PATTERNMatch

(AkiraNetwork.VirtualStorageLibrary.VirtualDirectory.PATTERNMatch.html)

`public void AddDirectory(VirtualPath directoryPath, VirtualDirectory directory, bool createsSubdirectories)`

up(AkiraNetwork.VirtualStorageLibrary.VirtualDirectory.up.html)

VirtualDirectory

(AkiraNetwork.VirtualStorageLibrary.VirtualDirectory.html)

Parameters

VirtualException

| Type  | Name                 | Description  |
|---|----------------------|--|
| AkiraNetwork.VirtualStorageLibrary.VirtualPath<br>(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)   | directoryPath        | The path of the directory to add                     |
| AkiraNetwork.VirtualStorageLibrary.VirtualDirectory<br>(AkiraNetwork.VirtualStorageLibrary.VirtualDirectory.html)                                     | directory            | The VirtualDirectory to add                          |
| bool<br>VirtualNodeContext<br>(https://learn.microsoft.com/dotnet/api/system.boolean)<br>(AkiraNetwork.VirtualStorageLibrary.VirtualNodeContext.html) | createSubdirectories | Indicates whether to create intermediate directories |

### AddDirectory(VirtualPath, bool)

VirtualNodeListConditions

AddDirectory

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)

Declaration

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)

VirtualNodeNotFoundException

`public void AddDirectory(VirtualPath path, bool createSubdirectories = false)`

VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeType.html)

Parameters

VirtualNodeTypeFilter

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeTypeFilter.html)

VirtualPath

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)



| Type  | Name             | Description                                     |
|---|------------------|---|
| VirtualPath<br>(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)  | <i>itemPath</i>  | The path of the item to add                     |
| T   | <i>data</i>      | The data for the item                           |
| bool<br>( <a href="https://learn.microsoft.com/dotnet/api/system.boolean">https://learn.microsoft.com/dotnet/api/system.boolean</a> )<br>(AkiraNetwork.VirtualStorageLibrary) | <i>overwrite</i> | Indicates whether to overwrite an existing node |

**AddLinkToDictionary(VirtualPath, VirtualPath)**

**NotifyNodeDelegate**  
Adds a new link to the link dictionary.  
(AkiraNetwork.VirtualStorageLibrary)

# PatternMatch Declaration

(AkiraNetwork.VirtualStorageLibrary)

```
VirtualCycleDetector  
public void AddLinkToDictionary(VirtualPath targetPath, VirtualPath linkPath)  
(AkiraNetwork.VirtualStorageLibrary)
```

## VirtualDirectory

## Parameters

| Type   | Description       |
|--|-------------------|
| VirtualException<br>(AkiraNetwork.VirtualStorageLibrary)                       |                   |
| VirtualPathGroup<br>(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)      | <i>targetPath</i> |
| VirtualPath<br>(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)<br>2.html | <i>linkPath</i>   |

VirtualID

## VirtualStorage Exceptions

**Type** VirtualItem  
(AkiraNetwork.VirtualStorageLibrary)  
**ArgumentException**  
VirtualItem<T>  
(<https://learn.microsoft.com/dotnet/api/AkiraNetwork.VirtualStorageLibrary>)

• The first note of each measure is a half note.

## AddNode(VirtualPath, VirtualNode, bool)

**VirtualNodeContext**  
Adds a node to the specified direction.

## VirtualNodeExtensions Declaration

```
    public void AddNode(VirtualPath nodeDirectoryPath, VirtualNode node, bool overwrite = false)
```

## VirtualNodeName (Alio-Nodes und VirtualStorages)

## Parameters

**Type** (AkiraNetwork.VirtualStorageLibrary)

VirtualPath VirtualNodeType  
(AkiraNetwork.VirtualStorage library)

(Akibanetwork.VirtualStorageEditor:VirtualNodeTypeFilter  
VirtualPathFilter)

(AkiraNetwork.VirtualStorageLibrary  
VirtualPath

VirtualPath  
(AkiraNetwork.VirtualStorageLibrary)

| Type   | Name                     | Description                                  |
|--|--------------------------|--|
| VirtualNodeNotFoundException<br>(AkiraNetwork.VirtualStorageLibrary.VirtualPathType) | <i>nodeDirectoryPath</i> | The path of the directory to add the node to |



| Type  | Name                     | Description  |
|---|--------------------------|--|
| VirtualPath<br>(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)  | <i>linkDirectoryPath</i> | The path of the directory to add the symbolic link to  |
| VirtualSymbolicLink<br>(AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink.html)<br>IVirtualWildcardMatcher   | <i>link</i>              | The VirtualSymbolicLink<br>(AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink.html)<br>to add |
| bool<br>(AkiraNetwork.VirtualStorageLibrary.NotifyNodeDelegate<br>(https://learn.microsoft.com/dotnet/api/system.boolean)<br>PatternMatch<br>(AkiraNetwork.VirtualStorageLibrary.VirtualCycleDetector | <i>overwrite</i>         | Indicates whether to overwrite an existing node  |

## ChangeDirectory(VirtualPath)

VirtualDirectory  
Changes the current working directory.  
(AkiraNetwork.VirtualStorageLibrary)

### Declaration

```
VirtualException  
(AkiraNetwork.VirtualStorageLibrary)
VirtualGroupCondition<T, TKey>
public void ChangeDirectory(VirtualPath path)
(AkiraNetwork.VirtualStorageLibrary
2.html)
```

### Parameters

| Type   | Name        | Description                           |
|--|-------------|---------------------------------------|
| VirtualItem<br>(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) | <i>path</i> | The path of the new working directory |

## ConvertToAbsolutePath(VirtualPath?, VirtualPath?)

VirtualNode  
(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeContext  
(AkiraNetwork.VirtualStorageLibrary)

### Declaration

```
VirtualNodeExtensions  
(AkiraNetwork.VirtualStorageLibrary)
public VirtualPath? ConvertToAbsolutePath(VirtualPath? relativePath, VirtualPath? basePath =
null)
(AkiraNetwork.VirtualStorageLibrary
VirtualNodeName  
(AkiraNetwork.VirtualStorageLibrary)
VirtualNodeNotFoundException  
(AkiraNetwork.VirtualStorageLibrary)
```

### Parameters

| Type   | Name                | Description   |
|--|---------------------|---|
| VirtualPath<br>(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)<br>VirtualNodeTypeFilter<br>(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) | <i>relativePath</i> | The relative path.                                    |
| VirtualPath<br>(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)   | <i>basePath</i>     | The base path. If not specified, CurrentPath is used. |

## Returns

| Type   | Description   |
|--|---|
| VirtualPath<br>(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) | The absolute path represented by VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html). |

## Exceptions

| Type   | Condition  |
|--|--|
| ArgumentException<br>(AkiraNetwork.VirtualStorageLibrary.ArgumentException<br>(https://learn.microsoft.com/dotnet/api/system.argumentexception)<br>NotifyNodeDelegate<br>(AkiraNetwork.VirtualStorageLibrary.PatternMatch) | Thrown if relativePath is null or an empty string. |

## CopyNode(VirtualPath, VirtualPath, bool, bool, List<VirtualNodeContext>?)

VirtualCycleDetector

Copies the node at the specified path.

VirtualDirectory

(AkiraNetwork.VirtualStorageLibrary)

VirtualException

```
public void CopyNode(VirtualPath sourcePath, VirtualPath destinationPath, bool overwrite = false, bool recursive = false, bool followLinks = false, List<VirtualNodeContext>? destinationContextList = null)
```

(AkiraNetwork.VirtualStorageLibrary)

2.html)

## Parameters

| Type   | Name                   | Description                          |
|--|------------------------|--------------------------------------|
| VirtualItem  |                        |                                      |
| VirtualPath<br>(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)   | sourcePath             | The source path of the copy.         |
| VirtualItem<T>   |                        |                                      |
| VirtualPath<br>(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)   | destinationPath        | The destination path of the copy.    |
| VirtualNode  |                        |                                      |
| bool<br>(https://learn.microsoft.com/dotnet/api/system.boolean)  | overwrite              | Whether to overwrite existing nodes. |
| VirtualNodeContext   |                        |                                      |
| bool<br>(https://learn.microsoft.com/dotnet/api/system.boolean)  | recursive              | Whether to copy recursively.         |
| bool<br>(https://learn.microsoft.com/dotnet/api/system.boolean)  | followLinks            | Whether to follow symbolic links.    |
| VirtualNodeName  |                        |                                      |
| List<br>(AkiraNetwork.VirtualStorageLibrary<br>(https://learn.microsoft.com/dotnet/api/system.collections.generic.list-1)) | destinationContextList | List of destination contexts.        |
| VirtualNodeNotFoundException<br>(AkiraNetwork.VirtualStorageLibrary.VirtualNodeNotFoundException)                          |                        |                                      |
| < VirtualNodeType  |                        |                                      |
| VirtualNodeContext   |                        |                                      |
| (AkiraNetwork.VirtualStorageLibrary)   |                        |                                      |
| (AkiraNetwork.VirtualStorageLibrary.VirtualNodeContext.html)   |                        |                                      |
| > VirtualNodeTypeFilter  |                        |                                      |
| (AkiraNetwork.VirtualStorageLibrary)   |                        |                                      |
| VirtualPath  |                        |                                      |

## Exceptions

(AkiraNetwork.VirtualStorageLibrary)

| Type  | Condition  |
|---|--|
| VirtualNodeNotFoundException<br>(AkiraNetwork.VirtualStorageLibrary.VirtualNodeNotFoundException.html)  | Thrown if the source node is not found.  |
| InvalidOperationException<br>(https://learn.microsoft.com/dotnet/api/system.invalidoperationexception)<br>(AkiraNetwork.VirtualStorageLibrary.InvalidOperationException.html) | Thrown if the source and destination paths are the same or if a recursive copy would result in a circular reference. |

## DirectoryExists(VirtualPath, bool)

Checks if the specified path.

PatternMatch

Declaration

```
public bool DirectoryExists(VirtualPath path, bool followLinks = false)
```

VirtualDirectory

(AkiraNetwork.VirtualStorageLibrary.VirtualDirectory.html)

Parameters

VirtualException

| Type  | Name        | Description                       |
|---|-------------|-----------------------------------|
| AkiraNetwork.VirtualStorageLibrary.VirtualPath<T, TKey><br>(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)  | path        | The directory's path.             |
| bool (https://learn.microsoft.com/dotnet/api/system.boolean)<br>(AkiraNetwork.VirtualStorageLibrary.Boolean.html) | followLinks | Whether to follow symbolic links. |

VirtualID

Returns

VirtualItem

| Type  | Description                                     |
|---|---|
| bool (https://learn.microsoft.com/dotnet/api/system.boolean)<br>(AkiraNetwork.VirtualStorageLibrary.Boolean.html) | True if the directory exists; otherwise, false. |

1.html)

VirtualNode

## ExpandPath(VirtualPath, VirtualNodeTypeFilter, bool, bool)

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

VirtualNodeContext

Expands a path that contains wildcards.

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeContext.html)

Declaration

VirtualNodeExtensions

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeExtensions.html)

```
public VirtualNodeList<Conditions> ExpandPath(VirtualPath path, VirtualNodeTypeFilter filter = VirtualNetwork.VirtualStorageLibrary.VirtualNodeTypeFilter.all, bool followLinks = true, bool resolveLinks = true)
```

VirtualNodeName

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)

Parameters

VirtualNodeNotFoundException

| Type   | Name        | Description           |
|--|-------------|-----------------------|
| AkiraNetwork.VirtualStorageLibrary.VirtualPath<T, TKey><br>(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) | path        | The path.             |
| VirtualNodeTypeFilter<br>(AkiraNetwork.VirtualStorageLibrary.VirtualNodeTypeFilter.html)                         | filter      | The node type filter. |
| AkiraNetwork.VirtualStorageLibrary.VirtualPath<T, TKey><br>(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) | VirtualPath |                       |

| Type  | Name                | Description                        |
|---|---------------------|------------------------------------|
| bool ( <a href="https://learn.microsoft.com/dotnet/api/system.boolean">https://learn.microsoft.com/dotnet/api/system.boolean</a> )      | <i>followLinks</i>  | Whether to follow symbolic links.  |
| bool ( <a href="https://learn.microsoft.com/dotnet/api/system.boolean">https://learn.microsoft.com/dotnet/api/system.boolean</a> )<br>▼ | <i>resolveLinks</i> | Whether to resolve symbolic links. |

Return Type: [AkiraNetwork.VirtualStorageLibrary](#)

| Type  | Description        |
|---|--------------------|
| IVirtualWildcardMatcher<br><a href="#">1.html</a> | The expanded paths |

Remarks: [VirtualException](#)

`followLinks` indicates whether to follow links during the recursive traversal of nodes when the terminal node of the path is a symbolic link.

`VirtualGroupCondition<T, TKey>`

`resolveLinks` indicates whether to resolve links when non-terminal nodes of the path are symbolic links.

[2.html](#)

VirtualID

## ExpandPathTree(VirtualPath, VirtualNodeTypeFilter, bool, bool)

VirtualItem

Expands the virtual path containing wildcards.

([AkiraNetwork.VirtualStorageLibrary](#))

Declaration: `public virtual IEnumerable<VirtualNodeContext> ExpandPathTree(VirtualPath path, VirtualNodeTypeFilter filter, bool followLinks = true, bool resolveLinks = true)`

([AkiraNetwork.VirtualStorageLibrary](#))

VirtualNodeContext

([AkiraNetwork.VirtualStorageLibrary](#))

| Type  | Name                | Description   |
|---|---------------------|---|
| virtualNodeExtensions   |                     |   |
| ( <a href="#">AkiraNetwork.VirtualStorageLibrary</a> )  | <i>path</i>         | The virtual path containing wildcards               |
| VirtualPath   |                     |   |
| VirtualNodeListConditions   |                     |   |
| ( <a href="#">AkiraNetwork.VirtualStorageLibrary</a> )  | <i>filter</i>       | The filter to apply for node types                  |
| VirtualNodeTypeFilter   |                     |   |
| ( <a href="#">AkiraNetwork.VirtualStorageLibrary</a> )  | <i>followLinks</i>  | A flag indicating whether to follow symbolic links  |
| bool  |                     |   |
| VirtualNodeType   |                     |   |
| ( <a href="https://learn.microsoft.com/dotnet/api/system.boolean">https://learn.microsoft.com/dotnet/api/system.boolean</a> ) | <i>resolveLinks</i> | A flag indicating whether to resolve symbolic links |
| bool  |                     |   |

[VirtualPath](#)

([AkiraNetwork.VirtualStorageLibrary](#))

## Returns

| Type   | Description  |
|--|--|
| IEnumerable<br>( <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> )<br><<br>VirtualNodeContext<br>(AkiraNetwork.VirtualStorageLibrary.VirtualNodeContext.html)<br>> IVirtualWildcardMatcher<br>(AkiraNetwork.VirtualStorageLibrary.VirtualWildcardMatcher.html)<br>NotifyNodeDelegate<br>(AkiraNetwork.VirtualStorageLibrary.NotifyNodeDelegate.html)<br>followLinks<br>(AkiraNetwork.VirtualStorageLibrary.VirtualStorageLibrary.followLinks.html)<br>PatternMatch<br>(AkiraNetwork.VirtualStorageLibrary.VirtualStorageLibrary.PatternMatch.html)<br>virtualPathIsASymbolicLink<br>(AkiraNetwork.VirtualStorageLibrary.VirtualStorageLibrary.virtualPathIsASymbolicLink.html)<br>resolvesVirtualCycles<br>(AkiraNetwork.VirtualStorageLibrary.VirtualStorageLibrary.resolvesVirtualCycles.html)<br>VirtualDirectory<br>(AkiraNetwork.VirtualStorageLibrary.VirtualDirectory.html) | An enumerable of VirtualNodeContext<br>(AkiraNetwork.VirtualStorageLibrary.VirtualNodeContext.html)<br>containing information about the expanded nodes |

## Remarks

followLinks indicates whether to follow links during the recursive traversal process when the terminal node in the virtual path is a symbolic link.  
resolvesVirtualCycles indicates whether to resolve links when nodes other than the terminal node in the virtual path are symbolic links.

## VirtualCycleDetector

VirtualCycleDetector indicates whether to resolve links when nodes other than the terminal node in the virtual path are symbolic links.

## GetDirectory(VirtualPath, bool)

(AkiraNetwork.VirtualStorageLibrary.GetDirectory.html)  
Gets the directory at the specified path.

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

## Declaration

VirtualDirectory  
public VirtualDirectory GetDirectory(VirtualPath path, bool followLinks = false)  
(AkiraNetwork.VirtualStorageLibrary.VirtualDirectory.html)

VirtualItem

## Parameters

(AkiraNetwork.VirtualStorageLibrary.VirtualStorageLibrary.html)

| Type   | Name        | Description                       |
|--|-------------|-----------------------------------|
| VirtualItem<T><br>(AkiraNetwork.VirtualStorageLibrary.VirtualItem.html)  | path        | The path of the directory.        |
| bool<br>( <a href="https://learn.microsoft.com/dotnet/api/system.boolean">https://learn.microsoft.com/dotnet/api/system.boolean</a> )<br>(AkiraNetwork.VirtualStorageLibrary.VirtualStorageLibrary.boolean.html) | followLinks | Whether to follow symbolic links. |

## VirtualNodeContext

## Returns

(AkiraNetwork.VirtualStorageLibrary.VirtualStorageLibrary.html)

## Type

VirtualNodeExtensions  
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeExtensions.html)

## Description

|   |   |
|---|---|
| VirtualDirectory<br>VirtualNodeListConditions<br>(AkiraNetwork.VirtualStorageLibrary.VirtualDirectory.html)<br>VirtualDirectoryName<br>(AkiraNetwork.VirtualStorageLibrary.VirtualDirectoryName.html) | The VirtualDirectory<br>(AkiraNetwork.VirtualStorageLibrary.VirtualDirectory.html) at the specified path. |
|---|---|

## VirtualNodeNotFoundException

## Exceptions

(AkiraNetwork.VirtualStorageLibrary.VirtualStorageLibrary.html)

## Type

VirtualNodeType  
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeType.html)

VirtualNodeNotFoundException  
(AkiraNetwork.VirtualStorageLibrary.VirtualStorageLibrary.VirtualNodeNotFoundException.html)

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

| Type   | Condition                           |
|--|-------------------------------------|
| VirtualNodeType<br>(AkiraNetwork.VirtualStorageLibrary.VirtualNodeType.html) | Thrown if the node cannot be found. |

## VirtualPath

(AkiraNetwork.VirtualStorageLibrary.VirtualStorageLibrary.VirtualPath.html)

## GetItem(VirtualPath, bool)

Gets the item at the specified path.

### Declaration

```
public VirtualItem<T> GetItem(VirtualPath path, bool followLinks = false)
```

**Parameters**  
1.html)

| Type  | Name        | Description                       |
|---|-------------|-----------------------------------|
| VirtualWildcardMatcher<br>(AkiraNetwork.VirtualStorageLibrary.VirtualPath(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)) | path        | The path of the item.             |
| bool(AkiraNetwork.VirtualStorageLibrary.Boolean)  | followLinks | Whether to follow symbolic links. |

PatternMatch

**Returns**  
AkiraNetwork.VirtualStorageLibrary.

| Type  | Description  |
|---|--|
| VirtualCycleDetector<br>(AkiraNetwork.VirtualStorageLibrary.VirtualItem<T>(AkiraNetwork.VirtualStorageLibrary.VirtualItem<T>.VirtualException)) |  |
| VirtualDirectory<br>(AkiraNetwork.VirtualStorageLibrary.VirtualItem<T>(AkiraNetwork.VirtualStorageLibrary.VirtualItem<T>.VirtualException))     | The VirtualItem<T> (AkiraNetwork.VirtualStorageLibrary.VirtualItem<T>.VirtualException) at the specified path. |

**Exceptions**  
2.html)

| Type  | Condition                           |
|---|-------------------------------------|
| VirtualID<br>(AkiraNetwork.VirtualStorageLibrary.VirtualItem<T>(AkiraNetwork.VirtualStorageLibrary.VirtualNodeNotFoundException)) | Thrown if the node cannot be found. |

**GetLinksFromDictionary(VirtualPath)**  
VirtualNode  
Retrieves all link paths associated with the specified target path from the link dictionary.  
(AkiraNetwork.VirtualStorageLibrary)

### Declaration

```
VirtualNodeContext  
(AkiraNetwork.VirtualStorageLibrary)  
VirtualNodeExtensions  
public HashSet<VirtualPath> GetLinksFromDictionary(VirtualPath targetPath)  
(AkiraNetwork.VirtualStorageLibrary)  
VirtualNodeListConditions
```

**Parameters**  
(AkiraNetwork.VirtualStorageLibrary)

| Type   | Name       | Description      |
|--|------------|------------------|
| VirtualNodeName<br>(AkiraNetwork.VirtualStorageLibrary.VirtualPath(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)) | targetPath | The target path. |

**Returns**  
VirtualNodeType

| Type  | Description          |
|---|----------------------|
| VirtualNodeTypeFilter<br>HashSet<VirtualPath>(AkiraNetwork.VirtualStorageLibrary.VirtualPath(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html))> | A set of link paths. |

## GetNode(VirtualPath, bool)

Gets the node at the specified path.

### Declaration

```
public VirtualNode GetNode(VirtualPath path, bool followLinks = false)
```

**Parameters**  
1.html

| Type  | Name               | Description                       |
|---|--------------------|-----------------------------------|
| (AkiraNetwork.VirtualStorageLibrary.VirtualPath) VirtualWildcardMatcher         | <i>path</i>        | The node's path.                  |
| bool (https://learn.microsoft.com/dotnet/api/system.boolean) NotifyNodeDelegate | <i>followLinks</i> | Whether to follow symbolic links. |
| PatternMatch  |                    |                                   |

**Returns**  
AkiraNetwork.VirtualStorageLibrary.VirtualNode

| Type  | Description  |
|---|--|
| (AkiraNetwork.VirtualStorageLibrary.VirtualNode) VirtualCycleDetector |  |
| (AkiraNetwork.VirtualStorageLibrary.VirtualNode) VirtualDirectory     | The VirtualNode  |
| (AkiraNetwork.VirtualStorageLibrary.VirtualNode) VirtualException     | (AkiraNetwork.VirtualStorageLibrary.VirtualNode.html) at the specified path. |
| (AkiraNetwork.VirtualStorageLibrary.VirtualGroupCondition<T, TKey>)   |  |

**Exceptions**  
2.html

| Type  | Condition                           |
|---|-------------------------------------|
| (AkiraNetwork.VirtualStorageLibrary.VirtualNodeNotFoundException) | Thrown if the node cannot be found. |

**GetNodeType(VirtualPath, bool)**  
VirtualPath

VirtualNode  
Gets the node type at the specified path.  
(AkiraNetwork.VirtualStorageLibrary)

### Declaration

```
public VirtualNodeType GetNodeType(VirtualPath path, bool followLinks = false)
```

(AkiraNetwork.VirtualStorageLibrary)

**Parameters**  
VirtualPath

| Type   | Name               | Description                       |
|--|--------------------|-----------------------------------|
| (AkiraNetwork.VirtualStorageLibrary.VirtualNodeName)         | <i>path</i>        | The node's path.                  |
| bool (https://learn.microsoft.com/dotnet/api/system.boolean) | <i>followLinks</i> | Whether to follow symbolic links. |
| VirtualNodeType  |                    |                                   |

**Returns**  
VirtualNodeTypeFilter

(AkiraNetwork.VirtualStorageLibrary)

VirtualPath

(AkiraNetwork.VirtualStorageLibrary)

| Type   | Description   |
|--|---|
| VirtualNodeType<br>(AkiraNetwork.VirtualStorageLibrary.VirtualNodeType.html) | The VirtualNodeType<br>(AkiraNetwork.VirtualStorageLibrary.VirtualNodeType.html) of the node. |

(AkiraNetwork.VirtualStorageLibrary)

## GetNodes(VirtualNodeTypeFilter, bool, bool)

IVirtualWildcardMatcher

Gets the nodes in the path tree starting from the current path.  
(AkiraNetwork.VirtualStorageLibrary)

Declaration

(AkiraNetwork.VirtualStorageLibrary)

PatternMatch

```
public IEnumerable<VirtualNode> GetNodes(VirtualNodeTypeFilter nodeType = VirtualNodeTypeFilter.All, bool recursive = false, bool followLinks = false)
```

VirtualCycleDetector

(AkiraNetwork.VirtualStorageLibrary)

Parameters

VirtualDirectory

| Type   | Name        | Description  |
|--|-------------|--|
| VirtualException   |             |  |
| VirtualNodeTypeFilter<br>(AkiraNetwork.VirtualStorageLibrary)          | nodeType    | The node type filter.<br>(AkiraNetwork.VirtualStorageLibrary.VirtualNodeTypeFilter.html) |
| bool<br>(https://learn.microsoft.com/dotnet/api/system.boolean.2.html) | recursive   | Whether to retrieve nodes recursively.   |
| VirtualID  |             |  |
| bool<br>(AkiraNetwork.VirtualStorageLibrary)                           | followLinks | Whether to follow symbolic links.<br>(AkiraNetwork.VirtualStorageLibrary.VirtualItem)    |

Return Value

VirtualItem<T>

| Type  | Description  |
|---|--|
| IEnumerable<VirtualNode><br>(http://AkiraNetwork.VirtualStorageLibrary/api/system.collection.generic.ienumerable-1) | The nodes in the path tree starting from the current path<br>IEnumerable<T><br>(https://learn.microsoft.com/dotnet/api/system.collections.generic.ienumerable-1) |

Remarks

VirtualNodeNotFoundException  
This method may be integrated, reorganized, or deprecated in the near future.  
(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeType

## GetNodes(VirtualPath, VirtualNodeTypeFilter, bool, bool)

VirtualNodeTypeFilter

Gets the nodes in the path tree starting from the specified path.  
(AkiraNetwork.VirtualStorageLibrary)

VirtualPath

(AkiraNetwork.VirtualStorageLibrary)

```
public IEnumerable<VirtualNode> GetNodes(VirtualPath basePath, VirtualNodeTypeFilter nodeType = VirtualNodeTypeFilter.All, bool recursive = false, bool followLinks = false)
```

## Parameters

| Type  | Name        | Description                            |
|---|-------------|--|
| VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)                     | basePath    | The base path.                         |
| VirtualNodeTypeFilter (AkiraNetwork.VirtualStorageLibrary.VirtualNodeTypeFilter.html) | nodeType    | The node type filter.                  |
| bool (https://learn.microsoft.com/dotnet/api/system.boolean)                          | recursive   | Whether to retrieve nodes recursively. |
| PatternMatch (AkiraNetwork.VirtualStorageLibrary.VirtualCycleDetector)                | followLinks | Whether to follow symbolic links.      |

## Returns

VirtualDirectory

| Type   | Description  |
|--|--|
| VirtualException<br>IEnumerable<VirtualNode> (https://learn.microsoft.com/dotnet/api/system.collections.generic.ienumerable-1) | The nodes in the path tree starting from the specified path<br>IEnumerable<T><br>(https://learn.microsoft.com/dotnet/api/system.collections.generic.ienumerable-1) |

## Remarks

(AkiraNetwork.VirtualStorageLibrary)

This method may be integrated, reorganized, or deprecated in the near future.

VirtualNode

(AkiraNetwork.VirtualStorageLibrary)

## GetNodesWithPaths(VirtualNodeTypeFilter, bool, bool)

(AkiraNetwork.VirtualStorageLibrary)

Gets the paths in the path tree starting from the current path.

VirtualNodeExtensions

(AkiraNetwork.VirtualStorageLibrary)

Declaration

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeListConditions

```
public IEnumerable<VirtualPath> GetNodesWithPaths(VirtualNodeTypeFilter nodeType = VirtualNodeTypeFilter.All, bool recursive = false, bool followLinks = false)
```

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeNotFoundException

## Parameters

| Type   | Name     | Description           |
|--|----------|-----------------------|
| VirtualNodeType<br>VirtualNodeTypeFilter (AkiraNetwork.VirtualStorageLibrary.VirtualNodeTypeFilter.html) | nodeType | The node type filter. |

VirtualPath

(AkiraNetwork.VirtualStorageLibrary)

| Type  | Name               | Description                            |
|---|--------------------|--|
| bool ( <a href="https://learn.microsoft.com/dotnet/api/system.boolean">https://learn.microsoft.com/dotnet/api/system.boolean</a> )      | <i>recursive</i>   | Whether to retrieve paths recursively. |
| bool ( <a href="https://learn.microsoft.com/dotnet/api/system.boolean">https://learn.microsoft.com/dotnet/api/system.boolean</a> )<br>▼ | <i>followLinks</i> | Whether to follow symbolic links.      |

Returns [AkiraNetwork.VirtualStorageLibrary](#)

| Type  | Description   |
|---|---|
| <a href="#">IVirtualWildcardMatcher</a>   |   |
| <a href="#">IEnumerable&lt;VirtualPath&gt;</a>  | The paths in the path tree starting from the current path   |
| <a href="#">(https://learn.microsoft.com/dotnet/api/system.collections.generic.ienumerable-1)</a> | <a href="#">IEnumerator&lt;T&gt;</a><br>( <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> ) |
| < PatternMatch  | .   |
| <a href="#">VirtualPath</a>   | .   |
| <a href="#">(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)</a>                             | .   |
| > (AkiraNetwork.VirtualStorageLibrary   |   |
| VirtualDirectory  |   |

Remarks

This method may be integrated, reorganized, or deprecated in the near future.  
([AkiraNetwork.VirtualStorageLibrary](#))

[VirtualException](#)

Gets the paths in the path tree starting from the specified path.

[VirtualPath](#)

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

[VirtualItem](#)

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

```
public IEnumerable<VirtualPath> GetNodesWithPaths(VirtualPath basePath, VirtualNodeTypeFilter nodeType = VirtualNodeTypeFilter.All, bool recursive = false, bool followLinks = false)
```

[1.html](#)

Parameters

[Node](#)

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

[VirtualNodeContext](#)

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

[VirtualNodeExtensions](#)

[VirtualNodeTypeFilter](#)

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

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

[VirtualNodeListConditions](#)

[bool](#)

[\(https://learn.microsoft.com/dotnet/api/system.boolean">https://learn.microsoft.com/dotnet/api/system.boolean](#))

[VirtualnodeName](#)

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

[bool](#)

[\(https://learn.microsoft.com/dotnet/api/system.boolean">https://learn.microsoft.com/dotnet/api/system.boolean](#))

[VirtualNodeNotFoundException](#)

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

[VirtualNodeType](#)

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

[VirtualPath](#)

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

| Type  | Description   |
|---|---|
| IEnumerable<br>( <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> )<br><<br>VirtualPath<br>(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)<br>> (AkiraNetwork.VirtualStorageLibrary<br>1.html) | The paths in the path tree starting from the specified path<br>IEnumerable<T><br>( <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> )<br>. |

**Remarks**

(AkiraNetwork.VirtualStorageLibrary  
This method may be integrated, reorganized, or deprecated in the near future.  
NotifyNodeDelegate

(AkiraNetwork.VirtualStorageLibrary  
PatternMatch  
(AkiraNetwork.VirtualStorageLibrary

## GetSymbolicLink(VirtualPath)

(AkiraNetwork.VirtualStorageLibrary

Gets VirtualSymbolicLink at the specified path.

(AkiraNetwork.VirtualStorageLibrary

**Declaration**

virtualDirectory

(AkiraNetwork.VirtualStorageLibrary

public VirtualSymbolicLink GetSymbolicLink(VirtualPath path)

(AkiraNetwork.VirtualStorageLibrary

VirtualGroupCondition<T, TKey>

(AkiraNetwork.VirtualStorageLibrary

**Parameters**

Type  
2.html)

VirtualID  
VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

(AkiraNetwork.VirtualStorageLibrary

VirtualItem  
Returns

(AkiraNetwork.VirtualStorageLibrary

Type virtualItem<T>

(AkiraNetwork.VirtualStorageLibrary

VirtualSymbolicLink  
1.html)

(AkiraNetwork.VirtualStorageLibrary.VirtualSy

mbolicLink.html)

(AkiraNetwork.VirtualStorageLibrary

VirtualNodeContext

(AkiraNetwork.VirtualStorageLibrary

VirtualNodeExtensions

This method does not have a followLinks parameter because it retrieves the symbolic link itself.

(AkiraNetwork.VirtualStorageLibrary

Exception VirtualNodeListConditions

(AkiraNetwork.VirtualStorageLibrary

Type VirtualnodeName

(AkiraNetwork.VirtualStorageLibrary

VirtualNodeNotFoundException

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeNotFoundException.html)

(AkiraNetwork.VirtualStorageLibrary

VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary

ItemExists(VirtualPath, bool)

(AkiraNetwork.VirtualStorageLibrary

Checks if an item exists at the specified path.

VirtualPath

(AkiraNetwork.VirtualStorageLibrary

Declaration

(AkiraNetwork.VirtualStorageLibrary

| Type  | Name | Description                    |
|---|------|--------------------------------|
| VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) | path | The path of the symbolic link. |

| Type                           | Description   |
|--------------------------------|---|
| VirtualSymbolicLink<br>1.html) | The<br>VirtualSymbolicLink<br>(AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink.html)<br>at the specified path. |

| Type   | Condition                           |
|--|-------------------------------------|
| VirtualNodeNotFoundException<br>(AkiraNetwork.VirtualStorageLibrary.VirtualNodeNotFoundException.html) | Thrown if the node cannot be found. |

| Type        | Description                                     |
|-------------|---|
| VirtualPath | Checks if an item exists at the specified path. |

(AkiraNetwork.VirtualStorageLibrary

Declaration

(AkiraNetwork.VirtualStorageLibrary

```
public bool ItemExists(VirtualPath path, bool followLinks = false)
```

#### Parameters

| Type   | Name        | Description                       |
|--|-------------|-----------------------------------|
| VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)    | path        | The item's path.                  |
| bool (https://learn.microsoft.com/dotnet/api/system.boolean) 1.html) | followLinks | Whether to follow symbolic links. |

IVirtualWildcardMatcher  
Returns: (AkiraNetwork.VirtualStorageLibrary)

| Type  | Description                                |
|---|--|
| NotifyNodeDelegate (AkiraNetwork.VirtualStorageLibrary) bool (https://learn.microsoft.com/dotnet/api/system.boolean) PatternMatch | True if the item exists; otherwise, false. |

(AkiraNetwork.VirtualStorageLibrary)  
VirtualCycleDetector  
(AkiraNetwork.VirtualStorageLibrary)

## MoveNode(VirtualPath, VirtualPath, bool, bool)

Move the node at the specified path to a new location.  
VirtualException

Declaration: (AkiraNetwork.VirtualStorageLibrary)

```
public void MoveNode(VirtualPath sourcePath, VirtualPath destinationPath, bool overwrite = false, bool resolveLinks = true)
```

VirtualID  
(AkiraNetwork.VirtualStorageLibrary)

Parameters: VirtualItem

| Type  | Name            | Description                          |
|---|-----------------|--------------------------------------|
| VirtualItem (AkiraNetwork.VirtualStorageLibrary)                  | sourcePath      | The source path of the node.         |
| VirtualPath (AkiraNetwork.VirtualStorageLibrary) VirtualPath.html | destinationPath | The destination path for the node.   |
| bool (AkiraNetwork.VirtualStorageLibrary) VirtualNodeContext      | overwrite       | Whether to overwrite existing nodes. |
| bool (AkiraNetwork.VirtualStorageLibrary) VirtualNodeExtensions   | resolveLinks    | Whether to resolve symbolic links.   |

(AkiraNetwork.VirtualStorageLibrary)

#### Remarks

resolves symbolic links when non-terminal nodes (from the top node to the parent of the terminal node) don't contain symbolic links.

(AkiraNetwork.VirtualStorageLibrary)  
Exceptions: VirtualNodeType

| Type  | Condition   |
|---|---|
| VirtualNodeTypeFilter<br>InvalidOperationException<br>(AkiraNetwork.VirtualStorageLibrary) VirtualPath<br>(https://learn.microsoft.com/dotnet/api/system.invalidoperationexception) | Thrown if the source and destination paths are the same, if attempting to move the root directory, or if the destination is a subdirectory of the source. |



## RemoveLinkFromDictionary(VirtualPath, VirtualPath)

Removes a link from the link dictionary.

### Declaration

```
public void RemoveLinkFromDictionary(VirtualPath targetPath, VirtualPath linkPath)
```

**Parameters**  
1.html

| Type  | Name              | Description           |
|---|-------------------|-----------------------|
| (AkiraNetwork.VirtualStorageLibrary.VirtualPath) VirtualWildcardMatcher | <i>targetPath</i> | The target path.      |
| (AkiraNetwork.VirtualStorageLibrary.VirtualPath) NotifyNodeDelegate     | <i>linkPath</i>   | The path of the link. |

VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

PatternMatch

(AkiraNetwork.VirtualStorageLibrary.VirtualPath) VirtualCycleDetector

## RemoveNode(VirtualPath, bool, bool, bool)

VirtualDirectory

Removes the node at the specified path.

(AkiraNetwork.VirtualStorageLibrary.VirtualPath) VirtualException

### Declaration

(AkiraNetwork.VirtualStorageLibrary.VirtualPath) VirtualException

```
public void RemoveNode(VirtualPath nodePath, bool recursive = false, bool followLinks = false, bool resolveLinks = true)
```

**Parameters**

(AkiraNetwork.VirtualStorageLibrary.VirtualPath) VirtualItem

| Type   | Name                | Description                       |
|--|---------------------|-----------------------------------|
| (AkiraNetwork.VirtualStorageLibrary.VirtualPath) VirtualItem<T>                    | <i>nodePath</i>     | The path of the node to remove.   |
| bool (https://learn.microsoft.com/dotnet/api/system.boolean) VirtualNode           | <i>recursive</i>    | Whether to remove recursively.    |
| bool (https://learn.microsoft.com/dotnet/api/system.boolean) VirtualNodeContext    | <i>followLinks</i>  | Whether to follow symbolic links. |
| bool (https://learn.microsoft.com/dotnet/api/system.boolean) VirtualNodeExtensions | <i>resolveLinks</i> | Whether to resolve link targets.  |

**Remarks**

(AkiraNetwork.VirtualStorageLibrary.VirtualPath) VirtualNodeListConditions

If *followLinks* is true and the terminal node in the path is a symbolic link, the path tree after resolving the path will be removed, and the symbolic link will also be removed. If false, only the symbolic link itself is removed.

(AkiraNetwork.VirtualStorageLibrary.VirtualPath) VirtualNodeName

*resolveLinks* indicates whether to resolve links when non-terminal nodes (from the top node to the parent of the terminal node) in the path are symbolic links.

(AkiraNetwork.VirtualStorageLibrary.VirtualPath) VirtualNodeException

(AkiraNetwork.VirtualStorageLibrary.VirtualPath) VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary.VirtualPath) InvalidOperationException

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

(AkiraNetwork.VirtualStorageLibrary.VirtualPath) VirtualStorageLibraryException

(AkiraNetwork.VirtualStorageLibrary.VirtualPath) VirtualStorageLibraryException

### Condition

Thrown when trying to remove the root node or a non-empty directory.

## ResolveLinkTarget(VirtualPath)

Resolves the target of the symbolic link at the specified path.

### Declaration

```
public VirtualPath ResolveLinkTarget(VirtualPath path)
```

**Parameters**  
1.html

| Type  | Name | Description                    |
|---|------|--------------------------------|
| (AkiraNetwork.VirtualStorageLibrary.VirtualPath)(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) | path | The path of the symbolic link. |

**Returns**  
PatternMatch

| Type  | Description      |
|---|------------------|
| (AkiraNetwork.VirtualStorageLibrary.VirtualPath)(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) | The target path. |

**Exceptions**  
(AkiraNetwork.VirtualStorageLibrary.VirtualException)

| Type  | Condition                           |
|---|-------------------------------------|
| (AkiraNetwork.VirtualStorageLibrary.VirtualNotFoundException)(AkiraNetwork.VirtualStorageLibrary.VirtualNotFoundException.html) | Thrown if the node cannot be found. |

2.html  
VirtualID

(AkiraNetwork.VirtualStorageLibrary.VirtualItem)

Sets (AkiraNetwork.VirtualStorageLibrary.VirtualItem) symbolic links in the specified list of link paths.

| Type  | Condition |
|---|-----------|
| (AkiraNetwork.VirtualStorageLibrary.VirtualNodeContext) |           |

**Parameters**  
VirtualNodeExtensions

| Type  | Name        | Description           |
|---|-------------|-----------------------|
| (AkiraNetwork.VirtualStorageLibrary.VirtualNodeListConditions)                | linkPathSet | A set of link paths.  |
| (https://learn.microsoft.com/dotnet/api/system.collections.generic.hashset-1) | nodeType    | The target node type. |

VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeTypeFilter)

**SetNode(VirtualPath, VirtualNode)**

Sets VirtualPath the specified path.

(AkiraNetwork.VirtualStorageLibrary)

## Declaration

```
public void SetNode(VirtualPath destinationPath, VirtualNode node)
```

## Parameters

| Type  | Name                   | Description  |
|---|------------------------|--|
| VirtualPath<br>(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)<br>VirtualWildcardMatcher  | <i>destinationPath</i> | The path where the node is set   |
| VirtualNode<br>NotifyNodeDelegate<br>(AkiraNetwork.VirtualStorageLibrary.VirtualNode.html)<br>PatternMatch<br>(AkiraNetwork.VirtualStorageLibrary.VirtualCycleDetector) | <i>node</i>            | The VirtualNode<br>(AkiraNetwork.VirtualStorageLibrary.VirtualNode.html)<br>to set |

**SetModeName(VirtualPath, VirtualNodeName, bool)**

VirtualException

**VirtualMachineName** Changes the name of a node.

[AkiraNetwork.VirtualStorageLibrary](#)

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

(AkiraN)

```
2.html)
public void SetNodeName(VirtualPath nodePath, VirtualnodeName newName, bool resolveLinks = t
VirtualID
rue).
(AkiraNetwork.VirtualStorageLibrary
```

Paro (Akira)

| Parameters   |              |   |  |
|--|--------------|---|--|
| Type   | Name         | Description                                 |  |
| VirtualItem<T><br>(AkiraNetwork.VirtualStorageLibrary)   |              |   |  |
| VirtualPath<br>(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)   | nodePath     | The path of the node to rename              |  |
| VirtualNodeName<br>VirtualNodeContext<br>(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)<br>(AkiraNetwork.VirtualStorageLibrary) | newName      | The new name                                |  |
| bool<br>VirtualNodeExtensions<br>(https://akiranetwork.virtualstorage.library/api/system.boolean)  | resolveLinks | Indicates whether to resolve symbolic links |  |

## **SymbolicLinkExists(VirtualPath)**

© 2007 Akria Network. All rights reserved.

`VirtualNodeNotFoundException`  
Checks if a symbolic link exists at the specified path.  
(AkiraNetwork.VirtualStorageLibrary)

### DeclarationNodeType

(AkiraNetwork VirtualStorage) library

(AkiraNetwork.virtual)

```
public bool SymbolicLinkExists(VirtualPath path)
```

## (AkiraNetwork.VirtualStorageLibrary)

## VirtualPath

**Parameters** (AkfNetwork.VirtualStorageLibrary)

| Type  | Name | Description               |
|---|------|---------------------------|
| VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) | path | The symbolic link's path. |

## Returns

| Type  | Description   |
|---|---|
| bool ( <a href="https://learn.microsoft.com/dotnet/api/system.boolean">https://learn.microsoft.com/dotnet/api/system.boolean</a> )<br>(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) | True if the symbolic link exists; otherwise, false. |

1.html)

IVirtualWildcardMatcher

## TryGetDirectory(VirtualPath, bool)

NotifyNodeDelegate

Tries to get the directory at the specified path.  
(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

PatternMatch

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

VirtualCycleDetector

```
public VirtualDirectory? TryGetDirectory(VirtualPath path, bool followLinks = false)
```

VirtualDirectory

Parameters  
(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

| Type  | Name        | Description                       |
|---|-------------|-----------------------------------|
| VirtualException<br>(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)   |             |                                   |
| Virtu   | path        | The path of the directory.        |
| bool ( <a href="https://learn.microsoft.com/dotnet/api/system.boolean">https://learn.microsoft.com/dotnet/api/system.boolean</a> )<br>(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) | followLinks | Whether to follow symbolic links. |

VirtualID

Returns  
(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

| Type  | Description  |
|---|--|
| VirtualItem<br>(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)                        |  |
| VirtualDirectory<br>VirtualItem<T><br>(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) | The found<br>VirtualDirectory<br>(AkiraNetwork.VirtualStorageLibrary.VirtualDirectory.html)<br>, or null if not found. |
| VirtualNode<br>(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)                        |  |
| VirtualNodeContext<br>(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)                 |  |

## TryGetItem(VirtualPath, bool)

VirtualNodeExtensions

Tries to get the item at the specified path.  
(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

VirtualNodeListConditions

Declaration  
(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

VirtualNodeName

```
public VirtualItem? TryGetItem(VirtualPath path, bool followLinks = false)
```

VirtualNodeNotFoundException

Parameters  
(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

| Type  | Name        | Description                       |
|---|-------------|-----------------------------------|
| VirtualNodeType<br>(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)  |             |                                   |
| VirtualNodeTypeFilter<br>(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)  | path        | The path of the item.             |
| bool ( <a href="https://learn.microsoft.com/dotnet/api/system.boolean">https://learn.microsoft.com/dotnet/api/system.boolean</a> )<br>(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) | followLinks | Whether to follow symbolic links. |

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

## Returns

| Type  | Description   |
|---|---|
| VirtualItem<br>(AkiraNetwork.VirtualStorageLibrary.VirtualItem-1.html)<br><T> | The found VirtualItem<T> (AkiraNetwork.VirtualStorageLibrary.VirtualItem-1.html), or null if not found. |

(AkiraNetwork.VirtualStorageLibrary-1.html)

## TryGetNode(VirtualPath, bool)

(AkiraNetwork.VirtualStorageLibrary-1.html)  
Tries to get the node at the specified path.  
NotifyNodeDelegate

### Declaration

```
public VirtualNode? TryGetNode(VirtualPath path, bool followLinks = false)
```

### Parameters

VirtualPath Directory

| Type  | Name        | Description                       |
|---|-------------|-----------------------------------|
| VirtualException  |             |                                   |
| VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)   | path        | The node's path.                  |
| VirtualGroupCondition<T, TKey><br>bool ( <a href="https://learn.microsoft.com/dotnet/api/system.boolean">https://learn.microsoft.com/dotnet/api/system.boolean</a> )<br>(AkiraNetwork.VirtualStorageLibrary-2.html) | followLinks | Whether to follow symbolic links. |

### Returns

VirtualID

| Type   | Description   |
|--|---|
| VirtualItem<br>VirtualNode<br>(AkiraNetwork.VirtualStorageLibrary.VirtualNode.html)<br>VirtualItem<T><br>VirtualNode.html<br>(AkiraNetwork.VirtualStorageLibrary-1.html) | The found VirtualNode<br>(AkiraNetwork.VirtualStorageLibrary.VirtualNode.html), or null if not found. |

VirtualNode

(AkiraNetwork.VirtualStorageLibrary-1.html)

## TryGetSymbolicLink(VirtualPath)

(AkiraNetwork.VirtualStorageLibrary-1.html)  
Tries to get the symbolic link at the specified path.  
VirtualNodeExtension

### Declaration

VirtualNodeListConditions

```
public VirtualSymbolicLink? TryGetSymbolicLink(VirtualPath path)
```

(AkiraNetwork.VirtualStorageLibrary-1.html)

### Parameters

VirtualNodeNotFoundException  
(AkiraNetwork.VirtualStorageLibrary-1.html)

| Type   | Name | Description |
|--|------|-------------|
| VirtualNodeType<br>(AkiraNetwork.VirtualStorageLibrary-1.html) |      |             |

VirtualNodeFilter  
(AkiraNetwork.VirtualStorageLibrary-1.html)

VirtualPath  
(AkiraNetwork.VirtualStorageLibrary-1.html)

| Type   | Description   |
|--|---|
| VirtualSymbolicLink<br>(AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink.html) | The found VirtualSymbolicLink<br>(AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink.html), or null if not found. |

## Remarks

This method does not have a followLinks parameter because it retrieves the symbolic link itself.  
(AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink.html)

IVirtualWildcardMatcher

(AkiraNetwork.VirtualStorageLibrary.NotifyNodeDelegate)

Tries to resolve the target of a symbolic link at the specified path.

PatternMatch

Declaration  
(AkiraNetwork.VirtualStorageLibrary)

VirtualCycleDetector

public VirtualPath TryResolveLinkTarget(VirtualPath path)

VirtualDirectory

(AkiraNetwork.VirtualStorageLibrary)

VirtualException

Type  
(AkiraNetwork.VirtualStorageLibrary)

| Type  | Name | Description                               |
|---|------|---|
| VirtualGroupCondition<T, TKey><br>VirtualPath<br>(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)<br>(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) | path | The path of the symbolic link.<br>2.html) |

## Returns

VirtualID

Type  
(AkiraNetwork.VirtualStorageLibrary)

| Type   | Description   |
|--|---|
| VirtualItem<br>VirtualPath<br>(AkiraNetwork.VirtualStorageLibrary)<br>(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)<br>VirtualItem<T><br>(AkiraNetwork.VirtualStorageLibrary)<br>1.html) | The target path, or null if the node cannot be found. |

## UpdateAllTargetNodeTypesInDictionary()

(AkiraNetwork.VirtualStorageLibrary)

Updates all target node types in the link dictionary.

VirtualNodeContext  
(AkiraNetwork.VirtualStorageLibrary)

Declaration  
VirtualNodeExtensions

(AkiraNetwork.VirtualStorageLibrary)

public void UpdateAllTargetNodeTypesInDictionary()  
VirtualNodeListConditions

(AkiraNetwork.VirtualStorageLibrary)

VirtualnodeName

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeNotNotFoundException

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeType

Declaration  
(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeTypeFilter

public void UpdateAllTargetNodeTypesInDictionary(VirtualPath directoryPath, VirtualDirectory newDirectory)

VirtualPath

(AkiraNetwork.VirtualStorageLibrary)

## Parameters

| Type   |  | Name                 | Description                         |
|--|--|----------------------|-------------------------------------|
| VirtualPath<br>(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)           |  | <i>directoryPath</i> | The path of the directory to update |
| VirtualDirectory<br>(AkiraNetwork.VirtualStorageLibrary.VirtualDirectory.html) |  | <i>newDirectory</i>  | The new data for the directory      |

**UpdateItem(VirtualPath, VirtualItem<T>)**

(AkiraNetwork.VirtualStorageLibrary)

## Notify Node Delegate

(AkiraNetwork VirtualStorage) library

## (AkkaNetwork) Declaration

## Patternmatch

(AkiraNetwork.VirtualMachineManager)

`pvVirtualCycleDetectorItem(VirtualPa`

(AkiraNetwork.Virtua

## VirtualDirectory

## Parameters (AkiraN)

Type virtualException

**UpdateLinkInDictionary(VirtualPath, VirtualPath)**

- `VirtualItem<T>` (AlienNet and VirtualStorage libraries)

# Updates the

1.html)

## Declaration VirtualNode

(AkiraNetwork.VirtualStorageLibrary)

Virtual Node Context

(AkiraNetwork VirtualStorage Library)

(AkiraN  
Visitors

## VirtualNodeExtensions Parameters

| Parameter  | Type  | Name                 | Description           |
|--|---|----------------------|-----------------------|
| (AkiraNetwork.VirtualStorageLibrary.VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)) | VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) | <i>linkPath</i>      | The path of the link. |
| (AkiraNetwork.VirtualStorageLibrary.VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)) | VirtualnodeName   |                      |                       |
| (AkiraNetwork.VirtualStorageLibrary.VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)) | <i>newTargetPath</i>  | The new target path. |                       |

**UpdateLinksToTarget(VirtualPath, VirtualPath)**

- [AkiraNetwork.VirtualStorageLibrary](#)

Updates the **Virtual**

(AkiraNetwork.VirtualStorageLibrary)

## DeclarationPath

(AkiraNetwork.VirtualStorageLibrary)

```
public void UpdateLinksToTarget(VirtualPath oldTargetPath, VirtualPath newTargetPath)
```

#### Parameters

| Type  | Name          | Description          |
|---|---------------|----------------------|
| VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) | oldTargetPath | The old target path. |
| VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) | newTargetPath | The new target path. |

IVirtualWildcardMatcher  
(AkiraNetwork.VirtualStorageLibrary.PatternMatcher)

### UpdateSymbolicLink(VirtualPath, VirtualSymbolicLink)

(AkiraNetwork.VirtualStorageLibrary.VirtualCycleDetector)

Updates a symbolic link.

Declaration  
VirtualCycleDetector  
(AkiraNetwork.VirtualStorageLibrary.VirtualDirectory)

```
public void UpdateSymbolicLink(VirtualPath linkPath, VirtualSymbolicLink newLink)
```

(AkiraNetwork.VirtualStorageLibrary.VirtualException)

#### Parameters

| Type  | Name     | Description                             |
|---|----------|---|
| VirtualGroupCondition<T, TKey>                                    |          |   |
| VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) | linkPath | The path of the symbolic link to update |

VirtualID  
VirtualSymbolicLink  
(AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink.html)

VirtualItem  
(AkiraNetwork.VirtualStorageLibrary.VirtualItem)

(AkiraNetwork.VirtualStorageLibrary.VirtualList)

VirtualItem<T>  
(AkiraNetwork.VirtualStorageLibrary.VirtualList)

### UpdateTargetNodeTypesInDictionary(VirtualPath)

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeContext)

Declaration  
VirtualNodeExtensions  
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeExtensions)

```
public void UpdateTargetNodeTypesInDictionary(VirtualPath targetPath)
```

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeListConditions)

Parameters  
VirtualnodeName  
(AkiraNetwork.VirtualStorageLibrary.VirtualnodeName)

VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

VirtualNodeNotFoundException  
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeNotFoundException)

VirtualNodeType  
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeType)

VirtualNodeTypeFilter  
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeTypeFilter)

VirtualPath  
(AkiraNetwork.VirtualStorageLibrary.VirtualPath)

Traverses the path to the target node specified by the virtual path.

VirtualPath  
(AkiraNetwork.VirtualStorageLibrary.VirtualPath)

Declaration  
VirtualStorageLibrary

```

public VirtualNodeContext WalkPathToTarget(VirtualPath targetPath, NotifyNodeDelegate? notifyNode = null, ActionNodeDelegate? actionNode = null, bool followLinks = true, bool exceptionEnabled = true)

```

#### Parameters

| Type  | Name             | Description   |
|---|------------------|---|
| (AkiraNetwork.VirtualStorageLibrary.VirtualPath)        | targetPath       | The target virtual path   |
| (AkiraNetwork.VirtualStorageLibrary.NotifyNodeDelegate) | notifyNode       | A delegate to notify information about the node when each node is reached   |
| (AkiraNetwork.VirtualStorageLibrary.ActionNodeDelegate) | actionNode       | A delegate to perform actions when each node is reached   |
| bool  | followLinks      | A flag indicating whether to follow symbolic links  |
| bool  | exceptionEnabled | A flag indicating whether to enable exceptions. If true, exceptions are thrown; if false, the Node in VirtualNodeContext (AkiraNetwork.VirtualStorageLibrary.VirtualNodeContext.html) will be returned as null. |

#### Returns

| Type   | Description   |
|--|---|
| (AkiraNetwork.VirtualStorageLibrary.VirtualNodeContext)        | An instance of VirtualNodeContext   |
| (AkiraNetwork.VirtualStorageLibrary.VirtualNodeExtension)      | (AkiraNetwork.VirtualStorageLibrary.VirtualNodeContext.html) containing information about the target node |
| (AkiraNetwork.VirtualStorageLibrary.VirtualNodeListConditions) |   |
| (AkiraNetwork.VirtualStorageLibrary.VirtualNodeName)           |   |
| (AkiraNetwork.VirtualStorageLibrary.VirtualNodeType)           | Traverses the path tree starting from the specified path.   |

## WalkPathTree(VirtualPath, VirtualNodeTypeFilter, bool, bool)

(AkiraNetwork.VirtualStorageLibrary.Traverses the path tree starting from the specified path.)

Declaration  
 AkiraNetwork.VirtualStorageLibrary  
 VirtualNodeTypeFilter  
 AkiraNetwork.VirtualStorageLibrary  
 VirtualPath  
 AkiraNetwork.VirtualStorageLibrary

```
public IEnumerable<VirtualNodeContext> WalkPathTree(VirtualPath basePath, VirtualNodeTypeFilter filter = VirtualNodeTypeFilter.All, bool recursive = true, bool followLinks = true, bool resolveLinks = true)
```

## Parameters

| Type   |  | Name         | Description   |
|--|--|--------------|---|
| (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)                              |  | basePath     | The base virtual path                               |
| (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)                              |  |              |   |
| (AkiraNetwork.VirtualStorageLibrary.VirtualNodeTypeFilter.NotifyNodeDelegate.html) |  | filter       | The filter to apply for node types                  |
| (AkiraNetwork.VirtualStorageLibrary.VirtualNodeTypeFilter.html)                    |  |              |   |
| PatternMatch   |  |              |   |
| bool(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)                          |  | recursive    | A flag indicating whether to traverse recursively   |
| bool(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)                          |  |              |   |
| bool(AkiraNetwork.VirtualStorageLibrary.VirtualDirectory.html)                     |  | followLinks  | A flag indicating whether to follow symbolic links  |
| bool(AkiraNetwork.VirtualStorageLibrary.VirtualException.html)                     |  |              |   |
| bool(AkiraNetwork.VirtualStorageLibrary.VirtualException.html)                     |  | resolveLinks | A flag indicating whether to resolve symbolic links |
| bool(AkiraNetwork.VirtualStorageLibrary.VirtualException.html)                     |  |              |   |

Retur(Ak)

| Returns  | Description   |
|--|---|
| <pre>2.html) Type VirtualID AkiraNetwork.VirtualStorageLibrary.IEnumerable&lt;VirtualItem&gt; (https://learn.microsoft.com/dotnet/api/system.collections.generic IEnumerable) (AkiraNetwork.VirtualStorageLibrary.IEnumerable&lt;VirtualItem&gt;) &lt; T&gt; AkiraNetwork.VirtualStorageLibrary.VirtualNodeContext (AkiraNetwork.VirtualStorageLibrary.VirtualNodeContext) htmlVirtualNode &gt; (AkiraNetwork.VirtualStorageLibrary.</pre> | <p>An enumerable of <code>VirtualNodeContext</code> (AkiraNetwork.VirtualStorageLibrary.VirtualNodeContext.html) containing information about the traversed nodes</p> |

Remarks (Akira)

**Remarks**  
VirtualNodeExtensions  
followLinks indicates whether to follow links during the recursive traversal process when the terminal node in the (AkiraNetwork.VirtualStorageLibrary) virtual path is a symbolic link.  
VirtualNodeListConditions

`resolveVirtualPath` links when nodes other than the terminal node in the virtual path are symbolic. `NodeName`

# Extension Methods

#### Virtueller Virtual-Nodetypenkonzept in Tableau

(A kind of a Network Virtual Structure library)

<http://www.akira-network.com/virtuallab.html>

Virtueller Netzwerkknoten Virtuelle Speicherbibliothek DebugText (T)

(AkiraMitsuiPath\VirtualStorage\library\Utilities\VirtualTextForm

(AkiraNetworkVirtualStorageLibrary. Utilities. VirtualExt4OnMatter. htm)»AkiraNetwork\_VirtualStorageLibrary\_Utility

## Virtueller Netzwerk-VirtualStorage-Server

```
s_VirtualTextFormatter_GenerateSingleTableDebugText_1__0_)
VirtualTextFormatter.GenerateTreeDebugText<T>(VirtualStorage<T>, VirtualPath, bool, bool)
(AkiraNetwork.VirtualStorageLibrary.Utilities.VirtualTextFormatter.html#AkiraNetwork_VirtualStorageLibrary_Utilitie
s_VirtualTextFormatter_GenerateTreeDebugText_1_AkiraNetwork_VirtualStorageLibrary_VirtualStorage__0_Akira
Network_VirtualStorageLibrary_VirtualPath_System_Boolean_System_Boolean_)
```

▼

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

# Class VirtualSymbolicLink

Represents a virtual symbolic link. A virtual symbolic link provides a virtual reference to other nodes (directories, items, files, etc.) within the virtual storage library. 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.  
(AkiraNetwork.VirtualStorageLibrary)

NotifyNodeDelegate

## Inheritance

(AkiraNetwork.VirtualStorageLibrary)

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

## Implements

VirtualDirectory  
IVirtualDeepCloneable (AkiraNetwork.VirtualStorageLibrary.IVirtualDeepCloneable-1.html) <VirtualNode  
(AkiraNetwork.VirtualStorageLibrary.VirtualNode.html)>  
VirtualException

## Inherited Members

VirtualGroupCondition<T, TKey>  
VirtualNode.Name  
(AkiraNetwork.VirtualStorageLibrary)  
(AkiraNetwork.VirtualStorageLibrary.VirtualNode.html#AkiraNetwork\_VirtualStorageLibrary\_VirtualNode\_Name)  
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)  
VirtualNode.VID  
VirtualNode  
VirtualNode.1.html  
(AkiraNetwork.VirtualStorageLibrary.VirtualNode.html#AkiraNetwork\_VirtualStorageLibrary\_VirtualNode\_VID)  
VirtualNode.IsReferencedInStorage  
VirtualNode  
(AkiraNetwork.VirtualStorageLibrary.VirtualNode.html#AkiraNetwork\_VirtualStorageLibrary\_VirtualNode\_IsReferen  
cedInStorage)  
VirtualNodeContext  
object.Equals(object) ([https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals\(system-  
object\)\)  
object.Equals\(object, object\) \(\[https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-  
equals\\(system-object,-system-object\\)\\)  
VirtualNodeListConditions  
object.GetHashCode\\(\\) \\(<https://learn.microsoft.com/dotnet/api/system.object.gethashcode>\\)  
object.GetType\\(\\) \\(<https://learn.microsoft.com/dotnet/api/system.object.gettype>\\)  
object.MemberwiseClone\\(\\) \\(<https://learn.microsoft.com/dotnet/api/system.object.memberwiseclone>\\)  
VirtualNodeNotFoundException  
object.ReferenceEquals\\(object, object\\) \\(<https://learn.microsoft.com/dotnet/api/system.object.referenceeequals>\\)  
\\(AkiraNetwork.VirtualStorageLibrary\\)\]\(https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-<br/>equals\(system-object,-system-object\)\)](https://learn.microsoft.com/dotnet/api/system.object.equals#system-object-equals(system-<br/>object))

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

Assembly: AkiraNetwork.dll

## Syntax

VirtualNodeTypeFilter  
(AkiraNetwork.VirtualStorageLibrary)  
VirtualPath  
(AkiraNetwork.VirtualStorageLibrary)

```
public class VirtualSymbolicLink : VirtualNode, IVirtualDeepCloneable<VirtualNode>
```

## Constructors

### **VirtualSymbolicLink()**

1.html)

Initializes a new instance of the VirtualSymbolicLink (AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink.html) class. The node name is automatically generated using the VirtualNodeName.GenerateNodeName method.

#### Declaration

(AkiraNetwork.VirtualStorageLibrary)

PatternMatch

```
public VirtualSymbolicLink()
(AkiraNetwork.VirtualStorageLibrary)
```

VirtualCycleDetector

(AkiraNetwork.VirtualStorageLibrary)

### **VirtualSymbolicLink(VirtualNodeName)**

(AkiraNetwork.VirtualStorageLibrary)

Initializes a new instance of the VirtualSymbolicLink (AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink.html) class with the specified VirtualNodeName.

VirtualGroupCondition<T, TKey>

#### Declaration

(AkiraNetwork.VirtualStorageLibrary)

2.html)

```
public VirtualSymbolicLink(VirtualNodeName name)
VirtualItem
```

(AkiraNetwork.VirtualStorageLibrary)

#### Parameters

(AkiraNetwork.VirtualStorageLibrary)

##### Type

VirtualItem<T>

##### Name

Description

(AkiraNetwork.VirtualStorageLibrary)

name

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)

The name of the symbolic link.

VirtualNode

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeContext

### **VirtualSymbolicLink(VirtualNodeName, VirtualPath?)**

VirtualNodeExtensions

Initializes a new instance of the VirtualSymbolicLink (AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink.html) class with the specified name and target path.

VirtualNodeListConditions

#### Declaration

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeName

```
(AkiraNetwork.VirtualStorageLibrary)
public VirtualSymbolicLink(VirtualNodeName name, VirtualPath? targetPath)
```

VirtualNodeNotFoundException

(AkiraNetwork.VirtualStorageLibrary)

#### Parameters

(AkiraNetwork.VirtualStorageLibrary)

##### Type

VirtualNodeTypeFilter

##### Name

Description

(AkiraNetwork.VirtualStorageLibrary)

name

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)

The name of the symbolic link.

(AkiraNetwork.VirtualStorageLibrary)

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

Declaration

```
public class VirtualSymbolicLink {
    public VirtualSymbolicLink(VirtualNodeName name, VirtualPath? targetPath, DateTime createdDate, DateTime updatedDate)
    : base(name, targetPath, createdDate, updatedDate) {
        // ...
    }
}
```

(AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink)

VirtualCycleDetector

Parameters

(AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink)

| Type  | Name               | Description                           |
|---|--------------------|---------------------------------------|
| VirtualDirectory  |                    |                                       |
| (AkiraNetwork.VirtualStorageLibrary.VirtualDirectory)             |                    |                                       |
| VirtualNodeName   | <i>name</i>        | The name of the symbolic link.        |
| (AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)         |                    |                                       |
| VirtualGroupCondition<T, TKey>                                    |                    |                                       |
| VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) | <i>targetPath</i>  | The target path of the symbolic link. |
| (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)             |                    |                                       |
| DateTime  | <i>createdDate</i> | The creation date.                    |
| VIRTUALID   |                    |                                       |
| (https://learn.microsoft.com/dotnet/api/system.datetime)          |                    |                                       |
| DateTime  | <i>updatedDate</i> | The update date.                      |
| VIRTUALITEM   |                    |                                       |
| (https://learn.microsoft.com/dotnet/api/system.datetime)          |                    |                                       |
| VirtualItem<T>  |                    |                                       |
| (AkiraNetwork.VirtualStorageLibrary.VirtualItem.html)             |                    |                                       |
| VirtualNodeContext  |                    |                                       |
| (AkiraNetwork.VirtualStorageLibrary.VirtualNodeContext)           |                    |                                       |
| VirtualNodeExtensions   |                    |                                       |
| (AkiraNetwork.VirtualStorageLibrary.VirtualNodeExtensions)        |                    |                                       |

## Properties

(AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink)

VirtualNodeContext

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeContext)

VirtualNodeExtensions

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeExtensions)

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

VirtualNodeListConditions

Declaration

```
public class VirtualSymbolicLink : VirtualNode {
    public NodeType { get; }
}
```

(AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink)

VirtualNodeNotFoundException

Property Value

VirtualNodeType

| Type   | Description |
|--|-------------|
| (AkiraNetwork.VirtualStorageLibrary.VirtualNodeType) |             |

| Type   | Description |
|--|-------------|
| (AkiraNetwork.VirtualStorageLibrary.VirtualNodeTypeFilter) |             |

| Type   | Description |
|--|-------------|
| (AkiraNetwork.VirtualStorageLibrary.VirtualNodeTypeFilter) |             |

VirtualPath

Overrides

(AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink)

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  
1.html)

IVirtualWildcardMatcher  
public VirtualNodeType TargetNodeType { get; set; }

NotifyNodeDelegate

(AkiraNetwork.VirtualStorageLibrary  
Property Value)

PatternMatch

Type  
(AkiraNetwork.VirtualStorageLibrary  
VirtualNodeType.html)

VirtualGroupDetector  
(AkiraNetwork.VirtualStorageLibrary.VIRTUALNODETYPE.html)

VirtualDirectory

(AkiraNetwork.VirtualStorageLibrary  
VirtualException)

## TargetPath

Gets or sets the target path of the symbolic link.  
VirtualGroupCondition<T, TKey>

(AkiraNetwork.VirtualStorageLibrary  
Declaration  
2.html)

VirtualPath  
public VirtualPath? TargetPath { get; set; }

(AkiraNetwork.VirtualStorageLibrary  
VirtualItem)

Property Value  
(AkiraNetwork.VirtualStorageLibrary  
VirtualItem<T>)

Type  
(AkiraNetwork.VirtualStorageLibrary  
VirtualPath.html)

VirtualNode

(AkiraNetwork.VirtualStorageLibrary  
VirtualNodeContext)

## Methods

VirtualNodeExtensions

(AkiraNetwork.VirtualStorageLibrary  
DeepClone(bool))

Creates a deep clone of the symbolic link.  
VirtualNodeName

(AkiraNetwork.VirtualStorageLibrary  
Declaration)

VirtualNodeNotFoundException

(AkiraNetwork.VirtualStorageLibrary  
public override VirtualNode DeepClone(bool recursive = false)  
VirtualNodeType)

(AkiraNetwork.VirtualStorageLibrary  
VirtualNodeTypeFilter)

(AkiraNetwork.VirtualStorageLibrary  
VirtualPath)

(AkiraNetwork.VirtualStorageLibrary  
VirtualPathFilter)

## 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> )<br>( <a href="#">AkiraNetwork.VirtualStorageLibrary.VirtualNode</a> ) | A deep clone of the symbolic link. |

1.html)

**Overrides**  
[VirtualWildcardMatcher](#)  
[VirtualNode.DeepClone\(bool\)](#)  
[\(AkiraNetwork.VirtualStorageLibrary.VirtualNode\)](#)  
[\(AkiraNetwork.VirtualStorageLibrary.VirtualNode.html#AkiraNetwork\\_VirtualStorageLibrary\\_VirtualNode\\_DeepClone\\_System\\_Boolean\)](#)  
[\(AkiraNetwork.VirtualStorageLibrary.VirtualNode\)](#)

PatternMatch

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

**Tostring()**  
[VirtualCycleDetector](#)

Returns a string representation of the symbolic link.

VirtualDirectory

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

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

VirtualException

**Properties**  
[\(AkiraNetwork.VirtualStorageLibrary.VirtualNode\)](#)

VirtualGroupCondition<T, TKey>

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

**Returns**

2.html)

| Type  | Description                                   |
|---|---|
| virtualID<br>( <a href="#">AkiraNetwork.VirtualStorageLibrary.VirtualItem</a> ) | A string representation of the symbolic link. |

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

**Overrides**  
[VirtualItem<T>](#)

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

1.html)

VirtualNode

**Update(VirtualNode)**  
[\(AkiraNetwork.VirtualStorageLibrary.VirtualNode\)](#)

VirtualNodeContext

Updates the current symbolic link with the data from the specified node.

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

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

VirtualStorageLibrary

**Properties**  
[VirtualNodeListConditionUpdate\(VirtualNode node\)](#)

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

VirtualnodeName

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

**Parameters**  
[Type](#)

virtualNodeNotFoundException

| Type   | Name | Description                     |
|--|------|---------------------------------|
| virtualNodeNotFoundException<br>( <a href="#">AkiraNetwork.VirtualStorageLibrary.VirtualNode</a> ) | node | The node to use for the update. |

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

**Overrides**  
[VirtualNodeTypeFilter](#)

VirtualNodeUpdate(VirtualNode)

([AkiraNetwork.VirtualStorageLibrary.VirtualNode.html#AkiraNetwork\\_VirtualStorageLibrary\\_VirtualNode\\_Update\\_AkiraNetwork\\_VirtualStorageLibrary\\_VirtualNode\\_](#))

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

## Exceptions

| Type   | Condition  |
|--|--|
| ArgumentException<br>( <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(VirtualNodeName)**

Implicitly converts the specified node name to a VirtualSymbolicLink

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

([AkiraNetwork.VirtualStorageLibrary](#))

**Declaration:**

```
virtual static implicit operator VirtualSymbolicLink(VirtualNodeName nodeName)
```

([AkiraNetwork.VirtualStorageLibrary](#))

VirtualException

([AkiraNetwork.VirtualStorageLibrary](#))

| Type   | Name     | Description               |
|--|----------|---------------------------|
| VirtualNodeName<br>( <a href="#">VirtualNodeName.html</a> )<br>( <a href="#">AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html</a> ) | nodeName | The node name to convert. |

**Returns:**

VirtualItem

| Type   | Description |
|--|-------------|
| VirtualItem<T><br>VirtualSymbolicLink ( <a href="#">AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink.html</a> )<br>( <a href="#">AkiraNetwork.VirtualStorageLibrary</a> )<br>1.html) |             |

VirtualNode

([AkiraNetwork.VirtualStorageLibrary](#))

### **implicit operator VirtualSymbolicLink(VirtualPath?)**

VirtualNodeContext

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.

VirtualNodeListConditions

([AkiraNetwork.VirtualStorageLibrary](#))

VirtualNodeType

([AkiraNetwork.VirtualStorageLibrary](#))

VirtualPathTypeFilter

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

([AkiraNetwork.VirtualStorageLibrary](#))

VirtualPath

([AkiraNetwork.VirtualStorageLibrary](#))

| Type   | Name       | Description                 |
|--|------------|-----------------------------|
| VirtualPath<br>( <a href="#">AkiraNetwork.VirtualStorageLibrary.VirtualPath.html</a> ) | targetPath | The target path to convert. |

**Returns:**

VirtualPath

([AkiraNetwork.VirtualStorageLibrary](#))

| Type  | Description |
|---|-------------|
| VirtualSymbolicLink (AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink.html) |             |

## implicit operator VirtualSymbolicLink(string)

Implicitly converts the specified node name to a VirtualSymbolicLink  
 (AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink.html).  
 (AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink.html).

### Declaration

```
(AkiraNetwork.VirtualStorageLibrary)
public static implicit operator VirtualSymbolicLink(string nodeName)
(AkiraNetwork.VirtualStorageLibrary)
```

#### PatternMatch

### Parameters

| Type                 | Name | Description |
|----------------------|------|-------------|
| VirtualCycleDetector |      |             |

| Type                                 | Name | Description |
|--------------------------------------|------|-------------|
| (AkiraNetwork.VirtualStorageLibrary) |      |             |

| Type   | Name     | Description               |
|--|----------|---------------------------|
| 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 convert. |

| Type                                 | Name | Description |
|--------------------------------------|------|-------------|
| (AkiraNetwork.VirtualStorageLibrary) |      |             |

### Returns

| Type             | Description |
|------------------|-------------|
| VirtualException |             |

| Type                                 | Description |
|--------------------------------------|-------------|
| (AkiraNetwork.VirtualStorageLibrary) |             |

| Type                           | Description |
|--------------------------------|-------------|
| VirtualGroupCondition<T, TKey> |             |

| Type  | Description |
|---|-------------|
| VirtualSymbolicLink (AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink.html) |             |

| Type   | Description |
|--------|-------------|
| 2.html |             |

| Type      | Description |
|-----------|-------------|
| VirtualID |             |

| Type                                 | Description |
|--------------------------------------|-------------|
| (AkiraNetwork.VirtualStorageLibrary) |             |

## implicit operator VirtualSymbolicLink((VirtualNodeName nodeName, VirtualPath? targetPath))

| Type           | Description |
|----------------|-------------|
| VirtualItem<T> |             |

| Type                | Description |
|---------------------|-------------|
| VirtualSymbolicLink |             |

| Type  | Description |
|---|-------------|
| (AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink.html) |             |

| Type   | Description |
|--------|-------------|
| 1.html |             |

### Declaration

| Type        | Description |
|-------------|-------------|
| VirtualNode |             |

| Type                                 | Description |
|--------------------------------------|-------------|
| (AkiraNetwork.VirtualStorageLibrary) |             |

| Type               | Description |
|--------------------|-------------|
| VirtualNodeContext |             |

| Type  | Description |
|---|-------------|
| VirtualSymbolicLink((VirtualNodeName nodeName, VirtualPath? |             |

| Type         | Description |
|--------------|-------------|
| targetPath)) |             |

| Type                                 | Description |
|--------------------------------------|-------------|
| (AkiraNetwork.VirtualStorageLibrary) |             |

| Type                  | Description |
|-----------------------|-------------|
| VirtualNodeExtensions |             |

| Type                                 | Description |
|--------------------------------------|-------------|
| (AkiraNetwork.VirtualStorageLibrary) |             |

### Parameters

| Type                      | Description |
|---------------------------|-------------|
| VirtualNodeListConditions |             |

| Type                               | Description |
|------------------------------------|-------------|
| AkiraNetwork.VirtualStorageLibrary |             |

| Type            | Description |
|-----------------|-------------|
| VirtualNodeName |             |

| Type   | Description |
|--|-------------|
| (VirtualNodeName (AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html) |             |

| Type                                 | Description |
|--------------------------------------|-------------|
| (AkiraNetwork.VirtualStorageLibrary) |             |

| Type   | Description |
|--|-------------|
| nodeName ( <a href="https://learn.microsoft.com/dotnet/api/system.valuetuple">https://learn.microsoft.com/dotnet/api/system.valuetuple</a> - |             |

| Type                         | Description |
|------------------------------|-------------|
| VirtualNodeNotFoundException |             |

| Type   | Description |
|--|-------------|
| akiranetwork.virtualstoragelibrary.virtualnodename,akiranetwork.virtualstoragelibrary.virt |             |

| Type                                 | Description |
|--------------------------------------|-------------|
| (AkiraNetwork.VirtualStorageLibrary) |             |

| Type        | Description |
|-------------|-------------|
| VirtualPath |             |

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

| Type   | Description |
|--|-------------|
| targetPath ( <a href="https://learn.microsoft.com/dotnet/api/system.valuetuple">https://learn.microsoft.com/dotnet/api/system.valuetuple</a> - |             |

| Type              | Description |
|-------------------|-------------|
| VirtualNodeFilter |             |

| Type   | Description |
|--|-------------|
| akiranetwork.virtualstoragelibrary.virtualnodename,akiranetwork.virtualstoragelibrary.virt |             |

| Type                                 | Description |
|--------------------------------------|-------------|
| (AkiraNetwork.VirtualStorageLibrary) |             |

| Type        | Description |
|-------------|-------------|
| VirtualPath |             |

| Type                                 | Description |
|--------------------------------------|-------------|
| (AkiraNetwork.VirtualStorageLibrary) |             |

| Type | Description |
|------|-------------|
| )    |             |

| Type                                 | Description |
|--------------------------------------|-------------|
| (AkiraNetwork.VirtualStorageLibrary) |             |

## Returns

| Type  | Description |
|---|-------------|
| VirtualSymbolicLink (AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink.html) |             |

## Implements

IVirtualDeepCloneable (AkiraNetwork.VirtualStorageLibrary.IVirtualDeepCloneable-1.html)  
IVirtualWildcardMatcher  
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeExtensions.html#AkiraNetwork\_VirtualStorageLibrary\_VirtualNodeExtensions\_ResolveNodeType\_AkiraNetwork\_VirtualStorageLibrary\_VirtualNode\_PatternMatch)  
VirtualNodeDelegate  
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeExtensions.html#AkiraNetwork\_VirtualStorageLibrary\_VirtualNodeExtensions\_ResolveNodeType\_AkiraNetwork\_VirtualStorageLibrary\_VirtualNode\_)  
VirtualTextFormatter.GenerateSingleTableDebugText <T>(T)  
VirtualCycleDetector  
(AkiraNetwork.VirtualStorageLibrary.Utilities.VirtualTextFormatter.html#AkiraNetwork\_VirtualStorageLibrary\_Utils\_VirtualTextFormatter\_GenerateSingleTableDebugText\_1\_\_0\_)  
VirtualDirectory  
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeExtensions.html#AkiraNetwork\_VirtualStorageLibrary\_VirtualNodeExtensions\_GenerateSingleTableDebugText\_1\_\_0\_)  
VirtualException  
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeExtensions.html#AkiraNetwork\_VirtualStorageLibrary\_VirtualNodeExtensions\_GenerateSingleTableDebugText\_1\_\_0\_)  
VirtualGroupCondition <T, TKey>  
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeExtensions.html#AkiraNetwork\_VirtualStorageLibrary\_VirtualNodeExtensions\_GenerateSingleTableDebugText\_1\_\_0\_)  
VirtualID  
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeExtensions.html#AkiraNetwork\_VirtualStorageLibrary\_VirtualNodeExtensions\_GenerateSingleTableDebugText\_1\_\_0\_)  
VirtualItem  
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeExtensions.html#AkiraNetwork\_VirtualStorageLibrary\_VirtualNodeExtensions\_GenerateSingleTableDebugText\_1\_\_0\_)  
VirtualItem <T>  
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeExtensions.html#AkiraNetwork\_VirtualStorageLibrary\_VirtualNodeExtensions\_GenerateSingleTableDebugText\_1\_\_0\_)  
VirtualNode  
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeExtensions.html#AkiraNetwork\_VirtualStorageLibrary\_VirtualNodeExtensions\_GenerateSingleTableDebugText\_1\_\_0\_)  
VirtualNodeContext  
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeExtensions.html#AkiraNetwork\_VirtualStorageLibrary\_VirtualNodeExtensions\_GenerateSingleTableDebugText\_1\_\_0\_)  
VirtualNodeExtensions  
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeExtensions.html#AkiraNetwork\_VirtualStorageLibrary\_VirtualNodeExtensions\_GenerateSingleTableDebugText\_1\_\_0\_)  
VirtualNodeListConditions  
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeExtensions.html#AkiraNetwork\_VirtualStorageLibrary\_VirtualNodeExtensions\_GenerateSingleTableDebugText\_1\_\_0\_)  
VirtualnodeName  
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeExtensions.html#AkiraNetwork\_VirtualStorageLibrary\_VirtualNodeExtensions\_GenerateSingleTableDebugText\_1\_\_0\_)  
VirtualNodeNotFoundException  
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeExtensions.html#AkiraNetwork\_VirtualStorageLibrary\_VirtualNodeExtensions\_GenerateSingleTableDebugText\_1\_\_0\_)  
VirtualNodeType  
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeExtensions.html#AkiraNetwork\_VirtualStorageLibrary\_VirtualNodeExtensions\_GenerateSingleTableDebugText\_1\_\_0\_)  
VirtualNodeTypeFilter  
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeExtensions.html#AkiraNetwork\_VirtualStorageLibrary\_VirtualNodeExtensions\_GenerateSingleTableDebugText\_1\_\_0\_)  
VirtualPath  
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeExtensions.html#AkiraNetwork\_VirtualStorageLibrary\_VirtualNodeExtensions\_GenerateSingleTableDebugText\_1\_\_0\_)

# 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)))

### (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() (<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)

<T> VirtualTextFormatter

(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<br>(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) |  | <i>basePath</i>    | The base path from which to start.          |
| bool (https://learn.microsoft.com/dotnet/api/system.boolean)<br>▼    |  | <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> | 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**

**DefaultWildcardMatcher**

        (AkiraNetwork.VirtualStorageLibrary.WildcardMatchers.DefaultWildcardMatcher.html)

- **AkiraNetwork.VirtualStorage**

**LibraryUtilities**

        (AkiraNetwork.VirtualStorageLibrary)

- **AkiraNetwork.VirtualStorage**

**PowerShellWildcardMatcher**

        (AkiraNetwork.VirtualStorageLibrary.WildcardMatchers.PowerShellWildcardMatcher.html)

**DefaultWildcardMatcher**

        Implements wildcard matching based on PowerShell's wildcard patterns. Provides pattern matching with support for specific escape characters.

        PowerShellWildcardMatcher

        (AkiraNetwork.VirtualStorageLibrary)

# Class DefaultWildcardMatcher

Implements default wildcard matching using regular expression patterns. Provides a straightforward mapping of regular expression symbols.

Inherited from

## AkiraNetwork.VirtualStorage

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

## (AkiraNetwork.VirtualStorage)

Implements

## AkiraNetwork.VirtualStorage

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

## Library.Utilities

Inherited from

## 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))

## Library.WildcardMatchers

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.VirtualStorage)

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

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

## PowerShellWildcardMatcher

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](#)).Wildcard Matchers ([AkiraNetwork.VirtualStorageLibrary.WildcardMatchers.html](#))

**Assembly:** VirtualStorageLibrary.dll

## Syntax

```
public class DefaultWildcardMatcher : IVirtualWildcardMatcher
```

# Properties

## Count

Gets the number of supported wildcard characters.

## 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> ) | The number of supported wildcard characters. |

## Patterns

Gets the list of regular expression patterns corresponding to the wildcard characters.

### + AkiraNetwork.VirtualStorage

Declaration  
Library

#### (AkiraNetwork.VirtualStorage)

```
public IEnumerable<string> Patterns { get; }
```

### + AkiraNetwork.VirtualStorage

Library Utilities

Property Value

#### (AkiraNetwork.VirtualStorage)

Type

### - AkiraNetwork.VirtualStorage

IEnumerable

#### Library.WildcardMatchers

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

#### (AkiraNetwork.VirtualStorage)

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

(AkiraNetwork.VirtualStorageLibrary)

PowerShellWildcardMatcher

(AkiraNetwork.VirtualStorageLibrary)

## WildcardDictionary

Gets the dictionary of regular expression symbols and their corresponding literals.

### Declaration

```
public ReadOnlyDictionary<string, string> WildcardDictionary { get; }
```

Property Value

| Type   | Description  |
|--|--|
| ReadOnlyDictionary<br>( <a href="https://learn.microsoft.com/dotnet/api/system.collections.objectmodel.readonlydictionary-2">https://learn.microsoft.com/dotnet/api/system.collections.objectmodel.readonlydictionary-2</a> )<br><string ( <a href="https://learn.microsoft.com/dotnet/api/system.string">https://learn.microsoft.com/dotnet/api/system.string</a> ),<br>string ( <a href="https://learn.microsoft.com/dotnet/api/system.string">https://learn.microsoft.com/dotnet/api/system.string</a> )> | A read-only dictionary where the keys are regular expression symbols and the values are their literal equivalents. |

## Wildcards

Gets the list of supported wildcard characters.

### Declaration

```
public IEnumerable<string> Wildcards { get; }
```

## Property Value

| Type   | Description                                    |
|--|--|
| IEnumerable<br>( <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> )<br><string ( <a href="https://learn.microsoft.com/dotnet/api/system.string">https://learn.microsoft.com/dotnet/api/system.string</a> )> | A collection of supported wildcard characters. |

## + AkiraNetwork.VirtualStorage

### Library

## Methods

## + AkiraNetwork.VirtualStorage

### PatternMatcher(string, string)

#### (AkiraNetwork.VirtualStorage)

Performs simple regular expression-based wildcard matching.

## - AkiraNetwork.VirtualStorage

Declaration

### Library.WildcardMatchers

#### (AkiraNetwork.VirtualStorage)

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

DefaultWildcardMatcher

(AkiraNetwork.VirtualStorageLibrary)

### Parameters

PowerShellWildcardMatcher

| Type  | Name     | Description   |
|---|----------|---|
| string<br>( <a href="https://learn.microsoft.com/dotnet/api/system.string">https://learn.microsoft.com/dotnet/api/system.string</a> ) | nodeName | The name of the node to match against the pattern.  |
| string<br>( <a href="https://learn.microsoft.com/dotnet/api/system.string">https://learn.microsoft.com/dotnet/api/system.string</a> ) | pattern  | The regular expression pattern to use for matching. |

### Returns

| Type  | Description  |
|---|--|
| bool<br>( <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\_)

▼

+ **AkiraNetwork.VirtualStorage  
Library**  
[\(AkiraNetwork.VirtualStorageLibrary\)](#)

+ **AkiraNetwork.VirtualStorage  
Library.Utilities**  
[\(AkiraNetwork.VirtualStorageLibraryUtilities\)](#)

- **AkiraNetwork.VirtualStorage  
Library.WildcardMatchers**  
[\(AkiraNetwork.VirtualStorageLibraryWildcardMatchers\)](#)

    DefaultWildcardMatcher  
[\(AkiraNetwork.VirtualStorageLibraryDefaultWildcardMatcher\)](#)  
    PowerShellWildcardMatcher  
[\(AkiraNetwork.VirtualStorageLibraryPowerShellWildcardMatcher\)](#)

# Class PowerShellWildcardMatcher

Implements wildcard matching based on PowerShell's wildcard patterns. Provides pattern matching with support for ~~Filter by title~~ specific escape characters.

Implements

## AkiraNetwork.VirtualStorage

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

## (AkiraNetwork.VirtualStorage)

Implements

## AkiraNetwork.VirtualStorage

IVirtualWildcardMatcher ([AkiraNetwork.VirtualStorageLibrary.IVirtualWildcardMatcher.html](https://learn.microsoft.com/dotnet/api/AkiraNetwork.VirtualStorageLibrary.IVirtualWildcardMatcher.html))

Library: Utilities

## (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))

Library: WildcardMatchers

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)))

DefaultWildcardMatcher

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

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

PowerShellWildcardMatcher

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](https://learn.microsoft.com/dotnet/api/AkiraNetwork.html)).VirtualStorageLibrary ([AkiraNetwork.VirtualStorageLibrary.html](https://learn.microsoft.com/dotnet/api/AkiraNetwork.VirtualStorageLibrary.html)).Wildcard Matchers ([AkiraNetwork.VirtualStorageLibrary.WildcardMatchers.html](https://learn.microsoft.com/dotnet/api/AkiraNetwork.VirtualStorageLibrary.WildcardMatchers.html))

**Assembly:** VirtualStorageLibrary.dll

Syntax

```
public class PowerShellWildcardMatcher : IVirtualWildcardMatcher
```

## Properties

### Count

Gets the number of supported wildcard characters.

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> ) | The number of supported wildcard characters. |

## Patterns

Gets the list of regular expression patterns corresponding to the wildcard characters.

### + AkiraNetwork.VirtualStorage

Declaration  
Library

#### (AkiraNetwork.VirtualStorage)

```
public IEnumerable<string> Patterns { get; }
```

### + AkiraNetwork.VirtualStorage

Library Utilities

Property Value

#### (AkiraNetwork.VirtualStorage)

Type

### - AkiraNetwork.VirtualStorage

IEnumerable

#### Library.WildcardMatchers

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

#### (AkiraNetwork.VirtualStorage)

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

(AkiraNetwork.VirtualStorageLibrary)

PowerShellWildcardMatcher

(AkiraNetwork.VirtualStorageLibrary)

## WildcardDictionary

Gets the dictionary of wildcard characters and their corresponding regular expression patterns.

### Declaration

```
public ReadOnlyDictionary<string, string> WildcardDictionary { get; }
```

Property Value

| Type   | Description   |
|--|---|
| ReadOnlyDictionary<br>( <a href="https://learn.microsoft.com/dotnet/api/system.collections.objectmodel.readonlydictionary-2">https://learn.microsoft.com/dotnet/api/system.collections.objectmodel.readonlydictionary-2</a> )<br><string ( <a href="https://learn.microsoft.com/dotnet/api/system.string">https://learn.microsoft.com/dotnet/api/system.string</a> ),<br>string ( <a href="https://learn.microsoft.com/dotnet/api/system.string">https://learn.microsoft.com/dotnet/api/system.string</a> )> | A read-only dictionary where the keys are wildcard characters and the values are their corresponding regular expression patterns. |

## Wildcards

Gets the list of supported wildcard characters.

### Declaration

```
public IEnumerable<string> Wildcards { get; }
```

## Property Value

| Type   | Description                                    |
|--|--|
| IEnumerable<br>( <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> )<br><string ( <a href="https://learn.microsoft.com/dotnet/api/system.string">https://learn.microsoft.com/dotnet/api/system.string</a> )> | A collection of supported wildcard characters. |

## + AkiraNetwork.VirtualStorage

### Library

## Methods

## + AkiraNetwork.VirtualStorage

### PatternMatcher(string, string)

#### (AkiraNetwork.VirtualStorage)

Performs pattern matching using the specified PowerShell wildcard pattern.

## - AkiraNetwork.VirtualStorage

Declaration

### Library.WildcardMatchers

#### (AkiraNetwork.VirtualStorage)

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

DefaultWildcardMatcher

(AkiraNetwork.VirtualStorageLibrary)

### Parameters

PowerShellWildcardMatcher

| Type  | Name     | Description  |
|---|----------|--|
| string<br>( <a href="https://learn.microsoft.com/dotnet/api/system.string">https://learn.microsoft.com/dotnet/api/system.string</a> ) | nodeName | The name of the node to match against the pattern.   |
| string<br>( <a href="https://learn.microsoft.com/dotnet/api/system.string">https://learn.microsoft.com/dotnet/api/system.string</a> ) | pattern  | The PowerShell wildcard pattern to use for matching. |

### Returns

| Type  | Description  |
|---|--|
| bool<br>( <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. |

### Remarks

If a wildcard character is escaped using a backtick (`), the character is treated as a literal. The escape character itself is represented by the backtick (`), and the character following it is the one being escaped.

## 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\_)

▼

- + **AkiraNetwork.VirtualStorage**  
**Library**  
**(AkiraNetwork.VirtualStorage)**
- + **AkiraNetwork.VirtualStorage**  
**Library.Utilities**  
**(AkiraNetwork.VirtualStorage)**
- **AkiraNetwork.VirtualStorage**  
**Library.WildcardMatchers**  
**(AkiraNetwork.VirtualStorage)**

DefaultWildcardMatcher  
(AkiraNetwork.VirtualStorageLibrary.WildcardMatchers.DefaultWildcardMatcher)

PowerShellWildcardMatcher  
(AkiraNetwork.VirtualStorageLibrary.WildcardMatchers.PowerShellWildcardMatcher)