

# 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

(AkiraNetwork.VirtualStorageLibrary)

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**

#### **([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**

## **([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](#))

VirtualItem<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**

## **([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**

## **([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
```

IVirtualDeepCloneable<T>

(AkiraNetwork.VirtualStorageLibrary.html)

Type Parameters

1.html)

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

NotifyNodeDelegate

(AkiraNetwork.VirtualStorageLibrary.html)

PatternMatch

(AkiraNetwork.VirtualStorageLibrary.html)

VirtualCycleDetector

(AkiraNetwork.VirtualStorageLibrary.html)

VirtualDirectory

(AkiraNetwork.VirtualStorageLibrary.html)

## Methods

DeepClone(bool)

(AkiraNetwork.VirtualStorageLibrary.html)

VirtualException

Creates a deep clone of the object.

(AkiraNetwork.VirtualStorageLibrary.html)

Declaration

virtual IVirtualDeepCloneable<T, TKey>

(AkiraNetwork.VirtualStorageLibrary.html)

T DeepClone(bool recursive = false)

VirtualID

(AkiraNetwork.VirtualStorageLibrary.html)

Parameters

VirtualItem

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

VirtualNodeContext

Returns

(AkiraNetwork.VirtualStorageLibrary.html)

Type	Description
T	The cloned object.

(AkiraNetwork.VirtualStorageLibrary.html)

VirtualNodeName

(AkiraNetwork.VirtualStorageLibrary.html)

VirtualNodeNotFoundException

(AkiraNetwork.VirtualStorageLibrary.html)

# 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



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

## Library

### 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.readonlydictionary-2>)

(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-1>)

(AkiraNetwork.VirtualStorageLibrary)

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

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeListConditions

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeName

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeNotFoundException

(AkiraNetwork.VirtualStorageLibrary)

PatternMatcher(string, string)

VirtualNodeType

## Methods

### PatternMatcher(string, string)

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

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

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

(AkiraNetwork.VirtualStorageLibrary)

Type	Name	Description
NotifyNodeDelegate		
VirtualPath	path	The virtual path of the node being notified.
VirtualNode	node	The instance of the node being notified.

(AkiraNetwork.VirtualStorageLibrary.PatternMatcher)

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

(AkiraNetwork.VirtualStorageLibrary.VirtualNode)

VirtualCycleDetector

(AkiraNetwork.VirtualStorageLibrary.VirtualNode)

(AkiraNetwork.VirtualStorageLibrary.VirtualNode.html)

VirtualDirectory

(AkiraNetwork.VirtualStorageLibrary)

## Extension Methods

(AkiraNetwork.VirtualStorageLibrary)

VirtualTextFormatter GenerateSingleTableDebugText<T>(T)

(VirtualGroupCondition<'T, Keys>)

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

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)

# 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)

*nodeName*

The name of the node to be matched.

VirtualCycleDetector

string (AkiraNetwork.VirtualStorageLibrary)

*pattern*

The pattern to match against.

(AkiraNetwork.VirtualStorageLibrary)

**Returns:** VirtualException

Type (AkiraNetwork.VirtualStorageLibrary)

**Description**

VirtualGroupCondition<T, TKey>

bool (AkiraNetwork.VirtualStorageLibrary)

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

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

## VirtualCycleDetector()

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

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

## Properties

VirtualNotFoundException  
    (AkiraNetwork.VirtualStorageLibrary)  
VirtualNodeType  
    VirtualNodeTypeFilter  
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> ) IVirtualDeepCloneable<T> (AkiraNetwork.VirtualStorageLibrary. 1.html) VirtualWildcardMatcher (AkiraNetwork.VirtualStorageLibrary. Gets the dictionary of detected cycles. NotifyNodeDelegate Declaration (AkiraNetwork.VirtualStorageLibrary. PatternMatch p(AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink) CycleDictionary { get; } VirtualCycleDetector (AkiraNetwork.VirtualStorageLibrary. VirtualDirectory Property Value 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 Dictionary (AkiraNetwork.VirtualStorageLibrary. ( <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> ) VirtualGroupCondition<T, TKey> < (AkiraNetwork.VirtualStorageLibrary. < 2.html) VirtualID VirtualID (AkiraNetwork.VirtualStorageLibrary.VirtualID.html) (AkiraNetwork.VirtualStorageLibrary. , VirtualItem VirtualSymbolicLink (AkiraNetwork.VirtualStorageLibrary. (AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLi VirtualItem<T> nk.html) > (AkiraNetwork.VirtualStorageLibrary. 1.html)	A dictionary where the keys are VirtualID (AkiraNetwork.VirtualStorageLibrary.VirtualID.html) s and the values are the corresponding VirtualSymbolicLink (AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink. html) 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 (AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink.html)	link	The VirtualSymbolicLink (AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink.html) to check.

### Returns

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

## Extension Methods

VirtualTextFormatter.GenerateSingleTableDebugText<T>(T VirtualID (AkiraNetwork.VirtualStorageLibrary.Utilities.VirtualTextFormatter.html#AkiraNetwork_VirtualStorageLibrary_Utils s_VirtualTextFormatter_GeneratesSingleTableDebugText_1__0_)	VirtualItem (AkiraNetwork.VirtualStorageLibrary.VirtualItem.html)
VirtualAllItem<T> (AkiraNetwork.VirtualStorageLibrary.VirtualAllItem.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)	

# 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.VirtualDirectory.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.VirtualNode)

Inherited Members

VirtualItem

AkiraNetwork.VirtualStorageLibrary

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

VirtualNode (AkiraNetwork.CreatedDataStorageLibrary)

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

VirtualNode (AkiraNetwork.UpdatedDataStorageLibrary)

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

VirtualNode (AkiraNetwork.UpdatedDataStorageLibrary)

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

VirtualNode (AkiraNetwork.BeforeConditionStorage)

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

VirtualNodeName

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

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

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

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

object (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
VirtualNodeName	name	The name of the directory.

VirtualNodeName

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)

VirtualNodeName

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeNotFoundException.html)

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeNotFoundException.html)

VirtualNodeNotFoundException

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

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

(AkiraNetwork.VirtualStorageLibrary.VirtualDirectory.html)

**Declaration**

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeTypeFilter.html)

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

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

## Parameters

Type		Name	Description
VirtualNodeName (AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)		name	The name of the directory.
DateTime (https://learn.microsoft.com/dotnet/api/system.datetime 1.html)		createdDate	The creation date.
DateTime (https://learn.microsoft.com/dotnet/api/system.datetime 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
IVirtualDeepCloneable<T> (AkiraNetwork.VirtualStorageLibrary int (https://learn.microsoft.com/dotnet/api/system.int32) 1.html)	

IVirtualWildcardMatcher  
(AkiraNetwork.VirtualStorageLibrary  
VirtualNodeList  
VirtualNodeContext  
VirtualPath)

### this[VirtualnodeName]

(AkiraNetwork.VirtualStorageLibrary  
VirtualNodeList  
VirtualNodeContext  
VirtualPath)

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

(AkiraNetwork.VirtualStorageLibrary  
VirtualCycleDetector  
VirtualDirectory)

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

VirtualException

Type	Name	Description
VirtualGroupCondition<T, TKey>	name	The name of the node.

VirtualID

(AkiraNetwork.VirtualStorageLibrary  
VirtualNodeList  
VirtualNodeContext  
VirtualPath)

VirtualItem

Type	Description
VirtualItem<T>	The node with the specified name.

VirtualNode

Type	Description
VirtualNode	The node with the specified name.

VirtualNotFoundException

Exceptions	Condition
VirtualNotFoundException	Thrown if the node is not found.

VirtualNodeExtensions

(AkiraNetwork.VirtualStorageLibrary  
VirtualNodeList  
VirtualNodeContext  
VirtualPath)

ItemCount

(AkiraNetwork.VirtualStorageLibrary  
VirtualNodeList  
VirtualNodeContext  
VirtualPath)

Gets the number of items within the directory.

(AkiraNetwork.VirtualStorageLibrary  
VirtualNodeNotFoundException  
VirtualNodeType)

VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary  
VirtualNodeList  
VirtualNodeContext  
VirtualPath)

VirtualNodeTypeFilter

(AkiraNetwork.VirtualStorageLibrary  
VirtualPath)

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](#) VirtualDirectoryName

VirtualGroupCondition<T, TKey>

(AkiraNetwork.VirtualStorageLibrary)

[2.html](#)

```
public IEnumerable<VirtualDirectoryName> 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>)<

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

VirtualNode

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeContext

Node Type: [AkiraNetwork.VirtualStorageLibrary](#)

VirtualNodeExtensions

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

Description

(AkiraNetwork.VirtualStorageLibrary)

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)  
`IVirtualDeepCloneable<T>`  
`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.

**Declaration**  
 (AkiraNetwork.VirtualStorageLibrary.VirtualNode.html#AkiraNetwork\_VirtualStorageLibrary\_VirtualNode\_NodesView)  
`2.html`

`public IEnumerable<VirtualNode> NodesView { get; }`  
 (AkiraNetwork.VirtualStorageLibrary.VirtualNode.html#AkiraNetwork\_VirtualStorageLibrary\_VirtualNode\_NodesView)

### Property Value

Type	Description
VirtualItem<T> (AkiraNetwork.VirtualStorageLibrary.VirtualNode.html#AkiraNetwork_VirtualStorageLibrary_VirtualNode_NodesView)	

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

VirtualNodeContext

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

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

**Declaration**  
 (AkiraNetwork.VirtualStorageLibrary.VirtualNode.html#AkiraNetwork\_VirtualStorageLibrary\_VirtualNode\_NodesViewCount)  
`VirtualNodeListConditions`

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

VirtualNodeNotFoundException

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

### Property Value

Type	Description
VirtualNodeType (AkiraNetwork.VirtualStorageLibrary.VirtualNode.html#AkiraNetwork_VirtualStorageLibrary_VirtualNode_NodesViewCount)	

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

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

VirtualPath

# SymbolicLinkCount

Gets the number of symbolic links within the directory.

## Declaration

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

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

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

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

## DeclarationNetwork.VirtualStorageLibrary

```
VirtualDirectory  
(AkiraNetwork.VirtualStorageLibrary  
public int SymbolicLinkViewCount { get; }  
VirtualException  
(AkiraNetwork.VirtualStorageLibrary
```

Property	Type	Description
VirtualGroupCondition<T, TKey>	(AkiraNetwork.VirtualStorageLibrary 2.html)	
VirtualID	int ( <a href="https://learn.microsoft.com/dotnet/api/system.int32">https://learn.microsoft.com/dotnet/api/system.int32</a> )	

# Methods

## Add(VirtualNode, bool)

(AkiraNetwork.VirtualStorageLibrary)  
Adds a node to the directory.

```
(AkiraNetwork.VirtualStorageLibrary  
Declaration  
    VirtualNodeExtensions  
  
(AkiraNetwork.VirtualStorageLibrary  
public VirtualNode Add(VirtualNode node, bool allowOverwrite = false)  
    VirtualNodeList conditions
```

## Parameters

Type	Name	Description
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeNotFoundException)		
(AkiraNetwork.VirtualStorageLibrary.VirtualNode)	node	The node to add.
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeType)		
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeTypeFilter)		
(AkiraNetwork.VirtualStorageLibrary.VirtualPath)		

Type	Name	Description
bool ( <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> ( <a href="#">AkiraNetwork.VirtualStorageLibrary.VirtualNode.html</a> )	The added node.

## Exceptions

Type	Condition
<code>IVirtualWildcardMatcher</code> ( <a href="#">AkiraNetwork.VirtualStorageLibrary.Arguments.Matcher.html</a> )	Thrown if an invalid node name is specified.
<code>InvalidOperationException</code> ( <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> <code>VirtualDirectoryName</code> ( <a href="#">AkiraNetwork.VirtualStorageLibrary.VirtualDirectoryName.html</a> )	<code>name</code>	The name of the directory.
<code>bool</code> ( <a href="https://learn.microsoft.com/dotnet/api/system.boolean">https://learn.microsoft.com/dotnet/api/system.boolean</a> )	<code>allowOverwrite</code>	If true, allows overwriting an existing node with the same name.

## Returns

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

### Returns

AkiraNetwork.VirtualStorageLibrary

Type	Description
VirtualException (AkiraNetwork.VirtualStorageLibrary VirtualItem<AkiraNetwork.VirtualStorageLibrary.VirtualItem-1.html> <T> VirtualGroupCondition, Key (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<VirtualNode> AddSymbolicLink(VirtualNodeName name, VirtualPath targetPath, bool allowOverwrite = false)
```

(AkiraNetwork.VirtualStorageLibrary

### Parameters

VirtualNodeName

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

VirtualNodeName

(AkiraNetwork.VirtualStorageLibrary

VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary

VirtualPath





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

#### Parameters

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

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)

#### Returns

Type	Description
IVirtualWildcardMatcher (AkiraNetwork.VirtualStorageLibrary.VirtualWildcardMatcher.html)	

VirtualDirectoryDelegate

(AkiraNetwork.VirtualStorageLibrary.VirtualDirectory.html)

PatternMatch

(AkiraNetwork.VirtualStorageLibrary.VirtualMatch.html)

#### Exceptions

Type	Condition
VirtualCycleDetector (AkiraNetwork.VirtualStorageLibrary.VirtualCycleDetector.html)	
VirtualNotFoundException (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

VirtualNodeExtensions

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeExtensions.html)

An enumerator for filtered nodes.

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

(AkiraNetwork.VirtualStorageLibrary.VirtualNode.html)

VirtualNodeName

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)

VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeType.html)

#### Declaration

VirtualNodeTypeFilter

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeTypeFilter.html)

VirtualPath

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

#### GetItem<T>(VirtualNodeName)

(AkiraNetwork.VirtualStorageLibrary.VirtualItem.html)

Gets the item with the specified name.

VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeType.html)

#### Declaration

VirtualPath

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

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

#### Parameters

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

Return Value  
[AkiraNetwork.VirtualStorageLibrary.VirtualItem-1.html](#)

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

NotifyNodeDelegate  
(AkiraNetwork.VirtualStorageLibrary)

Type Parameters  
PatternMatch  
Name  
(AkiraNetwork.VirtualStorageLibrary)

T VirtualCycleDetector The data type of the item.  
(AkiraNetwork.VirtualStorageLibrary)

VirtualDirectory  
(AkiraNetwork.VirtualStorageLibrary)

Exceptions  
(AkiraNetwork.VirtualStorageLibrary)

Type	Condition
VirtualException (AkiraNetwork.VirtualStorageLibrary)	Thrown if the node is not found.
VirtualNotFoundException (AkiraNetwork.VirtualStorageLibrary)	Thrown if the specified node is not an item.

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

(AkiraNetwork.VirtualStorageLibrary)

VirtualItem  
(AkiraNetwork.VirtualStorageLibrary)

GetNodesView()  
(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  
(AkiraNetwork.VirtualStorageLibrary)

>Returns  
(AkiraNetwork.VirtualStorageLibrary)

Type	Description
virtualNodeListConditions (AkiraNetwork.VirtualStorageLibrary)	An enumeration of nodes based on the display conditions.

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

<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 (AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html) 1.html)	name	The name of the symbolic link.

### Returns

Type	Description
VirtualWildcardMatcher (AkiraNetwork.VirtualStorageLibrary.NotifyNodeDelegate)	The symbolic link with the specified name.

### Exceptions

Type	Condition
VirtualDirectory (AkiraNetwork.VirtualStorageLibrary.VirtualNotFoundException)	Thrown if the node is not found.
VirtualException (AkiraNetwork.VirtualStorageLibrary.VirtualNodeNotFoundException.html) I) (AkiraNetwork.VirtualStorageLibrary.VirtualGroupCondition<T, TKey> InvalidOperationException (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> ) n) VirtualID (AkiraNetwork.VirtualStorageLibrary.VirtualItem VirtualItem (AkiraNetwork.VirtualStorageLibrary.VirtualItem<T>))	Thrown if the specified node is not a symbolic link.

## ItemExists(VirtualNodeName)

### Parameters

Type	Name	Description
VirtualNodeListConditions (AkiraNetwork.VirtualStorageLibrary.VirtualNodeName)	name	The name of the item to check.

### Returns

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

Type  
IVirtualDeepCloneable<T>  
(AkiraNetwork.VirtualStorageLibrary.html)

Name

Description

VirtualWildcardMatcher  
(AkiraNetwork.VirtualStorageLibrary.html)  
NotifyNodeDelegate  
(AkiraNetwork.VirtualStorageLibrary.html)

name

The name of the node to check.

#### Returns

PatternMatch

Type  
AkiraNetwork.VirtualStorageLibrary

Description

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

True if the node exists; otherwise, false.

VirtualDirectory

(AkiraNetwork.VirtualStorageLibrary)

## **Remove(VirtualNode)**

(AkiraNetwork.VirtualStorageLibrary)

Removes the specified node from the directory.

VirtualGroupCondition<T, TKey>

Declaration  
2.html

VirtualID  
public void Remove(VirtualNode node)  
(AkiraNetwork.VirtualStorageLibrary)

VirtualItem

Parameters  
AkiraNetwork.VirtualStorageLibrary

Type  
VirtualItem<T>  
(AkiraNetwork.VirtualStorageLibrary)

Name

Description

VirtualNode  
(AkiraNetwork.VirtualStorageLibrary)

node

The node to remove.

### Exceptions

Type  
VirtualNodeContext  
(AkiraNetwork.VirtualStorageLibrary)

Condition

VirtualNodeNotFoundException  
(AkiraNetwork.VirtualStorageLibrary)

Thrown if the node is not found.

VirtualNodeListConditions  
(AkiraNetwork.VirtualStorageLibrary)

VirtualnodeName  
(AkiraNetwork.VirtualStorageLibrary)

## **SymbolicLinkExists(VirtualnodeName)**

VirtualNodeNotFoundException

Checks whether a symbolic link with the specified name exists.

(AkiraNetwork.VirtualStorageLibrary)

### Declaration

VirtualNodeType  
(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeTypeFilter  
public bool SymbolicLinkExists(VirtualnodeName name)

(AkiraNetwork.VirtualStorageLibrary)

VirtualPath

## Parameters

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

## Returns

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

## IVirtualWildcardMatcher

### (AkiraNetwork.VirtualStorageLibrary.VirtualNode)

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> (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 (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 ( <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 to 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

(AkiraNetwork VirtualStorage library)

### Virtual Directory

Type	Description	Name	
<code>(AkiraNetwork.VirtualStorageLibrary.VirtualException)</code>			
<code>(AkiraNetwork.VirtualStorageLibrary.VirtualDirectory)</code> <code>(AkiraNetwork.VirtualStorageLibrary.VirtualDirectory.html)</code>	The directory to which the node is added.	<code>directory</code>	
<code>(AkiraNetwork.VirtualStorageLibrary.VirtualGroupCondition&lt;T, TKey&gt;)</code>			
<code>(AkiraNetwork.VirtualStorageLibrary.VirtualNode)</code> <code>(AkiraNetwork.VirtualStorageLibrary.VirtualNode.html)</code>	The node to add.	<code>node</code>	

## VirtualID

Return AkiraNetwork.VirtualStorageLibrary

Type	Description
VirtualItem (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 (AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)	nodeName	The node name of directory to add.

## Returns

Type	Description
IVirtualDeepCloneable<T> VirtualDirectory (AkiraNetwork.VirtualStorageLibrary.VirtualDirectory.html) (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
(AkiraNetwork.VirtualStorageLibrary.VirtualGroupCondition<T, TKey>	nodeName	The node name to convert.

### Return Value

Type	Description
VirtVirtualItem (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
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)	nodeName	The node name to convert.

### Return Value

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

## Properties

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

## Extension Methods

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

## Overrides

(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\_Utilitie

s\_VirtualNodeName  
(AkiraNetwork.VirtualStorageLibrary)

VirtualTextFormatter.GenerateSingleTableDebugText\_1\_\_0\_)

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeNotFoundException

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeTypeFilter

(AkiraNetwork.VirtualStorageLibrary)

VirtualPath

(AkiraNetwork.VirtualStorageLibrary)



## Syntax

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



## Constructors

(AkiraNetwork.VirtualStorageLibrary  
1.html)

### **VirtualItem(VirtualNodeName)**

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

NotifyNodeDelegate

Declaration  
PatternMatch

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

(AkiraNetwork.VirtualStorageLibrary  
VirtualPath

Parameters  
Directory

(AkiraNetwork.VirtualStorageLibrary  
Type  
VirtualException

Type	Name	Description
VirtualException	name	The name of the item.

(AkiraNetwork.VirtualStorageLibrary  
VirtualGroupCondition<T, TKey>

(AkiraNetwork.VirtualStorageLibrary  
2.html)

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

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

VirtualItem

Declaration  
VirtualItem<T>

protected VirtualItem(VirtualNodeName name, DateTime createdDate)

1.html)

VirtualNode

Parameters  
(AkiraNetwork.VirtualStorageLibrary  
VirtualNodeContext

Type	Name	Description
VirtualNodeContext	name	The name of the item.

(AkiraNetwork.VirtualStorageLibrary  
VirtualNodeName

VirtualNodeExtensions

(AkiraNetwork.VirtualStorageLibrary VirtualNodeName.html)	name	The name of the item.
--	------	-----------------------

DateTimes ( <a href="https://learn.microsoft.com/dotnet/api/system.datetime">https://learn.microsoft.com/dotnet/api/system.datetime</a> )	createdDate	The creation date of the item.
---	-------------	--------------------------------

VirtualNodeName

(AkiraNetwork.VirtualStorageLibrary  
VirtualNodeNotFoundException

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

(AkiraNetwork.VirtualStorageLibrary  
VirtualNodeType

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

(AkiraNetwork.VirtualStorageLibrary  
VirtualNodeTypeFilter

(AkiraNetwork.VirtualStorageLibrary  
VirtualPath

(AkiraNetwork.VirtualStorageLibrary  
VirtualPathFilter

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

## Parameters

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

## Methods

VirtualMatch  
(AkiraNetwork.VirtualStorageLibrary)

VirtualCycleDetector  
(AkiraNetwork.VirtualStorageLibrary)

VirtualDirectory  
(AkiraNetwork.VirtualStorageLibrary)

VirtualException  
(AkiraNetwork.VirtualStorageLibrary)

**DeepClone(bool)**

VirtualDeepCloneable<T>  
(AkiraNetwork.VirtualStorageLibrary)

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

VirtualID  
Declaration  
(AkiraNetwork.VirtualStorageLibrary)

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

VirtualItem<T>  
(AkiraNetwork.VirtualStorageLibrary)

## Parameters

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

VirtualNodeExtensions

Returns  
(AkiraNetwork.VirtualStorageLibrary)

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

VirtualNodeType  
VirtualNode.DeepClone(bool)

(AkiraNetwork.VirtualStorageLibrary)

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

VirtualPath  
(AkiraNetwork.VirtualStorageLibrary)

# Implements

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

## Extension Methods

VirtualTextFormatter.GenerateSingleTableDebugText<T>(T)  
(AkiraNetwork.VirtualStorageLibrary.Utilities.VirtualTextFormatter.html#AkiraNetwork\_VirtualStorageLibrary\_Utils  
ies\_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

VirtualNodeExtensions

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

## Syntax

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

### Type Parameters

Name	Description
T	(AkiraNetwork.VirtualStorageLibrary.VirtualItem-1.html) The user-defined type library can be freely specified by the user of this library.

1.html)  
IVirtualWildcardMatcher  
(AkiraNetwork.VirtualStorageLibrary.NotifyNodeDelegate

(AkiraNetwork.VirtualStorageLibrary.VirtualPathMatch  
VirtualPathMatch)

(AkiraNetwork.VirtualStorageLibrary.VirtualPathMatch)

Initializes a new instance of the VirtualItem<T> (AkiraNetwork.VirtualStorageLibrary.VirtualItem-1.html) class. The node name is automatically generated by the GenerateNodeName(string) (AkiraNetwork.VirtualStorageLibrary.VirtualnodeName.html#AkiraNetwork\_VirtualStorageLibrary\_VirtualnodeName\_GenerateNodeName\_System\_String) method.

VirtualException  
Declaration  
(AkiraNetwork.VirtualStorageLibrary.VirtualItem)

```
public VirtualItem()  
(AkiraNetwork.VirtualStorageLibrary.VirtualItem-2.html)
```

VirtualID  
(AkiraNetwork.VirtualStorageLibrary.VirtualItem)

VirtualItem(VirtualDirectoryName)  
VirtualItem

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

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

```
public VirtualItem(VirtualDirectoryName name)  
(AkiraNetwork.VirtualStorageLibrary.VirtualItem-2.html)
```

VirtualNodeContext  
Parameters  
(AkiraNetwork.VirtualStorageLibrary.VirtualItem)

Type  
VirtualNodeExtensions  
(AkiraNetwork.VirtualStorageLibrary.VirtualItem)

VirtualNodeName  
VirtualNodeName (AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)

VirtualNodeListConditions  
(AkiraNetwork.VirtualStorageLibrary.VirtualItem)

VirtualPath  
VirtualPath

VirtualNodeNotNotFoundException  
Initialization  
(AkiraNetwork.VirtualStorageLibrary.VirtualItem-1.html)

VirtualNodeType  
Declaration  
(AkiraNetwork.VirtualStorageLibrary.VirtualItem-1.html)

```
public VirtualItem(VirtualDirectoryName name, T? itemData)  
(AkiraNetwork.VirtualStorageLibrary.VirtualItem-2.html)
```

VirtualPath  
Parameters  
(AkiraNetwork.VirtualStorageLibrary.VirtualItem)

Name	Description
name	The name of the item.

### VirtualItem(VirtualDirectoryName, T?)

VirtualNodeNotFoundException  
Initialization  
(AkiraNetwork.VirtualStorageLibrary.VirtualItem-1.html)

VirtualNodeType  
Declaration  
(AkiraNetwork.VirtualStorageLibrary.VirtualItem-1.html)

```
public VirtualItem(VirtualDirectoryName name, T? itemData)  
(AkiraNetwork.VirtualStorageLibrary.VirtualItem-2.html)
```

VirtualPath  
Parameters  
(AkiraNetwork.VirtualStorageLibrary.VirtualItem)

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

## (AkiraNetwork.VirtualStorageLibrary 1.html)

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

(AkiraNetwork.VirtualStorageLibrary)

Initial Notify Node Delegate of the VirtualItem<T> (AkiraNetwork.VirtualStorageLibrary.VirtualItem-1.html) class.

(AkiraNetwork.VirtualStorageLibrary)

## Declaration PatternMatch

(AkiraNetwork VirtualStorageLibrary)

```
public virtual Item<VirtualNodeName> name, T? itemData, DateTime createdDate)
```

(AkiraNetwork VirtualStorage library)

Vista del Directorio

## VirtualDirectory Parameters

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

1.html

## VirtualNode

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

### **VirtualNodeContext**

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

Virtual Network Functions

# Virtual Node Extensions Declaration

(AkiraNetwork.VirtualSt

```
VirtualNodeListConditions  
public VirtualItem(VirtualNodeName name, T? itemData, DateTime createdDate, DateTime updated  
(AkiraNetwork VirtualStorage librar
```

VirtualNodeName

## Virtueller NodeName (AkiraNetworks VirtualStorage Library)

(AkiraNetwork.virtualStorageLib)

Parameters			
Type	Name	Description	
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeType)			
Virt	<i>name</i>	The name of the item.	
(AkiraNetwork.VirtualStorageLibrary.VirtualNode)			
Main			
(AkiraNetwork.VirtualStorageLibrary.VirtualPath)			

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 ( <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 ( <a href="https://learn.microsoft.com/dotnet/api/system.datetime">https://learn.microsoft.com/dotnet/api/system.datetime</a> ) VirtualWildcardMatcher ( <a href="https://learn.microsoft.com/dotnet/api/system.datetime">https://learn.microsoft.com/dotnet/api/system.datetime</a> ) NotifyNodeDelegate (AkiraNetwork.VirtualStorageLibrary.VirtualWildcardMatcher.NotifyNodeDelegate)	<i>updatedDate</i>	The last update date of the item.
PatternMatch (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 class VirtualGroupCondition : VirtualCondition<T, TKey>
```

VirtualID

### PropertyValue

(AkiraNetwork.VirtualStorageLibrary.VirtualGroupCondition)

### Type

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

(AkiraNetwork.VirtualStorageLibrary.VirtualGroupCondition)

1.html

### NodeType

(AkiraNetwork.VirtualStorageLibrary.VirtualNode)

Gets the type of the node.

### Declaration

```
AkiraNetwork.VirtualStorageLibrary.VirtualNodeExtensions
public override VirtualNodeType NodeType { get; }
```

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeExtensions)

### PropertyName

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeExtensions)

### Type

Type	Description
VirtualNodeNotFoundException	(AkiraNetwork.VirtualStorageLibrary.VirtualNodeExtensions)
VirtualNodeType	(AkiraNetwork.VirtualStorageLibrary.VirtualNodeExtensions)

### Overrides

VirtualNodeTypeFilter	(AkiraNetwork.VirtualStorageLibrary.VirtualNodeExtensions)
VirtualNode.NodeType	(AkiraNetwork.VirtualStorageLibrary.VirtualNodeExtensions)
e) VirtualPath	(AkiraNetwork.VirtualStorageLibrary.VirtualNodeExtensions)

# 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)
(AkiraNetwork.VirtualStorageLibrary)

VirtualPath
(AkiraNetwork.VirtualStorageLibrary)
Parameters

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

▼

## ~VirtualItem()

(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	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 <a href="https://learn.microsoft.com/dotnet/api/system.argumentexception">https://learn.microsoft.com/dotnet/api/system.argumentexception</a> (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.VirtualNode.html#AkiraNetwork\_VirtualStorageLibrary\_VirtualNode\_VirtualDirectory\_)

Declaration

VirtualException

(AkiraNetwork.VirtualStorageLibrary.VirtualNode.html#AkiraNetwork\_VirtualStorageLibrary\_VirtualNode\_VirtualException\_)

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

## Parameters

Type	Name	Description
VirtualID (AkiraNetwork.VirtualStorageLibrary.VirtualNode.html#AkiraNetwork_VirtualStorageLibrary_VirtualNode_VirtualID_)		
VirtualnodeName (AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html#AkiraNetwork_VirtualStorageLibrary_VirtualNodeName_VirtualnodeName_)	name	The name of the node.

## Returns

VirtualItem<T>

Type	Description
VirtualNode (AkiraNetwork.VirtualStorageLibrary.VirtualNode.html#AkiraNetwork_VirtualStorageLibrary_VirtualNode_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.VirtualNode.html#AkiraNetwork\_VirtualStorageLibrary\_VirtualNode\_VirtualNodeType\_)

```
public static implicit operator VirtualItem<T>(string name)
  VirtualNodeTypeFilter
  (AkiraNetwork.VirtualStorageLibrary.VirtualNode.html#AkiraNetwork_VirtualStorageLibrary_VirtualNode_VirtualNodeTypeFilter_)
```

## Parameters

VirtualPath

(AkiraNetwork.VirtualStorageLibrary.VirtualNode.html#AkiraNetwork\_VirtualStorageLibrary\_VirtualNode\_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 (AkiraNetwork.VirtualStorageLibrary.VirtualItem-1.html) allItem-1.html) <T>IVirtualWildcardMatcher (AkiraNetwork.VirtualStorageLibrary.NotifyNodeDelegate (AkiraNetwork.VirtualStorageLibrary)	A VirtualItem<T> (AkiraNetwork.VirtualStorageLibrary.VirtualItem-1.html) 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 (AkiraNetwork.VirtualStorageLibrary)		

## Returns

Type	Description
VirtualNodeExtensions (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

VIRTUALITEMCONDITION<T, TKey>

(AkiraNetwork.VirtualStorageLibrary.

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

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

VIRTUALID

(AkiraNetwork.VirtualStorageLibrary.

## Extension Methods

(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

↳ VirtualStorageLibrary

## Syntax

[VirtualNode.html](#)

VirtualNode

↳ VirtualNode : IVirtualDeepCloneable<VirtualNode>

VirtualNodeContext

↳ VirtualStorageLibrary

VirtualNodeExtensions

↳ VirtualStorageLibrary

VirtualNodeListConditions

↳ VirtualStorageLibrary

VirtualNodeName

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

```
(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) DateTime ( <a href="https://learn.microsoft.com/dotnet/api/system.datetime">https://learn.microsoft.com/dotnet/api/system.datetime</a> ) 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) 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
(AkiraNetwork.VirtualStorageLibrary) VirtualNodeListConditions	

(AkiraNetwork.VirtualStorageLibrary)

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

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeNotFoundException

(AkiraNetwork.VirtualStorageLibrary)

## NodeType

VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeTypeFilter

## Declaration

(AkiraNetwork.VirtualStorageLibrary)

VirtualPath

(AkiraNetwork.VirtualStorageLibrary)

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

## Property Value

Type	Description
VirtualNodeType (AkiraNetwork.VirtualStorageLibrary.VirtualNodeType.html) (AkiraNetwork.VirtualStorageLibrary 1.html) IVirtualWildcardMatcher (AkiraNetwork.VirtualStorageLibrary NotifyNodeDelegate Gets the updated date of the node. This date represents the last time the node was modified. It is set to the current date and time at the moment of instantiation or cloning. PatternMatch Declaration (AkiraNetwork.VirtualStorageLibrary VirtualCycleDetector p(AkiraNetwork.VirtualStorageLibrary get; ) VirtualDirectory (AkiraNetwork.VirtualStorageLibrary VirtualException	

## UpdatedDate

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

PatternMatch

## Declaration

```
VirtualCycleDetector  
p(AkiraNetwork.VirtualStorageLibrary get; )  
VirtualDirectory  
(AkiraNetwork.VirtualStorageLibrary  
VirtualException
```

## Property Value

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

VirtualID

## VID

Gets the VID of node  
(AkiraNetwork.VirtualStorageLibrary  
VirtualItem

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

```
public VirtualID VID { get; }  
VirtualNode  
(AkiraNetwork.VirtualStorageLibrary  
VirtualNodeContext
```

## Property Value

Type	Description
VirtualNodeExtensions VirtualNodeListConditions (AkiraNetwork.VirtualStorageLibrary VirtualNodeName (AkiraNetwork.VirtualStorageLibrary VirtualNodeType (AkiraNetwork.VirtualStorageLibrary VirtualPath	

## Methods

### DeepClone(bool)

Creates a deep copy of the entity. However, the CreatedDate and UpdatedDate should not be cloned as they are set to the current date and time at the time of cloning.

```
VirtualPath  
Declaration  
(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

VirtualNodeContext

(AkiraNetwork.VirtualStorageLibrary)

## Implements

IVirtualDeepCloneable<T>

(AkiraNetwork.VirtualStorageLibrary.IVirtualDeepCloneable-1.html)

VirtualNodeListConditions

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeExtensions

(AkiraNetwork.VirtualStorageLibrary)

IVirtualDeepCloneable<T>

(AkiraNetwork.VirtualStorageLibrary.IVirtualDeepCloneable-1.html)

VirtualNodeListConditions

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeExtensions

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeExtensions

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeExtensions

(AkiraNetwork.VirtualStorageLibrary)

VirtualPath

(AkiraNetwork.VirtualStorageLibrary)

## Extension Methods

(AkiraNetwork.VirtualStorageLibrary)

VirtualTextFormatter.GenerateSingleTableDebugText<T>(T)

VirtualNodeNotFoundException

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

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

VirtualNodeType

VirtualNodeExtensions.ResolveNodeType(VirtualNode)

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

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

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

VirtualNodeListConditions

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

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

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

## Property Value

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

## ResolvedLink

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

(AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink.html)

### Declaration

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

(AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink.html)

VirtualDirectory

### Property Value

(AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink.html)

Type	Description
VirtualException	The resolved symbolic link, or null.

(AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink.html)

VirtualGroupCondition<T, TKey>

(AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink.html)

(AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink.html)

2.html)

VirtualID

(AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink.html)

VirtualItem

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

(AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink.html)

### Declaration

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

(AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink.html)

VirtualNodeContext

(AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink.html)

### Property Value

(AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink.html)

Type	Description
VirtualPath	The resolved path, or null.

(AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink.html)

VirtualNodeName

(AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink.html)

VirtualNodeNotFoundException

(AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink.html)

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

VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink.html)

### Declaration

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

(AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink.html)

VirtualPath

(AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink.html)

## 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)

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

## Inherited Members

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)

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

Assembly (AkiraNetwork.VirtualStorageLibrary)

2.html)

## Syntax

VirtualID

(AkiraNetwork.VirtualStorageLibrary)  
public static class VirtualNodeExtensions  
↳ VirtualItem

(AkiraNetwork.VirtualStorageLibrary)

VirtualItem<T>

(AkiraNetwork.VirtualStorageLibrary)

1.html)

## Methods

VirtualNode

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeType(VirtualNode)

(AkiraNetwork.VirtualStorageLibrary)

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)

VirtualNodeListConditions

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeName

public static VirtualNodeType ResolveNodeType(this VirtualNode node)  
(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeNotFoundException

(AkiraNetwork.VirtualStorageLibrary)

## Parameters

VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary)

VIRTUALNODETYPEFILTER

(AkiraNetwork.VirtualStorageLibrary)

VirtualPath

(AkiraNetwork.VirtualStorageLibrary)

Type	Name	Description
VIRTUALNODETYPEFILTER (AkiraNetwork.VirtualStorageLibrary)	node	The virtual node to resolve the type for.

## Returns

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

(AkiraNetwork.VirtualStorageLibrary.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()  
VirtualNodeListConditions()  
VirtualNodeListConditions()  
VirtualNodeListConditions()

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()  
VirtualNodeListConditions()

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

VirtualNodeName  
(AkiraNetwork.VirtualStorageLibrary)

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

Declaration  
VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary)  
VirtualNodeTypeFilter listConditions(VirtualNodeTypeFilter filter, VirtualGroupCondition<Virtual  
Node, object>? groupCondition, List<VirtualSortCondition<VirtualNode>>? sortConditions)

VirtualPath  
(AkiraNetwork.VirtualStorageLibrary)

## Parameters

Type	Name	Description
VirtualNodeTypeFilter (AkiraNetwork.VirtualStorageLibrary.VirtualNodeTypeFilter.html)	<i>filter</i>	The filter criteria for the virtual nodes.
VirtualGroupCondition (AkiraNetwork.VirtualStorageLibrary.VirtualGroupCondition-2.html) <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 (AkiraNetwork.VirtualStorageLibrary (https://learn.microsoft.com/dotnet/api/system.collections.generic.list-1) NotifyNodeDelegate < (AkiraNetwork.VirtualStorageLibrary PatternMatch VirtualSortCondition (AkiraNetwork.VirtualStorageLibrary (AkiraNetwork.VirtualStorageLibrary.VirtualSortCondition-1.html) <VirtualNode (AkiraNetwork.VirtualStorageLibrary.VirtualNode.html)>> (AkiraNetwork.VirtualStorageLibrary VirtualDirectory (AkiraNetwork.VirtualStorageLibrary VirtualException (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

VirtualItemVirtualNodeTypeFilter 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 (AkiraNetwork.VirtualStorageLibrary	The filter specifying the types of virtual nodes. The default value is All.
(AkiraNetwork.VirtualStorageLibrary	All
VirtualNodeExtensions (VirtualNodeTypeFilter.html)	(AkiraNetwork.VirtualStorageLibrary.VirtualNodeTypeFilter.html#AkiraNetwork_VirtualStorageLibrary_VirtualNodeTypeFilter_All)
VirtualNodeListConditions (AkiraNetwork.VirtualStorageLibrary	

VirtualnodeName  
(AkiraNetwork.VirtualStorageLibrary

VirtualNodeNotFoundException  
(AkiraNetwork.VirtualStorageLibrary

## GroupCondition

VirtualGroupCondition  
(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 (AkiraNetwork.VirtualStorageLibrary.VirtualGroupCondition-2.html) < VirtualNode (AkiraNetwork.VirtualStorageLibrary.VirtualNode.html) , object ( <a href="https://learn.microsoft.com/dotnet/api/system.object">https://learn.microsoft.com/dotnet/api/system.object</a> )> IVirtualWildcardMatcher (AkiraNetwork.VirtualStorageLibrary) NotifyNodeDelegate (AkiraNetwork.VirtualStorageLibrary)	The condition for grouping virtual nodes. The default value is <code>null</code> . You can specify <code>null</code> if no grouping condition is needed.
SortConditions (AkiraNetwork.VirtualStorageLibrary) PatternMatch Specifies the sorting conditions, determining the order in which the virtual nodes are arranged. (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> (AkiraNetwork.VirtualStorageLibrary) List 2.html ( <a href="https://learn.microsoft.com/dotnet/api/system.collections.generic.list-1">https://learn.microsoft.com/dotnet/api/system.collections.generic.list-1</a> ) list-1(AkiraNetwork.VirtualStorageLibrary) < VirtualItem VirtualSortCondition (AkiraNetwork.VirtualStorageLibrary.VirtualSortCondition-1.html) < (AkiraNetwork.VirtualStorageLibrary) VirtualNode (AkiraNetwork.VirtualStorageLibrary.VirtualNode.html) >> (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

**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)

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

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName)

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)

**Assembly:** VirtualStorageLibrary.dll

VirtualItem

Syntax  
(AkiraNetwork.VirtualStorageLibrary)

VirtualItem<T>

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

VirtualNode

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeContext

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeExtensions

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeListConditions

**VirtualNodeName(string)**

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

class VirtualNodeName with the specified name.

(AkiraNetwork.VirtualStorageLibrary)

**Declaration**

VirtualNodeNotFoundException

(AkiraNetwork.VirtualStorageLibrary)

public VirtualNodeName(string name)

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeTypeFilter

(AkiraNetwork.VirtualStorageLibrary)

VirtualPath

(AkiraNetwork.VirtualStorageLibrary)

**Parameters**

name (string)

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	Name	Gets the name of the node.
(AkiraNetwork.VirtualStorageLibrary.html)		
VirtualItem	Declaration	AkiraNetwork.VirtualStorageLibrary
(AkiraNetwork.VirtualStorageLibrary.html)	Type	VirtualItem<T>
VirtualItem<T>		public string Name { get; }
(AkiraNetwork.VirtualStorageLibrary.html)		1.html)
VirtualNode	Property Value	VirtualNode
(AkiraNetwork.VirtualStorageLibrary.html)	Type	AkiraNetwork.VirtualStorageLibrary
VirtualNodeContext		

## Methods

CompareTo(VirtualNodeName?)	VirtualNodeName	
(AkiraNetwork.VirtualStorageLibrary.html)		CompareTo instance with a specified VirtualNodeName
(AkiraNetwork.VirtualStorageLibrary.html)	VirtualNodeName	(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html).
VirtualNodeTypeFilter		
(AkiraNetwork.VirtualStorageLibrary.html)	Declaration	AkiraNetwork.VirtualStorageLibrary
VirtualPath		
(AkiraNetwork.VirtualStorageLibrary.html)		

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

## Parameters

Type	Name	Description
VirtualNodeName (AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html) IVirtualWildcardMatcher (AkiraNetwork.VirtualStorageLibrary.NotifyNodeDelegate (AkiraNetwork.VirtualStorageLibrary.	other	The VirtualNodeName (AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html) to compare with.

## Returns

Type	Description
PatternMatch (AkiraNetwork.VirtualStorageLibrary.int VirtualCycleDetector (http://learn.microsoft.com/dotnet/api/system.io.virtualdirectory nt32)VirtualDirectory (AkiraNetwork.VirtualStorageLibrary.VirtualException (AkiraNetwork.VirtualStorageLibrary.VirtualGroupCondition<T, TKey> (AkiraNetwork.VirtualStorageLibrary.Compared this instance with a specified object. 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 object (https://learn.microsoft.com/dotnet/api/system.object) (AkiraNetwork.VirtualStorageLibrary.	obj	The object to compare with.

VirtualNodeContext

## Returns

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

## Exceptions

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

## Equals(VirtualNodeName?)

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

### Declaration

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

### Parameters

Type	AkiraNetwork.VirtualStorageLibrary.IVirtualWildcardMatcher	Name	Description
	NotifyNodeDelegate VirtualNodeName (AkiraNetwork.VirtualStorageLibrary (AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html) PatternMatch (AkiraNetwork.VirtualStorageLibrary VirtualCycleDetector (AkiraNetwork.VirtualStorageLibrary VirtualDirectory	other	The VirtualNodeName (AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html) to compare with this instance.

### Returns

VirtualDirectory

### Type

AkiraNetwork.VirtualStorageLibrary

### Description

```
VirtualException  
bool true if the specified  
(AkiraNetwork.VirtualStorageLibrary  
(https://learn.microsoft.com/dotnet/VirtualNodeName  
VirtualGroupCondition<T, TKey>  
net/api/system.boolean)  
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)  
(AkiraNetwork.VirtualStorageLibrary  
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

(AkiraNetwork.VirtualStorageLibrary

### Parameters

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

### Returns

VirtualNodeName

(AkiraNetwork.VirtualStorageLibrary

### Type

AkiraNetwork.VirtualStorageLibrary.VirtualNodeNotFoundException

### Description

true if the specified object is equal to this instance; otherwise, false .

```
bool  
(AkiraNetwork.VirtualStorageLibrary  
(https://learn.microsoft.com/dotnet/api/system.bool  
VirtualNodeType  
(AkiraNetwork.VirtualStorageLibrary  
olean)
```

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

Thrown when the prefix is an empty string.

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

1.html)

VirtualNode

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeContext

(AkiraNetwork.VirtualStorageLibrary)

Returns a hash code for this instance.

VirtualNodeExtensions

Declaration

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeListConditions

public static int GetHashCode()

VirtualNodeName

(AkiraNetwork.VirtualStorageLibrary)

Returns

VirtualNodeNotFoundException

Type (AkiraNetwork.VirtualStorageLibrary)

Description

VirtualNodeType

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

A hash code for this instance.

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeTypeFilter

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 (AkiraNetwork.VirtualStorageLibrary.VirtualnodeName.html)	nodeName	The node name to check.

PatternMatch  
**Returns**

Type	Description
VirtualCycleDetector (AkiraNetwork.VirtualStorageLibrary.VirtualDirectory.html)	true if the node name is valid; otherwise, false .

VirtualException

(AkiraNetwork.VirtualStorageLibrary.VirtualCycleDetector.html)

VirtualGroupCondition<T, TKey>

(AkiraNetwork.VirtualStorageLibrary.VirtualGroupCondition.html)

**ResetCounter()**  
Resets the counter used for node name generation.  
2.html)

VirtualID

(AkiraNetwork.VirtualStorageLibrary.VirtualID.html)

VirtualItem

public static void ResetCounter()  
(AkiraNetwork.VirtualStorageLibrary.VirtualItem.html)

VirtualItem<T>

(AkiraNetwork.VirtualStorageLibrary.VirtualItem.html)

**ToString()**  
VirtualNode

Returns the name of the node as a string.

VirtualNodeContext

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

VirtualNodeExtensions

public static string ToString()  
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeExtensions.html)

VirtualNodeListConditions

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeListConditions.html)

**Returns**  
VirtualnodeName

Type	Description
AkiraNetwork.VirtualStorageLibrary.VirtualnodeName (AkiraNetwork.VirtualStorageLibrary.VirtualnodeName.html)	The name of the node as a string.

VirtualNodeType

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

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

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

**Operators**

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.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)
1.html)

```

### Parameters

Type	Name	Description
VirtualnodeName (AkiraNetwork.VirtualStorageLibrary.VirtualnodeName.html) VirtualPatternMatch (AkiraNetwork.VirtualStorageLibrary.VirtualnodeName.html) VirtualCycleDetector (AkiraNetwork.VirtualStorageLibrary.VirtualnodeName.html)	left	The left VirtualnodeName (AkiraNetwork.VirtualStorageLibrary.VirtualnodeName.html) to compare.
VirtualnodeName (AkiraNetwork.VirtualStorageLibrary.VirtualnodeName.html) VirtualGroupCondition<T, TKey> (AkiraNetwork.VirtualStorageLibrary.VirtualnodeName.html)	right	The right VirtualnodeName (AkiraNetwork.VirtualStorageLibrary.VirtualnodeName.html) to compare.

### Returns

Type	Description
VirtualID (AkiraNetwork.VirtualStorageLibrary.VirtualID.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.VirtualNodeContext)

### Declaration

```

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

```

### Parameters

Type	Name	Description
VirtualNotFoundException (AkiraNetwork.VirtualStorageLibrary.VirtualNotFoundException.html)	nodeName	The VirtualnodeName (AkiraNetwork.VirtualStorageLibrary.VirtualnodeName.html) to convert.

```

(AkiraNetwork.VirtualStorageLibrary.VirtualPath)
1.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 VirtualNodeName (AkiraNetwork.VirtualStorageLibrary) (AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)	<i>left</i>	The left VirtualNodeName (AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html) to compare.
VirtualNodeType (AkiraNetwork.VirtualStorageLibrary) VirtualNodeTypeFilter (AkiraNetwork.VirtualStorageLibrary) VirtualPath (AkiraNetwork.VirtualStorageLibrary)	<i>right</i>	The right VirtualNodeName (AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html) to compare.

## Returns

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

## Implements

IComparable<T> ( <a href="https://learn.microsoft.com/dotnet/api/system.icomparable-1">https://learn.microsoft.com/dotnet/api/system.icomparable-1</a> )	
INotifyNodeDelegate ( <a href="https://learn.microsoft.com/dotnet/api/system.inotifynodedelegate">https://learn.microsoft.com/dotnet/api/system.inotifynodedelegate</a> )	
IPatternMatch ( <a href="https://learn.microsoft.com/dotnet/api/AkiraNetwork.VirtualStorageLibrary.PatternMatch">https://learn.microsoft.com/dotnet/api/AkiraNetwork.VirtualStorageLibrary.PatternMatch</a> )	
IVirtualCycleDetector ( <a href="https://learn.microsoft.com/dotnet/api/AkiraNetwork.VirtualStorageLibrary.VirtualCycleDetector">https://learn.microsoft.com/dotnet/api/AkiraNetwork.VirtualStorageLibrary.VirtualCycleDetector</a> )	
VirtualTextFormatter.GenerateSingleTableDebugText<T>(T) ( <a href="https://learn.microsoft.com/dotnet/api/AkiraNetwork.VirtualStorageLibrary.Utilities.VirtualTextFormatter.html#AkiraNetwork_VirtualStorageLibrary_Utilsitics_VirtualTextFormatter_GenerateSingleTableDebugText_1__0_">https://learn.microsoft.com/dotnet/api/AkiraNetwork.VirtualStorageLibrary.Utilities.VirtualTextFormatter.html#AkiraNetwork_VirtualStorageLibrary_Utilsitics_VirtualTextFormatter_GenerateSingleTableDebugText_1__0_</a> )	
VirtualException ( <a href="https://learn.microsoft.com/dotnet/api/AkiraNetwork.VirtualStorageLibrary.VirtualException">https://learn.microsoft.com/dotnet/api/AkiraNetwork.VirtualStorageLibrary.VirtualException</a> )	
VirtualGroupCondition<T, TKey> ( <a href="https://learn.microsoft.com/dotnet/api/AkiraNetwork.VirtualStorageLibrary.VirtualGroupCondition_1">https://learn.microsoft.com/dotnet/api/AkiraNetwork.VirtualStorageLibrary.VirtualGroupCondition_1</a> )	
VirtualID ( <a href="https://learn.microsoft.com/dotnet/api/AkiraNetwork.VirtualStorageLibrary.VirtualID">https://learn.microsoft.com/dotnet/api/AkiraNetwork.VirtualStorageLibrary.VirtualID</a> )	
VirtualItem ( <a href="https://learn.microsoft.com/dotnet/api/AkiraNetwork.VirtualStorageLibrary.VirtualItem">https://learn.microsoft.com/dotnet/api/AkiraNetwork.VirtualStorageLibrary.VirtualItem</a> )	
VirtualItem<T> ( <a href="https://learn.microsoft.com/dotnet/api/AkiraNetwork.VirtualStorageLibrary.VirtualItem_1">https://learn.microsoft.com/dotnet/api/AkiraNetwork.VirtualStorageLibrary.VirtualItem_1</a> )	
VirtualNode ( <a href="https://learn.microsoft.com/dotnet/api/AkiraNetwork.VirtualStorageLibrary.VirtualNode">https://learn.microsoft.com/dotnet/api/AkiraNetwork.VirtualStorageLibrary.VirtualNode</a> )	
VirtualNodeContext ( <a href="https://learn.microsoft.com/dotnet/api/AkiraNetwork.VirtualStorageLibrary.VirtualNodeContext">https://learn.microsoft.com/dotnet/api/AkiraNetwork.VirtualStorageLibrary.VirtualNodeContext</a> )	
VirtualNodeExtensions ( <a href="https://learn.microsoft.com/dotnet/api/AkiraNetwork.VirtualStorageLibrary.VirtualNodeExtensions">https://learn.microsoft.com/dotnet/api/AkiraNetwork.VirtualStorageLibrary.VirtualNodeExtensions</a> )	
VirtualNodeListConditions ( <a href="https://learn.microsoft.com/dotnet/api/AkiraNetwork.VirtualStorageLibrary.VirtualNodeListConditions">https://learn.microsoft.com/dotnet/api/AkiraNetwork.VirtualStorageLibrary.VirtualNodeListConditions</a> )	
VirtualnodeName ( <a href="https://learn.microsoft.com/dotnet/api/AkiraNetwork.VirtualStorageLibrary.VirtualnodeName">https://learn.microsoft.com/dotnet/api/AkiraNetwork.VirtualStorageLibrary.VirtualnodeName</a> )	
VirtualNodeNotFoundException ( <a href="https://learn.microsoft.com/dotnet/api/AkiraNetwork.VirtualStorageLibrary.VirtualNodeNotFoundException">https://learn.microsoft.com/dotnet/api/AkiraNetwork.VirtualStorageLibrary.VirtualNodeNotFoundException</a> )	
VirtualNodeType ( <a href="https://learn.microsoft.com/dotnet/api/AkiraNetwork.VirtualStorageLibrary.VirtualNodeType">https://learn.microsoft.com/dotnet/api/AkiraNetwork.VirtualStorageLibrary.VirtualNodeType</a> )	
VirtualNodeTypeFilter ( <a href="https://learn.microsoft.com/dotnet/api/AkiraNetwork.VirtualStorageLibrary.VirtualNodeTypeFilter">https://learn.microsoft.com/dotnet/api/AkiraNetwork.VirtualStorageLibrary.VirtualNodeTypeFilter</a> )	
VirtualPath ( <a href="https://learn.microsoft.com/dotnet/api/AkiraNetwork.VirtualStorageLibrary.VirtualPath">https://learn.microsoft.com/dotnet/api/AkiraNetwork.VirtualStorageLibrary.VirtualPath</a> )	

# 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>)

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

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

VirtualException

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

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

VirtualMessage

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

VirtualSource

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

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

VirtualTarget

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

Exception (ObjectState) (<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)))

(AkiraNetwork.VirtualStorageLibrary)

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.VirtualStorageLibrary)

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

(AkiraNetwork.VirtualStorageLibrary)

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

(AkiraNetwork.VirtualStorageLibrary)

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>)

(AkiraNetwork.VirtualStorageLibrary)

**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()

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.

VirtualNodeDelegate

(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

message

The error message.

## VirtualNodeNotFoundException(string, Exception)

(AkiraNetwork.VirtualStorageLibrary)

Initializes a new instance of the VirtualNodeNotFoundException

VirtualItem

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

(AkiraNetwork.VirtualStorageLibrary)

VirtualItem<T>

Declaration

(AkiraNetwork.VirtualStorageLibrary)

1.html

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

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeContext

Parameters

(AkiraNetwork.VirtualStorageLibrary)

Type

VirtualNodeExtensions

(AkiraNetwork.VirtualStorageLibrary)

string

VirtualNodeListConditions

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

(AkiraNetwork.VirtualStorageLibrary)

Name

Description

message

The error message.

Exception

(AkiraNetwork.VirtualStorageLibrary)

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

VirtualNodeNotFoundException

(AkiraNetwork.VirtualStorageLibrary)

innerException

The exception that is the cause of this exception.

## Implements

(AkiraNetwork.VirtualStorageLibrary)

ISerializable

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

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.

# 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 (AkiraNetwork.VirtualStorageLibrary)	
VirtualException (AkiraNetwork.VirtualStorageLibrary)	Represents a directory node.
Item (AkiraNetwork.VirtualStorageLibrary)	Represents an item node with a user-defined type T.
VirtualGroupCondition<T, TKey> (AkiraNetwork.VirtualStorageLibrary)	
None (AkiraNetwork.VirtualStorageLibrary)	Represents an unspecified or unknown node type.
SymbolicLink VirtualID (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. 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 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

Syntax

(AkiraNetwork.VirtualStorageLibrary)  
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

Declaration

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeNotFoundException

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

VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary)

Parameters

VirtualNodeTypeFilter

(AkiraNetwork.VirtualStorageLibrary)

VirtualPath

(AkiraNetwork.VirtualStorageLibrary)

Type	Name	Description
IEnumerable ( <a href="https://learn.microsoft.com/dotnet/api/system.collections.generic.ienumerable-1">https://learn.microsoft.com/dotnet/api/system.collections.generic.ienumerable-1</a> ) <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	VirtualDirectory (AkiraNetwork.VirtualStorageLibrary)	Name	path
	string ( <a href="https://learn.microsoft.com/dotnet/api/system.string">https://learn.microsoft.com/dotnet/api/system.string</a> )		The initial path string.

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

## 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)
```

### Type

```
(AkiraNetwork.VirtualStorageLibrary)
```

### Description

An integer representing the depth of the path.

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

```
(AkiraNetwork.VirtualStorageLibrary)
```

```
VirtualnodeName
```

```
(AkiraNetwork.VirtualStorageLibrary)
```

```
VirtualNodeNotFoundException
```

Gets the directory part of the path.

```
VirtualNodeType
```

### Declaration

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

## Property Value

Type	Description
VirtualPath (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 (AkiraNetwork.VirtualStorageLibrary)	A string representing the dot symbol.

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

VirtualNodeException  
(AkiraNetwork.VirtualStorageLibrary)

VirtualGroupCondition<T, TKey>  
(AkiraNetwork.VirtualStorageLibrary)

### DotDot

2.html)

VirtualDoubleDot  
Gets the double dot ("..") symbol.  
(AkiraNetwork.VirtualStorageLibrary)

Declaration

VirtualItem  
(AkiraNetwork.VirtualStorageLibrary)

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

1.html)

Property Value

Type	Description
VirtualNode (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

FixedDepthListConditions  
(AkiraNetwork.VirtualStorageLibrary)

Gets the depth of the fixed path.

VirtualNodeName

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeNotFoundException

(AkiraNetwork.VirtualStorageLibrary)

public int FixedDepth { get; }

VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary)

Property Value

TypeFilter  
(AkiraNetwork.VirtualStorageLibrary)

VirtualPath  
(AkiraNetwork.VirtualStorageLibrary)



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

## Property Value

Type	Description
bool ( <a href="https://learn.microsoft.com/dotnet/api/system.boolean">https://learn.microsoft.com/dotnet/api/system.boolean</a> ) 1.html) IVirtualWildcardMatcher (AkiraNetwork.VirtualStorageLibrary NotifyNodeDelegate (AkiraNetwork.VirtualStorageLibrary 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	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 VirtualGroupCondition<T, TKey> bool (AkiraNetwork.VirtualStorageLibrary ( <a href="https://learn.microsoft.com/dotnet/api/system.boolean">https://learn.microsoft.com/dotnet/api/system.boolean</a> ) n) VirtualID (AkiraNetwork.VirtualStorageLibrary VirtualItem (AkiraNetwork.VirtualStorageLibrary IsEmpty VirtualItem<T> Gets a value indicating whether the instance's path is empty. 1.html) Declaration VirtualNode (AkiraNetwork.VirtualStorageLibrary public bool IsEmpty { get; } VirtualNodeContext (AkiraNetwork.VirtualStorageLibrary VirtualNode VirtualNodeExtensions Type VirtualNodeListConditions bool (AkiraNetwork.VirtualStorageLibrary ( <a href="https://learn.microsoft.com/dotnet/api/system.boolean">https://learn.microsoft.com/dotnet/api/system.boolean</a> ) (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 Property Value VirtualException	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  
Property Value  
VirtualException

## IsEndsWithSlash

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

VirtualNodeTypeFilter  
Declaration  
(AkiraNetwork.VirtualStorageLibrary  
VirtualPath  
(AkiraNetwork.VirtualStorageLibrary  
Property Value  
VirtualException

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

## Property Value

Type	Description
bool ( <a href="https://learn.microsoft.com/dotnet/api/system.boolean">https://learn.microsoft.com/dotnet/api/system.boolean</a> ) 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> (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  
(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 (AkiraNetwork.VirtualStorageLibrary)	A VirtualNodeName (AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html) representing the node name of the path.



Gets the path separator character.

## Declaration

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



## Property Value

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

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)  
NotifyNodeDelegate

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)  
VirtualMatch

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)  
VirtualCycleDetector

**AddEndSlash()** (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)  
VirtualDirectory

Adds a trailing slash to the path if it doesn't already have one.  
(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

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

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

2.html)

**Returns** VirtualID  
(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

Type (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)  
VirtualItem

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

VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)  
with a trailing slash.  
1.html)

VirtualNode

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)  
VirtualNodeContext

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)  
VirtualNodeExtensions

Declaration (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)  
VirtualNodeListConditions

```
public VirtualPath AddEndSlash()  
VirtualnodeName
```

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)  
VirtualNodeNotFoundException

Type (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)  
VirtualNodeType

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

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)  
VirtualPath

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

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

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)  
VirtualNode

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)  
VirtualMatch

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)  
VirtualCycleDetector

**AddStartSlash()** (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)  
VirtualDirectory

Adds a leading slash to the path if it doesn't already have one.  
(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

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

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

2.html)

**Returns** VirtualID  
(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

Type (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)  
VirtualItem

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

VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)  
with a leading slash.  
1.html)

VirtualNode

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)  
VirtualNodeContext

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)  
VirtualNodeExtensions

Declaration (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)  
VirtualNodeListConditions

```
public VirtualPath AddStartSlash()  
VirtualnodeName
```

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)  
VirtualNodeNotFoundException

Type (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)  
VirtualNodeType

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

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)  
VirtualPath

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

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

Determines whether one path is a subdirectory of another path.

## Declaration

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

Parameters	(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)		
Type	VIRTUALWILDCARDMATCHER	Name	Description
	(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	path1	The first VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)
	(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	path2	The second VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

Retuřák - Akira Network Virtual Storage Library

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

`(AkiraNetwork.VirtualStorageLibrary.Exceptions)`

Exceptions	Type	Condition
<code>VirtualItem&lt;T&gt;</code>	( <a href="#">AkiraNetwork.VirtualStorageLibrary</a> )	
<code>ArgumentException</code> ( <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)

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

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

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 (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	<i>path</i>	The VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) whose parts to combine.
int (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 (AkiraNetwork.VirtualStorageLibrary)	
VirtualPathMatch (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	A new VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) representing the combined path.
VirtualPathCycleDetector	

# 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(object)**

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

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

# Declaration

## VirtualNodeTypeFilter

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

(AkiraNetwork.virtualStorageLibrary)

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

## CompareTo(object?)

Compare object  
(AkiraNetwork.VirtualStorageLibrary)

(AkiraNetwork.VirtualStorageLibrary)

# Declaration

## VirtualNodeTypeFilter

(AkiraNetwork.VirtualStorageLibrary  
VirtualPath 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
(AkiraNetwork.VirtualStorageLibrary.VirtualPathException)	Thrown if the object is not a VirtualPath
ArgumentOutOfRangeException	(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)
PatternMatchException	.
(AkiraNetwork.VirtualStorageLibrary.VirtualPathException)	.

VirtualDirectory  
(AkiraNetwork.VirtualStorageLibrary.VirtualPath)

**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

Type	Description
bool	true if the specified VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) is equal to the instance's path; otherwise, false.

VirtualNodeNotFoundException  
(AkiraNetwork.VirtualStorageLibrary.VirtualPath)

VirtualNodeType  
(AkiraNetwork.VirtualStorageLibrary.VirtualPath)

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

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

Declaration  
VirtualPath  
(AkiraNetwork.VirtualStorageLibrary.VirtualPath)

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

#### Parameters

Type	Name	Description
object ( <a href="https://learn.microsoft.com/dotnet/api/system.object">https://learn.microsoft.com/dotnet/api/system.object</a> ) 1.html)	obj	The object to compare with the instance'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> ) ( <a href="https://learn.microsoft.com/dotnet/api/system.boolean">https://learn.microsoft.com/dotnet/api/system.boolean</a> ) VirtualCycleDetector	true if the specified object is equal to the instance's path; otherwise, false.

#### Overrides

VirtualDirectory 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)
VirtualItem
public override int GetHashCode() ( <a href="https://learn.microsoft.com/dotnet/api/system.int32">https://learn.microsoft.com/dotnet/api/system.int32</a> )

Declaration
(AkiraNetwork.VirtualStorageLibrary)
VirtualItem<T>
public override int GetHashCode() ( <a href="https://learn.microsoft.com/dotnet/api/system.int32">https://learn.microsoft.com/dotnet/api/system.int32</a> )

Returns
1.html)

Type	Description
VirtualNode	

int ( <a href="https://learn.microsoft.com/dotnet/api/system.int32">https://learn.microsoft.com/dotnet/api/system.int32</a> )	The hash code for this path.
---	------------------------------

VirtualNodeContext	
--------------------	--

(AkiraNetwork.VirtualStorageLibrary)	
--------------------------------------	--

VirtualNodeExtensions	
-----------------------	--

object.GetHashCode() ( <a href="https://learn.microsoft.com/dotnet/api/system.object.gethashcode">https://learn.microsoft.com/dotnet/api/system.object.gethashcode</a> )	
--	--

(AkiraNetwork.VirtualStorageLibrary)	
--------------------------------------	--

VirtualNodeListConditions	
---------------------------	--

(AkiraNetwork.VirtualStorageLibrary)	
--------------------------------------	--

VirtualNodeName	
-----------------	--

(AkiraNetwork.VirtualStorageLibrary)	
--------------------------------------	--

Retrieves the VirtualStorageLibrary instance's path.	
--	--

VirtualNodeNotFoundException	
------------------------------	--

(AkiraNetwork.VirtualStorageLibrary)	
--------------------------------------	--

VirtualNodeType	
-----------------	--

public VirtualNodeName GetNodeName() ( <a href="https://learn.microsoft.com/dotnet/api/system.int32">https://learn.microsoft.com/dotnet/api/system.int32</a> )	
--	--

VirtualNodeTypeFilter	
-----------------------	--

(AkiraNetwork.VirtualStorageLibrary)	
--------------------------------------	--

VirtualPath	
-------------	--

(AkiraNetwork.VirtualStorageLibrary)	
--------------------------------------	--

Type	Description
VirtualNodeName (AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html)	A VirtualNodeName (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 (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
IVirtualWildcardMatcher List<VirtualnodeName> (AkiraNetwork.VirtualStorageLibrary) (https://learn.microsoft.com/dotnet/api/system.collections.generic.list-1) < NotifyNodeDelegate m.collections.generic.list-1> (AkiraNetwork.VirtualStorageLibrary) < PatternMatch VirtualnodeName (AkiraNetwork.VirtualStorageLibrary) (AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html) < VirtualCycleDetector odeName.html> (AkiraNetwork.VirtualStorageLibrary) > VirtualDirectory (AkiraNetwork.VirtualStorageLibrary) VirtualException (AkiraNetwork.VirtualStorageLibrary) VirtualGroupCondition<T, TKey>	A List<T> (https://learn.microsoft.com/dotnet/api/system.collections.generic.list-1) of VirtualnodeName (AkiraNetwork.VirtualStorageLibrary.VirtualnodeName.html) 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
```

(AkiraNetwork.VirtualStorageLibrary)

VirtualItem<T>

(AkiraNetwork.VirtualStorageLibrary)

Type  
1.html

Type	Name	Description
VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	basePath	The base path.

VirtualNodeContext

(AkiraNetwork.VirtualStorageLibrary)

Type  
virtualNodeExtensions

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

Exceptions  
VirtualNodeNotFoundException

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeTypeFilter

(AkiraNetwork.VirtualStorageLibrary)

VirtualPath

(AkiraNetwork.VirtualStorageLibrary)

Type	Condition
InvalidOperationException ( <a href="https://learn.microsoft.com/dotnet/api/system.invalidoperationexception">https://learn.microsoft.com/dotnet/api/system.invalidoperationexception</a> ) (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. 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> VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) VirtualID (AkiraNetwork.VirtualStorageLibrary.VirtualItem VirtualItem	<i>parentPath</i>	The parent VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) . . .

#### Return

(AkiraNetwork.VirtualStorageLibrary.VirtualItem<T>

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

#### Exceptions

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeExtensions  
VirtualNodeException

Type	Condition
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeListConditions ArgumentOutOfRangeException ( <a href="https://learn.microsoft.com/dotnet/api/system.argumentexception">https://learn.microsoft.com/dotnet/api/system.argumentexception</a> ) VirtualNodeName (AkiraNetwork.VirtualStorageLibrary.VirtualNodeName VirtualNodeNotFoundException (AkiraNetwork.VirtualStorageLibrary.VirtualNodeName 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 (AkiraNetwork.VirtualStorageLibrary.VirtualPath VirtualNodeTypeFilter)	

#### Declaration

(AkiraNetwork.VirtualStorageLibrary.VirtualPath

(AkiraNetwork.VirtualStorageLibrary.VirtualPath)

```
public VirtualPath NormalizePath()
```

## Returns

Type	Description
VirtualPath (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> (AkiraNetwork.VirtualStorageLibrary)	path	The path string to normalize.

VirtualID  
Returns  
(AkiraNetwork.VirtualStorageLibrary)

Type	Description
VirtualItem (AkiraNetwork.VirtualStorageLibrary)	A normalized path string.

VirtualItem<T>  
(AkiraNetwork.VirtualStorageLibrary)

Exceptions  
(AkiraNetwork.VirtualStorageLibrary)

Type	Condition
InvalidOperationContext (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 (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	path	The VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) to compare with.

## Returns

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

Type	Description
bool VirtualWildcardMatcher (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.

VirtualDirectory  
(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

VirtualException

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

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

## Returns

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

Type	Description
VirtualID (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

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

VirtualNodeContext

Removes the trailing slash from the path.

AkiraNetwork.VirtualStorageLibrary.VirtualPath.html

VirtualNodeExtensions

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

VirtualNodeListConditions

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

VirtualnodeName

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

VirtualNodeNotFoundException

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

VirtualNodeFilter

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

VirtualPath

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

VirtualPath

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

VirtualPath

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

Type	Description
Virtu	A new VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) without the trailing slash.

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

### Returns

Type	Description
VirtualPath (AkiraNetwork.VirtualStorageLibrary. VirtualPath VirtualPath.html) (AkiraNetwork.VirtualStorageLibrary.Virtu alPath.html) (AkiraNetwork.VirtualStorageLibrary. VirtualNodeContext (AkiraNetwork.VirtualStorageLibrary. VirtualNodeExtensions	A new VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) 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.

Type	Name	Description
VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	<i>path1</i>	The first VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) .
VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	<i>path2</i>	The second VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) .

Returns [NotifyNodeDelegate](#)

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

VirtualDirectory

(AkiraNetwork.VirtualStorageLibrary)

VirtualException

**operator +(VirtualPath, char)**  
(AkiraNetwork.VirtualStorageLibrary)

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

Declaration

VirtualID

p(AkiraNetwork.VirtualStorageLibrary) operator +(VirtualPath path, char chr)

VirtualItem

(AkiraNetwork.VirtualStorageLibrary)

Parameters

VirtualItem<T>

Type	Name	Description
VirtualPath VirtualNode (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	<i>path</i>	The VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) .
VirtualNodeContext  VirtualNodeExtensions (https://realnetworks.com/api/system. charVirtualNodeListConditions (AkiraNetwork.VirtualStorageLibrary)	<i>chr</i>	The character to combine with the path.

VirtualnodeName

(AkiraNetwork.VirtualStorageLibrary)

**Type** VirtualNodeNotFoundException

VirtualPath

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeTypeFilter

(AkiraNetwork.VirtualStorageLibrary)

VirtualPath

(AkiraNetwork.VirtualStorageLibrary)

**operator +(VirtualPath, string)**

(AkiraNetwork.VirtualStorageLibrary)

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

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) VirtualPath VirtualWildcardMatcher (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) NotifyNodeDelegate (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) PatternMatch (https://learn.microsoft.com/en-us/dotnet/api/system.string) VirtualCycleDetector (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) 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 (AkiraNetwork.VirtualStorageLibrary.VirtualPath)	chr	The character to combine with the path.
VirtualNodeListConditions (https://learn.microsoft.com/en-us/dotnet/api/system.chatVirtualNodeName)	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 (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	A new VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) representing the combined path.

▼

## operator +(string VirtualPath)

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

VirtualWildcardMatcher

(AkiraNetwork.VirtualStorageLibrary)

NotifyNodeDelegate

(AkiraNetwork.VirtualStorageLibrary)

public static VirtualPath operator +(string str, VirtualPath path)

PatternMatch

(AkiraNetwork.VirtualStorageLibrary)

Parameters

VirtualCycleDetector

Type	Name	Description
VirtualDirectory (AkiraNetwork.VirtualStorageLibrary)	str	The string to combine with the path.
VirtualPath (AkiraNetwork.VirtualStorageLibrary)	path	The VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

Returns

(AkiraNetwork.VirtualStorageLibrary)

Type VirtualItem<T>

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

VirtualNodeContext

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeFilter

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

VirtualNodeListConditions

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeName

(AkiraNetwork.VirtualStorageLibrary)

public static bool operator ==(VirtualPath? left, VirtualPath? right)

VirtualNodeNotFoundException

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeTypeFilter

(AkiraNetwork.VirtualStorageLibrary)

VirtualPath

(AkiraNetwork.VirtualStorageLibrary)

Type	Name	Description
VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	<i>left</i>	The left VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) .
VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	<i>right</i>	The right VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) .

Returns [NotifyNodeDelegate](#)

Type	Description
PatternMatch 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>

Declaration

(AkiraNetwork.VirtualStorageLibrary)

2.html)

public static implicit operator string(VirtualPath? virtualPath)

(AkiraNetwork.VirtualStorageLibrary)

VirtualItem

(AkiraNetwork.VirtualStorageLibrary)

Parameters

(AkiraNetwork.VirtualStorageLibrary)

Type	Name	Description
VirtualItem<T> (AkiraNetwork.VirtualStorageLibrary) VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	<i>virtualPath</i>	The VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) object.

VirtualNodeContext

(AkiraNetwork.VirtualStorageLibrary)

Returns

VirtualNodeExtensions

Type	Description
VirtualNodeListConditions string (AkiraNetwork.VirtualStorageLibrary) ( <a href="https://learn.microsoft.com/dotnet/api/system.string">https://learn.microsoft.com/dotnet/api/system.string</a> )	The string representation of the VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html), or an empty string if the path is null. ( <a href="https://learn.microsoft.com/dotnet/api/system.string">https://learn.microsoft.com/dotnet/api/system.string</a> )

VirtualNodeNotFoundException

(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 (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.

## Declaration

```
public static bool operator !=(VirtualPath? left, VirtualPath? right)  
2.html)
```

## Parameters

Type	Name	Description
VirtualItem (AkiraNetwork.VirtualStorageLibrary)	left	The left VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)
VirtualPath VirtualItem<T> (AkiraNetwork.VirtualStorageLibrary)	right	The right VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

## Returns

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

## Implements

IEquatable<T>	<a href="https://learn.microsoft.com/dotnet/api/system.iequatable-1">https://learn.microsoft.com/dotnet/api/system.iequatable-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 ( <a href="https://learn.microsoft.com/dotnet/api/system.collections.generic.ienumerable-1">https://learn.microsoft.com/dotnet/api/system.collections.generic.ienumerable-1</a> )  <T> ▼	source	The sequence of elements to sort.
List ( <a href="https://learn.microsoft.com/dotnet/api/system.collections.generic.list-1">https://learn.microsoft.com/dotnet/api/system.collections.generic.list-1</a> )  (AkiraNetwork.VirtualStorageLibrary) generic.list-1.html  < IVirtualWildcardMatcher VirtualSortCondition (AkiraNetwork.VirtualStorageLibrary) (AkiraNetwork.VirtualStorageLibrary.VirtualSortCondition-1.html) (AkiraNetwork.VirtualStorageLibrary) <T>PatternMatch	sortConditions	A list of conditions that define the sort order. If null or empty, no sorting is applied.

ReturnVirtualCycleDetector

Type	Description
(AkiraNetwork.VirtualStorageLibrary.VirtualDirectory)	
IEnumerable<T>(AkiraNetwork.VirtualStorageLibrary.VirtualGroupCondition<T, TKey>)	An <code>IEnumerable&lt;T&gt;</code> containing the sorted elements.
(https://learn.microsoft.com/dotnet/api/system.collections.generic.ienumerable-1)	

2.html)

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

**GroupAndSort<T>(IEnumerable<T>, VirtualGroupCondition<T, object>?,**

**List<VirtualSortCondition<T>>?**

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

© AkiraNetwork Virtualization

(AlienNet and VirtuosoGTC) and libraries

```
(AkiraNetwork.VirtualStorageLibrary)
public static List<VirtualSortCondition<T>> GroupAndSort<T>(this IEnumerable<T> source, VirtualGroupCondition<T> condition, VirtualSortCondition<T> sortCondition) {
```

VirtualNodeName

(AkiraNetwork.VirtualStorageLibrary)

**Parameters** VirtualNodeNotFoundException

Type	(AkiraNetwork.VirtualStorageLibrary.IVirtualNodeFilter<T>)	Name	Description
	<p>VirtualNodeType IEnumerable&lt;(AkiraNetwork.VirtualStorageLibrary.IVirtualNodeFilter&lt;T&gt; source, string VirtualPath)&gt; (https://learn.microsoft.com/dotnet/api/system.collection.generic.ienumerable-1) &lt;T&gt; VirtualPath (AkiraNetwork.VirtualStorageLibrary.IVirtualNodeFilter&lt;T&gt;)</p>	source	The sequence of elements to group and sort.

Type	Name	Description
<p>VirtualGroupCondition (AkiraNetwork.VirtualStorageLibrary.VirtualGroupCondition- n-2.html)</p> <p>&lt;T, object (<a href="https://learn.microsoft.com/dotnet/api/system.object">https://learn.microsoft.com/dotnet/api/system.object</a>) &gt; (AkiraNetwork.VirtualStorageLibrary- 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 (<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>) NotifyNodeDelegate &lt; (AkiraNetwork.VirtualStorageLibrary- VirtualSortCondition PatternMatcher (AkiraNetwork.VirtualStorageLibrary.VirtualSortCondition- 1.html) VirtualCycleDetector &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
IEnumerable<T> ( <a href="https://learn.microsoft.com/dotnet/api/system.collections.generic.ienumerable-1">https://learn.microsoft.com/dotnet/api/system.collections.generic.ienumerable-1</a> ) <T>VirtualID (AkiraNetwork.VirtualStorageLibrary)	An IEnumerable<T> ( <a href="https://learn.microsoft.com/dotnet/api/system.collections.generic.ienumerable-1">https://learn.microsoft.com/dotnet/api/system.collections.generic.ienumerable-1</a> ) containing the 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

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

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**  
(AkiraNetwork.VirtualStorageLibrary  
1.html)

Type	Description
VirtualWildcardMatcher	An array of node names that are not allowed in the virtual storage.

## NodeListConditions

Gets or sets the conditions for listing nodes.

### Declaration

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

VirtualID

**Property Value**  
(AkiraNetwork.VirtualStorageLibrary  
1.html)

Type	Description
VirtualItem	The conditions used to filter, group, and sort nodes.

## PathDot

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

### Declaration

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

VirtualNodeName

**Property Value**  
(AkiraNetwork.VirtualStorageLibrary  
1.html)

Type	Description
VirtualNodeType	The representation of the current directory.

VirtualNodeTypeFilter

**PathDotDot**  
(AkiraNetwork.VirtualStorageLibrary  
1.html)



## Property Value

Type	Description
string ( <a href="https://learn.microsoft.com/dotnet/api/system.string">https://learn.microsoft.com/dotnet/api/system.string</a> )	The prefix added to auto-generated directory names.

▼

## PrefixItem

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 string ( <a href="https://learn.microsoft.com/dotnet/api/system.string">https://learn.microsoft.com/dotnet/api/system.string</a> )	The prefix added to auto-generated 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 string ( <a href="https://learn.microsoft.com/dotnet/api/system.string">https://learn.microsoft.com/dotnet/api/system.string</a> ) VirtualNodeExtensions	The prefix added to auto-generated symbolic link names.

## Settings

Gets the current instance of the settings.  
VirtualNodeName  
(AkiraNetwork.VirtualStorageLibrary)

### Declaration

```
VirtualNodeNotFoundException  
(AkiraNetwork.VirtualStorageLibrary  
VirtualNodeType  
public static VirtualStorageSettings Settings { get; }  
(AkiraNetwork.VirtualStorageLibrary)
```

### Property Value

Type	Description
VirtualPath (AkiraNetwork.VirtualStorageLibrary)	

Type	Description
VirtualStorageSettings (AkiraNetwork.VirtualStorageLibrary.VirtualStorageSettings.html)	The current instance of VirtualStorageSettings (AkiraNetwork.VirtualStorageLibrary.VirtualStorageSettings.html) .

## (AkiraNetwork.VirtualStorageLibrary 1.html)

## WildcardMatcher

(AkiraNetwork.VirtualStorageLibrary)  
Gets or sets the wildcard matcher.  
NotifyNodeDelegate

## Declaration

```
PatternMatch  
(AkiraNetwork.VirtualStorageLibrary) public IVirtualWildcardMatcher? WildcardMatcher { get; set; }  
VirtualCycleDetector
```

(AkiraNetwork VirtualStorage library)

## Property Value

Type	Description
(AkiraNetwork.VirtualStorageLibrary)	
VirtualException	
IVirtualWildcardMatcher	The implementation of wildcard matching used in the storage.
(AkiraNetwork.VirtualStorageLibrary)	
VirtualGroupConditionKey	
IVirtualWildcardMatcher.html	
(AkiraNetwork.VirtualStorageLibrary)	

# Methods

(AkiraNetwork.VirtualStorageLibrary)

## VirtualItem<T>

ialized)

1.html)

alizes the settings. Existing settings are maintained.

(Akira N)

**Declaration** VirtualNodeContext

Virtual  
(AkiraN)

(AkiraNetwork.Virtu  
public static void

## VirtualNodeExtensions (AkiraNetwork.VirtualStorageLibrary)

## VirtualNodeListConditions

# Extension Method

VirtualTextFormatter.GenerateSingleTab

VirtualNetwork.VirtuS

```
    s VirtualTextFormatter.GenerateSingleT
```

## **VirtualNodeType**

## VirtuaNode type (AkiraNetwork VirtualStorage library)

Akka.NET

`VirtualNodeFilter`  
(AkiraNetwork.VirtualStorageLibrary)

#### **VirtualPath**

(AkiraNetwork VirtualStorage) library

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

**Assembly** [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 (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> ) (AkiraNetwork.VirtualStorageLibrary 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
VirtualException (AkiraNetwork.VirtualStorageLibrary string ( <a href="https://learn.microsoft.com/dotnet/api/system.string">https://learn.microsoft.com/dotnet/api/system.string</a> ) VirtualGroupCondition<T, TKey> (AkiraNetwork.VirtualStorageLibrary 2.html)	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  
VirtualNode.cs)
```

## Property Value

Type	Description
VirtualNodeContext (AkiraNetwork.VirtualStorageLibrary char ( <a href="https://learn.microsoft.com/dotnet/api/system.char">https://learn.microsoft.com/dotnet/api/system.char</a> ) (AkiraNetwork.VirtualStorageLibrary VirtualNode.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  
VirtualPath  
(AkiraNetwork.VirtualStorageLibrary  
VirtualPath.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

### Property Value

Type	Description
(AkiraNetwork.VirtualStorageLibrary)	

VirtualStorageState (AkiraNetwork.VirtualStorageLibrary.VirtualStorageState.html)

VirtualPath

(AkiraNetwork.VirtualStorageLibrary)



Type	Name	Description
VirtualNodeTypeFilter (AkiraNetwork.VirtualStorageLibrary.VirtualNodeTypeFilter.html)	<i>filter</i>	The filter for node types.
VirtualGroupCondition (AkiraNetwork.VirtualStorageLibrary.VirtualGroupCondition-2.html) <VirtualNode (AkiraNetwork.VirtualStorageLibrary.VirtualNode.html), object (https://learn.microsoft.com/dotnet/api/system.object)> (AkiraNetwork.VirtualStorageLibrary.VirtualGroupCondition-2.html)	<i>groupCondition</i>	The condition for grouping nodes.
List (https://learn.microsoft.com/dotnet/api/system.collections.generic.list-1) IVirtualWildcardMatcher < (AkiraNetwork.VirtualStorageLibrary.VirtualWildcardMatcher.html) NotifyNodeDelegate < (AkiraNetwork.VirtualStorageLibrary.NotifyNodeDelegate.html) VirtualSortCondition (AkiraNetwork.VirtualStorageLibrary.VirtualSortCondition-1.html) PatternMatch <VirtualNode (AkiraNetwork.VirtualStorageLibrary.VirtualNode.html)>> (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>

(AkiraNetwork.VirtualStorageLibrary

## Inherited Members

NotifyNodeDelegate

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

PatternMatch

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>)

(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 (<https://learn.microsoft.com/akiranetwork.html>).VirtualStorageLibrary (<https://learn.microsoft.com/akiranetwork.html>).VirtualStorageLibrary.html)

Assembly AkiraNetwork.VirtualStorageLibrary

2.html)

## Syntax

VirtualID

(AkiraNetwork.VirtualStorageLibrary

public class VirtualStorage<T>

VirtualItem

(AkiraNetwork.VirtualStorageLibrary

Type Parameters T

(AkiraNetwork.VirtualStorageLibrary

1.html)

T VirtualNode The user-defined type within the virtual storage

(AkiraNetwork.VirtualStorageLibrary

VirtualNodeContext

(AkiraNetwork.VirtualStorageLibrary

VirtualNodeExtensions

(AkiraNetwork.VirtualStorageLibrary

## Constructors

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

(AkiraNetwork.VirtualStorageLibrary

## CurrentPath

Gets or sets the current virtual path.

## Declaration

```
    public VirtualPath CurrentPath { get; }
```

Property	Value	
Type	VirtualWildcardMatcher	Description
	(AkiraNetwork.VirtualStorageLibrary.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.  
(AIR\_SDK\_VirtualCycleDetector.h)

## 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> (Akiri.Network.VirtualStorageLibrary.VirtualCycleDetector.html)	An instance of the cycle detector class

# CycleDetectorForTree [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

<http://trial.NorthExtGen.com>

(AkiraNetwork VirtualStorage) library

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

Type	Description
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeName)	
(AkiraNetwork.VirtualStorageLibrary.VirtualCycleDetector)	An instance of the cycle detector class
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeNotFoundException)	
(AkiraNetwork.VirtualStorageLibrary.VirtualCycleDetector.html)	
(AkiraNetwork.VirtualStorageLibrary.VirtualStorageLibrary)	

## 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 (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	path	The virtual path
bool IWildcardMatcher (AkiraNetwork.VirtualStorageLibrary. <a href="https://learn.microsoft.com/dotnet/api/system.boolean">https://learn.microsoft.com/dotnet/api/system.boolean</a> ) n (AkiraNetwork.VirtualStorageLibrary.PatternMatch)	followLinks	Flag indicating whether to follow symbolic links

**Property Value**  
(AkiraNetwork.VirtualStorageLibrary)

Type	Description
VirtualCycleDetector (AkiraNetwork.VirtualStorageLibrary.VirtualDirectory) (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>  
(AkiraNetwork.VirtualStorageLibrary)

**Property Value**

Type	Description
VirtualNode (AkiraNetwork.VirtualStorageLibrary.Dictionary 2) VirtualNodeContext <VirtualPath>(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html), VirtualNodeExtensions HashSet<VirtualPath>(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)>> VirtualNodeListConditions (AkiraNetwork.VirtualStorageLibrary)	A dictionary representing the link mappings.

VirtualNodeName

**Root**  
(AkiraNetwork.VirtualStorageLibrary)

VirtualNotFoundException  
Gets the root directory of the virtual storage.  
(AkiraNetwork.VirtualStorageLibrary)

**Declaration**

```
public VirtualDirectory Root { get; }
```

VirtualPath

**Property Value**  
(AkiraNetwork.VirtualStorageLibrary)

Type	Description
VirtualDirectory (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.VirtualDirectory.html)

Parameters

VirtualException

Type	Name	Description
AkiraNetwork.VirtualStorageLibrary.VirtualPath VirtualGroupCondition<T, TKey> (AkiraNetwork.VirtualStorageLibrary.VirtualPath.VirtualGroupCondition.html) VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) VirtualID	directoryPath	The path of the directory to add
AkiraNetwork.VirtualStorageLibrary.VirtualDirectory (AkiraNetwork.VirtualStorageLibrary.VirtualDirectory.html) VirtualItem<T> (AkiraNetwork.VirtualStorageLibrary.VirtualItem.html)	directory	The VirtualDirectory (AkiraNetwork.VirtualStorageLibrary.VirtualDirectory.html) to add
bool VirtualNode (https://learn.microsoft.com/dotnet/api/system.boolean) VirtualNodeContext 1.html)	createSubdirectories	Indicates whether to create intermediate directories

### AddDirectory(VirtualPath, bool)

VirtualNodeListConditions

AddDirectory

(AkiraNetwork.VirtualStorageLibrary.VirtualNode.AddDirectory.html)

VirtualnodeName

(AkiraNetwork.VirtualStorageLibrary.VirtualnodeName.html)

VirtualNodeNotFoundException

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeNotFoundException.html)

VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeType.html)

Parameters

VirtualNodeTypeFilter

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeTypeFilter.html)

VirtualPath

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)



Type	Name	Description
VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	<i>itemPath</i>	The path of the item to add
T	<i>data</i>	The data for the item
bool ( <a href="https://learn.microsoft.com/dotnet/api/system.boolean">https://learn.microsoft.com/dotnet/api/system.boolean</a> ) (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	<i>overwrite</i>	Indicates whether to overwrite an existing node

## AddLinkToDictionary(VirtualPath, VirtualPath)

NotifyNodeDelegate  
Adds a new link to the link dictionary.  
(AkiraNetwork.VirtualStorageLibrary)

Declaration  
(AkiraNetwork.VirtualStorageLibrary)

```
public void AddLinkToDictionary(VirtualPath targetPath, VirtualPath linkPath)
(AkiraNetwork.VirtualStorageLibrary)
```

VirtualDirectory

Parameters  
(AkiraNetwork.VirtualStorageLibrary)

Type	Name	Description
VirtualException (AkiraNetwork.VirtualStorageLibrary)		
VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	<i>targetPath</i>	The target path of the link.
VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	<i>linkPath</i>	The path of the link.

VirtualID

Exceptions  
(AkiraNetwork.VirtualStorageLibrary)

Type	Condition
VirtualItem (AkiraNetwork.VirtualStorageLibrary)	
ArgumentException VirtualItem<T> ( <a href="https://learn.microsoft.com/dotnet/api/system.argumentexception">https://learn.microsoft.com/dotnet/api/system.argumentexception</a> ) (AkiraNetwork.VirtualStorageLibrary)	Thrown if the target path is not an absolute path.

1.html)

VirtualNode

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeContext  
Adds a node to the specified directory.

VirtualNodeExtensions

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeListConditions

```
public void AddNode(VirtualPath nodeDirectoryPath, VirtualNode node, bool overwrite = false)
(AkiraNetwork.VirtualStorageLibrary)
```

VirtualnodeName

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeNotFoundException

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeTypeFilter

(AkiraNetwork.VirtualStorageLibrary)

VirtualPath

(AkiraNetwork.VirtualStorageLibrary)

Type	Name	Description
VirtualPath (AkiraNetwork.VirtualStorageLibrary)	<i>nodeDirectoryPath</i>	The path of the directory to add the node to
VirtualPath (AkiraNetwork.VirtualStorageLibrary)		

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



Type	Name	Description
VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	<i>linkDirectoryPath</i>	The path of the directory to add the symbolic link to
VirtualSymbolicLink (AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink.html) IVirtualWildcardMatcher	<i>link</i>	The VirtualSymbolicLink (AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink.html) to add
bool (AkiraNetwork.VirtualStorageLibrary.NotifyNodeDelegate (https://learn.microsoft.com/dotnet/api/system.boolean) PatternMatch (AkiraNetwork.VirtualStorageLibrary.VirtualCycleDetector	<i>overwrite</i>	Indicates whether to overwrite an existing node

## ChangeDirectory(VirtualPath)

VirtualDirectory  
Changes the current working directory.  
(AkiraNetwork.VirtualStorageLibrary)

VirtualException  
(AkiraNetwork.VirtualStorageLibrary)  
VirtualGroupCondition<T, TKey>  
public void ChangeDirectory(VirtualPath path)  
(AkiraNetwork.VirtualStorageLibrary  
2.html)

### Parameters

Type	Name	Description
VirtualItem (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	<i>path</i>	The path of the new working directory

## ConvertToAbsolutePath(VirtualPath?, VirtualPath?)

VirtualNode  
(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeContext  
(AkiraNetwork.VirtualStorageLibrary)

Converts a relative path to an absolute path.  
(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeExtensions  
(AkiraNetwork.VirtualStorageLibrary)  
VirtualNodeListCondition<T> ConvertToAbsolutePath(VirtualPath? relativePath, VirtualPath? basePath = null)  
(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeName  
(AkiraNetwork.VirtualStorageLibrary)

Parameters  
VirtualNodeNotFoundException  
(AkiraNetwork.VirtualStorageLibrary)

Type	Name	Description
VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) VirtualNodeTypeFilter (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	<i>relativePath</i>	The relative path.
VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	<i>basePath</i>	The base path. If not specified, CurrentPath is used.

## Returns

Type	Description
VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	The absolute path represented by VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html).

## Exceptions

Type	Condition
ArgumentException (AkiraNetwork.VirtualStorageLibrary.ArgumentException (https://learn.microsoft.com/dotnet/api/system.argumentexception) NotifyNodeDelegate (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 (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	sourcePath	The source path of the copy.
VirtualItem<T>		
VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	destinationPath	The destination path of the copy.
VirtualNode		
bool (https://learn.microsoft.com/dotnet/api/system.boolean)	overwrite	Whether to overwrite existing nodes.
VirtualNodeContext		
bool (https://learn.microsoft.com/dotnet/api/system.boolean)	recursive	Whether to copy recursively.
bool (https://learn.microsoft.com/dotnet/api/system.boolean)	followLinks	Whether to follow symbolic links.
VirtualNodeName		
List<VirtualNodeContext> (AkiraNetwork.VirtualStorageLibrary)	destinationContextList	List of destination contexts.
< VirtualNodeType		
VirtualNodeContext		
(AkiraNetwork.VirtualStorageLibrary)		
> VirtualNodeTypeFilter		
VirtualPath		

## Exceptions

(AkiraNetwork.VirtualStorageLibrary)

Type	Condition
VirtualNodeNotFoundException (AkiraNetwork.VirtualStorageLibrary.VirtualNodeNotFoundException.html)	Thrown if the source node is not found.
InvalidOperationException (https://learn.microsoft.com/dotnet/api/system.invalidoperationexception) (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> (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	path	The directory's path.
bool (https://learn.microsoft.com/dotnet/api/system.boolean) (AkiraNetwork.VirtualStorageLibrary.Boolean.html)	followLinks	Whether to follow symbolic links.

VirtualID

Returns

VirtualItem

Type	Description
bool (https://learn.microsoft.com/dotnet/api/system.boolean) (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> (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	path	The path.
VirtualNodeTypeFilter (AkiraNetwork.VirtualStorageLibrary.VirtualNodeTypeFilter.html)	filter	The node type filter.
AkiraNetwork.VirtualStorageLibrary.VirtualPath<T, TKey> (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> ) ▼	<i>resolveLinks</i>	Whether to resolve symbolic links.

Return Type: [AkiraNetwork.VirtualStorageLibrary](#)

Type	Description
IVirtualWildcardMatcher <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 Item<T>`

([AkiraNetwork.VirtualStorageLibrary](#))

```
public IEnumerable<VirtualNodeContext> ExpandPathTree(VirtualPath path, VirtualNodeTypeFilter filter, bool followLinks = true, bool resolveLinks = true)
```

([AkiraNetwork.VirtualStorageLibrary](#))

VirtualNodeContext

([AkiraNetwork.VirtualStorageLibrary](#))

Type	Name	Description
<code>virtualNodeExtensions</code>		
( <a href="#">AkiraNetwork.VirtualStorageLibrary</a> )	<i>path</i>	The virtual path containing wildcards
<code>VirtualPath</code>		
( <a href="#">VirtualNodeListConditions</a> )	<i>filter</i>	The filter to apply for node types
( <a href="#">AkiraNetwork.VirtualStorageLibrary</a> )		
<code>VirtualNodeTypeFilter</code>		
( <a href="#">AkiraNetwork.VirtualStorageLibrary</a> )	<i>followLinks</i>	A flag indicating whether to follow symbolic links
<code>bool</code>		
( <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
<code>bool</code>		

`VirtualPath`

([AkiraNetwork.VirtualStorageLibrary](#))

## Returns

Type	Description
IEnumerable ( <a href="https://learn.microsoft.com/dotnet/api/system.collections.generic.ienumerable-1">https://learn.microsoft.com/dotnet/api/system.collections.generic.ienumerable-1</a> ) < VirtualNodeContext (AkiraNetwork.VirtualStorageLibrary.VirtualNodeContext.html) > IVirtualWildcardMatcher (AkiraNetwork.VirtualStorageLibrary.VirtualWildcardMatcher.html) NotifyNodeDelegate (AkiraNetwork.VirtualStorageLibrary.NotifyNodeDelegate.html) followLinks (AkiraNetwork.VirtualStorageLibrary.VirtualStorageLibrary.followLinks.html) PatternMatch (AkiraNetwork.VirtualStorageLibrary.VirtualStorageLibrary.PatternMatch.html) virtualPathIsASymbolicLink (AkiraNetwork.VirtualStorageLibrary.VirtualStorageLibrary.virtualPathIsASymbolicLink.html) resolvesVirtualCycles (AkiraNetwork.VirtualStorageLibrary.VirtualStorageLibrary.resolvesVirtualCycles.html) VirtualDirectory (AkiraNetwork.VirtualStorageLibrary.VirtualDirectory.html)	An enumerable of VirtualNodeContext (AkiraNetwork.VirtualStorageLibrary.VirtualNodeContext.html) 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> (AkiraNetwork.VirtualStorageLibrary.VirtualItem.html)	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> ) (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 VirtualNodeListConditions (AkiraNetwork.VirtualStorageLibrary.VirtualDirectory.html) VirtualDirectoryName (AkiraNetwork.VirtualStorageLibrary.VirtualDirectoryName.html)	The VirtualDirectory (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 (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 (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 (AkiraNetwork.VirtualStorageLibrary.VirtualItem<T>(AkiraNetwork.VirtualStorageLibrary.VirtualItem<T>.VirtualException))	
VirtualDirectory (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 (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 (AkiraNetwork.VirtualStorageLibrary.VirtualPath(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html))	targetPath	The target path.

**Returns**  
VirtualNodeType

Type	Description
VirtualNodeTypeFilter 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 (AkiraNetwork.VirtualStorageLibrary.VirtualNodeType.html)	The VirtualNodeType (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 (AkiraNetwork.VirtualStorageLibrary)	nodeType	The node type filter. (AkiraNetwork.VirtualStorageLibrary.VirtualNodeTypeFilter.html)
bool (https://learn.microsoft.com/dotnet/api/system.boolean.2.html)	recursive	Whether to retrieve nodes recursively.
VirtualID		
bool (AkiraNetwork.VirtualStorageLibrary)	followLinks	Whether to follow symbolic links. (AkiraNetwork.VirtualStorageLibrary.VirtualItem)

Return Value

VirtualItem<T>

Type	Description
IEnumerable<VirtualNode> (http://AkiraNetwork.VirtualStorageLibrary/api/system.collection.generic.ienumerable-1)	The nodes in the path tree starting from the current path IEnumerable<T> (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 IVirtualWildcardMatcher (AkiraNetwork.VirtualStorageLibrary.VirtualNodeTypeFilter.html) (AkiraNetwork.VirtualStorageLibrary.VirtualNodeTypeFilter.html)	nodeType	The node type filter.
bool (https://learn.microsoft.com/dotnet/api/system.boolean) (AkiraNetwork.VirtualStorageLibrary.VirtualCycleDetector)	recursive	Whether to retrieve nodes recursively.
PatternMatch bool (https://learn.microsoft.com/dotnet/api/system.boolean) (AkiraNetwork.VirtualStorageLibrary.VirtualCycleDetector)	followLinks	Whether to follow symbolic links.

## Returns

VirtualDirectory

Type	Description
AkiraNetwork.VirtualStorageLibrary.VirtualException IEnumerable<VirtualNode> (AkiraNetwork.VirtualStorageLibrary.VirtualNode.html) (https://learn.microsoft.com/dotnet/api/system.collections.generic.ienumerable-1) (AkiraNetwork.VirtualStorageLibrary.VirtualNode.html) < 2.html> VirtualNode VirtualNode (AkiraNetwork.VirtualStorageLibrary.VirtualNode.html) ) VirtualItem > (AkiraNetwork.VirtualStorageLibrary.VirtualItem<T> (AkiraNetwork.VirtualStorageLibrary.VirtualItem<T>))	The nodes in the path tree starting from the specified path IEnumerable<T> (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)

VirtualNotFoundException

## Parameters

Type	Name	Description
VirtualNodeType 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> ) ▼	<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> ( <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 ( <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> ) < VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) > (AkiraNetwork.VirtualStorageLibrary 1.html)	The paths in the path tree starting from the specified path IEnumerable<T> ( <a href="https://learn.microsoft.com/dotnet/api/system.collections.generic.ienumerable-1">https://learn.microsoft.com/dotnet/api/system.collections.generic.ienumerable-1</a> ) .

**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
VirtualID VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	path	The path of the symbolic link.

Type	Description
VirtualSymbolicLink 1.html)	The VirtualSymbolicLink (AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink.html) at the specified path.

Type	Condition
VirtualnodeName (AkiraNetwork.VirtualStorageLibrary	Thrown if the node cannot be found.

Type	Condition
VirtualPath	

(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.PatternMatch)	True if the item exists; otherwise, false.

(AkiraNetwork.VirtualStorageLibrary.VirtualCycleDetector)

(AkiraNetwork.VirtualStorageLibrary.VirtualDirectory)

## 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

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

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

▼

## NodeExists(VirtualPath, bool)

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)  
1.html)  
Checks if a node exists at the specified path.  
VirtualWildcardMatcher

Declaration  
(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

NotifyNodeDelegate

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)  
public bool NodeExists(VirtualPath path, bool followLinks = false)

PatternMatch

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

Parameters  
(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

Type	Name	Description
VirtualDirectory (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	path	The node's path.
bool ( <a href="https://learn.microsoft.com/dotnet/api/system.boolean">https://learn.microsoft.com/dotnet/api/system.boolean</a> ) (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	followLinks	Whether to follow symbolic links.

Returns  
(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

Type	Description
VirtualID bool ( <a href="https://learn.microsoft.com/dotnet/api/system.boolean">https://learn.microsoft.com/dotnet/api/system.boolean</a> ) (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	True if the node exists; otherwise, false.

VirtualItem  
(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

## RemoveLinkByLinkPath(VirtualPath)

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)  
Removes the link with the specified link path from the link dictionary.

VirtualNode  
(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

VirtualNodeContext  
public void RemoveLinkByLinkPath(VirtualPath linkPath)

VirtualNodeExtensions  
(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

Parameters  
(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

Type	Name	Description
VirtualNodeListConditions (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	linkPath	The link path.

Exceptions  
(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

Type	Condition
VirtualNodeType (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	Thrown if the link path is not an absolute path.

ArgumentException  
(<https://learn.microsoft.com/dotnet/api/system.argumentexception>)  
(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

VirtualNodeFilter  
(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

VirtualPath  
(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

## 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 (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html))	<i>targetPath</i>	The target path.
VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	<i>linkPath</i>	The path of the link.

PatternMatch

(AkiraNetwork.VirtualStorageLibrary.VirtualCycleDetector)

VirtualCycleDetector

## RemoveNode(VirtualPath, bool, bool, bool)

VirtualDirectory

Removes the node at the specified path.

(AkiraNetwork.VirtualStorageLibrary.VirtualException)

### Declaration

VirtualException

(AkiraNetwork.VirtualStorageLibrary.VirtualGroupCondition, VirtualPath)

```
virtual void RemoveNode(VirtualPath nodePath, bool recursive = false, bool followLinks = false, bool resolveLinks = true);
```

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

2.html)

VirtualID

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

Type	Name	Description
(AkiraNetwork.VirtualStorageLibrary.VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html))	<i>nodePath</i>	The path of the node to remove.
VirtualItem<T> (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)		
bool (https://learn.microsoft.com/dotnet/api/system.boolean)	<i>recursive</i>	Whether to remove recursively.
VirtualNode (https://learn.microsoft.com/dotnet/api/system.boolean)	<i>followLinks</i>	Whether to follow symbolic links.
VirtualNodeContext		
bool (https://learn.microsoft.com/dotnet/api/system.boolean)	<i>resolveLinks</i>	Whether to resolve link targets.
VirtualNodeExtensions		

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.

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.

VirtualNodeException

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

Type	Condition
VirtualNodeTypeFilter (AkiraNetwork.VirtualStorageLibrary.InvalidOperationException)	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)

## SetLinkTargetNodeType(HashSet<VirtualPath>, VirtualNodeType)

VirtualItem

Sets (AkiraNetwork.VirtualStorageLibrary.VirtualNode) symbolic links in the specified list of link paths.

**Declaration**  
(AkiraNetwork.VirtualStorageLibrary)

1.html)

```
public void SetLinkTargetNodeType(HashSet<VirtualPath> linkPathSet, VirtualNodeType nodeTyp  
e)
```

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeContext

**Parameters**  
(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeExtensions

Type	Name	Description
(AkiraNetwork.VirtualStorageLibrary)	linkPathSet	A set of link paths.
(https://learn.microsoft.com/dotnet/api/system.collections.generic.hashset-1)	nodeType	The target node type.

VirtualNodeName

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary)

**SetNode(VirtualPath, VirtualNode)**

(AkiraNetwork.VirtualStorageLibrary)

Sets VirtualPath the specified path.

(AkiraNetwork.VirtualStorageLibrary)

## Declaration

```
public void SetNode(VirtualPath destinationPath, VirtualNode node)
```

## Parameters

Type	Name	Description
VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.WildcardMatcher)	<i>destinationPath</i>	The path where the node is set
VirtualNode (AkiraNetwork.VirtualStorageLibrary.VirtualNode.NotifyNodeDelegate) (AkiraNetwork.VirtualStorageLibrary.VirtualNode.html) PatternMatch (AkiraNetwork.VirtualStorageLibrary.VirtualCycleDetector)	<i>node</i>	The VirtualNode (AkiraNetwork.VirtualStorageLibrary.VirtualNode.html) to set

**SetModeName(VirtualPath, VirtualNodeName, bool)**

VirtualException

**Virtual-Exception**  
Changes the name of a node.

VMware vSphere 5.5

# Virtual Declaration

(AkiraNetwork.VirtualStorageLibrary)

2.html)

```
public void SetNodeName(Virtual  
VirtualId
```

*virtualD  
rue).*

(AkiraNetwork)

Parrot AdressA

Type	VirtualItem<T> (AkiraNetwork.VirtualStorageLibrary)	Name	Description
VirtualPath	VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	nodePath	The path of the node to rename
VirtualNodeName	VirtualNodeContext (AkiraNetwork.VirtualStorageLibrary.VirtualNodeName.html) (AkiraNetwork.VirtualStorageLibrary)	newName	The new name
bool	VirtualNodeExtensions (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 sp

Checks if a symbolic link  
(AkiraNetwork.Vi)

**clVirtualNodeType**

(AkiraNetwork Virtual)

#### **VirtualNodeTypeFilter**

```
public bool symbolicLinkExists(
```

(AkıraNetwo

Virtual

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> ) (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 (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> ) 2.html)	followLinks	Whether to follow symbolic links.

VirtualID

Returns  
(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

Type	Description
VirtualItem (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	
VirtualDirectory VirtualItem<T> (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	The found VirtualDirectory (AkiraNetwork.VirtualStorageLibrary.VirtualDirectory.html) , or null if not found.
VirtualNode (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	
VirtualNodeContext (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 (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)		
VirtualNodeTypeFilter (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> ) VirtualPath.html	followLinks	Whether to follow symbolic links.

(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

## Returns

Type	Description
VirtualItem (AkiraNetwork.VirtualStorageLibrary.VirtualItem-1.html) <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> bool ( <a href="https://learn.microsoft.com/dotnet/api/system.boolean">https://learn.microsoft.com/dotnet/api/system.boolean</a> ) (AkiraNetwork.VirtualStorageLibrary-2.html)	followLinks	Whether to follow symbolic links.

### Returns

VirtualID

Type	Description
VirtualItem VirtualNode (AkiraNetwork.VirtualStorageLibrary.VirtualNode.html) VirtualItem<T> VirtualNode.html (AkiraNetwork.VirtualStorageLibrary-1.html)	The found VirtualNode (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

Type	Name	Description
VirtualNodeType		

VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)

path

The path of the symbolic link.

VirtualNodeTypeFilter

### Returns

VirtualPath

(AkiraNetwork.VirtualStorageLibrary-1.html)

Type	Description
VirtualSymbolicLink (AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink.html)	The found VirtualSymbolicLink (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)

## Parameters

VirtualException

Type	Name	Description
AkiraNetwork.VirtualStorageLibrary.VirtualGroupCondition<T, TKey> VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	path	The path of the symbolic link.

2.html)

## Returns

VirtualID

Type	Description
AkiraNetwork.VirtualStorageLibrary.VirtualItem VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html) (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	The target path, or null if the node cannot be found.

VirtualItem<T>  
(AkiraNetwork.VirtualStorageLibrary)

1.html)

## UpdateAllTargetNodeTypesInDictionary()

(AkiraNetwork.VirtualStorageLibrary)

Updates all target node types in the link dictionary.

VirtualNodeContext

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeExtensions

(AkiraNetwork.VirtualStorageLibrary)

```
public void UpdateAllTargetNodeTypesInDictionary()
```

VirtualNodeListConditions

(AkiraNetwork.VirtualStorageLibrary)

VirtualnodeName

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeNotFoundException

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeTypeFilter

```
public void UpdateAllTargetNodeTypesInDictionary(VirtualPath directoryPath, VirtualDirectory newDirectory)
```

VirtualPath

(AkiraNetwork.VirtualStorageLibrary)

## UpdateDirectory(VirtualPath, VirtualDirectory)

(AkiraNetwork.VirtualStorageLibrary)

VirtualDirectory

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary)

VirtualNodeTypeFilter

```
public void UpdateDirectory(VirtualPath directoryPath, VirtualDirectory newDirectory)
```

VirtualPath

(AkiraNetwork.VirtualStorageLibrary)

## Parameters

Type	Name	Description
VirtualPath (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	<i>directoryPath</i>	The path of the directory to update
VirtualDirectory (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)

## 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)	<b>VirtualPath</b> (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	<i>linkPath</i>	The path of the link.
(AkiraNetwork.VirtualStorageLibrary)	<b>VirtualNodeName</b>		
(AkiraNetwork.VirtualStorageLibrary)	<b>VirtualPath</b> (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 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)		filter	The filter to apply for node types
(AkiraNetwork.VirtualStorageLibrary.VirtualNodeTypeFilter.NotifyNodeDelegate.html)			
PatternMatch			
bool	(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	recursive	A flag indicating whether to traverse recursively
bool	(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	followLinks	A flag indicating whether to follow symbolic links
bool	(AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)	resolveLinks	A flag indicating whether to resolve symbolic links

Return(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) html&gt; VirtualNode &gt; (AkiraNetwork.VirtualStorageLibrary.VirtualNodeContext)</pre>	<p>An enumerable of <code>VirtualNodeContext</code> (see <a href="#">VirtualNodeContext.html</a>) containing information about the traversed nodes</p>

Revised (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. VirtualNodeGetConditions

**Virtual Node Conditions**  
resolves links when nodes other than the terminal node in the virtual path are symbols.

# Extension Methods

(AKL) a NETWORK.VIRTUALSTORAGELIBRARY

## Virtually TexaFondatyper GenerateLinkTableI

(Akira Akitan de kōtōku wa isto o sōtōri ga nai hōdai)

### **Virtualized Nested Type Filter**

View (AttnFlatworld) VirtualStorageUtil

# Virtuální křesťanské normativní funkce bácevského folklóru

## (AkiraVirtualDiskVirtualStorageLibrary.Util

(AkiraNetwork.VirtualStorageLibrary)

```
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  
(AkiraNetwork.VirtualStorageLibrary.VirtualNode.html#AkiraNetwork\_VirtualStorageLibrary\_VirtualNode\_VID)  
VirtualNode.IsReferencedInStorage  
(AkiraNetwork.VirtualStorageLibrary)  
(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) {
        // ...
    }
}
```

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

## 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; }
}
```

VirtualNodeNotNotFoundException

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeNotFoundException)

Property Value

VirtualNodeType

(AkiraNetwork.VirtualStorageLibrary.VirtualNodeType)

VirtualNodeTypeFilter

VirtualNodeType (AkiraNetwork.VirtualStorageLibrary.VirtualNodeType.html)

VirtualPath

Overrides

(AkiraNetwork.VirtualStorageLibrary.VirtualNode)

Type	Description
VirtualNodeType	

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

## TargetNodeType

Gets or sets the type of the target node of the symbolic link.

(AkiraNetwork.VirtualStorageLibrary  
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; }

VirtualItem

(AkiraNetwork.VirtualStorageLibrary  
Property Value)

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  
Parameters)

VirtualNodeTypeFilter

(AkiraNetwork.VirtualStorageLibrary  
VirtualPath)

(AkiraNetwork.VirtualStorageLibrary  
VirtualPathFilter)

## Description



## Exceptions

Type	Condition
ArgumentException ( <a href="https://learn.microsoft.com/dotnet/api/system.argumentexception">https://learn.microsoft.com/dotnet/api/system.argumentexception</a> )	Thrown if the specified node is not a VirtualSymbolicLink.

## Operators

### **implicit operator VirtualSymbolicLink(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

virtualGroupCondition<T, TKey>

([AkiraNetwork.VirtualStorageLibrary](#))

VirtualNodeName

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

VirtualID

([AkiraNetwork.VirtualStorageLibrary](#))

#### Returns

VirtualItem

Type	Description
AkiraNetwork.VirtualStorageLibrary	

VirtualItem<T>

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

([AkiraNetwork.VirtualStorageLibrary](#))

1.html)

VirtualNode

([AkiraNetwork.VirtualStorageLibrary](#))

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.

([AkiraNetwork.VirtualStorageLibrary](#))

VirtualNodeListConditions

([AkiraNetwork.VirtualStorageLibrary](#))

VirtualNodeName

public static implicit operator VirtualSymbolicLink(VirtualPath? targetPath)

([AkiraNetwork.VirtualStorageLibrary](#))

VirtualNodeNotFoundException

([AkiraNetwork.VirtualStorageLibrary](#))

VirtualNodeType

([AkiraNetwork.VirtualStorageLibrary](#))

VirtualNodeTypeFilter

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

([AkiraNetwork.VirtualStorageLibrary](#))

VirtualPath

([AkiraNetwork.VirtualStorageLibrary](#))

Type	Name	Description
AkiraNetwork.VirtualStorageLibrary		

VirtualPath	targetPath	The target path to convert.
-------------	------------	-----------------------------

VirtualNodeFilter		
-------------------	--	--

VirtualNodeName		
-----------------	--	--

VirtualSymbolicLink		
---------------------	--	--

VirtualSymbolicLink		
---------------------	--	--

VirtualSymbolicLink		
---------------------	--	--

VirtualSymbolicLink		
---------------------	--	--

VirtualSymbolicLink		
---------------------	--	--

VirtualSymbolicLink		
---------------------	--	--

VirtualSymbolicLink		
---------------------	--	--

VirtualSymbolicLink		
---------------------	--	--

VirtualSymbolicLink		
---------------------	--	--

VirtualSymbolicLink		
---------------------	--	--

VirtualSymbolicLink		
---------------------	--	--

VirtualSymbolicLink		
---------------------	--	--

VirtualSymbolicLink		
---------------------	--	--

VirtualSymbolicLink		
---------------------	--	--

VirtualSymbolicLink		
---------------------	--	--

VirtualSymbolicLink		
---------------------	--	--

VirtualSymbolicLink		
---------------------	--	--

VirtualSymbolicLink		
---------------------	--	--

VirtualSymbolicLink		
---------------------	--	--

VirtualSymbolicLink		
---------------------	--	--

VirtualSymbolicLink		
---------------------	--	--

VirtualSymbolicLink		
---------------------	--	--

VirtualSymbolicLink		
---------------------	--	--

VirtualSymbolicLink		
---------------------	--	--

VirtualSymbolicLink		
---------------------	--	--

VirtualSymbolicLink		
---------------------	--	--

VirtualSymbolicLink		
---------------------	--	--

VirtualSymbolicLink		
---------------------	--	--

VirtualSymbolicLink		
---------------------	--	--

VirtualSymbolicLink		
---------------------	--	--

VirtualSymbolicLink		
---------------------	--	--

VirtualSymbolicLink		
---------------------	--	--

VirtualSymbolicLink		
---------------------	--	--

VirtualSymbolicLink		
---------------------	--	--

VirtualSymbolicLink		
---------------------	--	--

VirtualSymbolicLink		
---------------------	--	--

VirtualSymbolicLink		
---------------------	--	--

VirtualSymbolicLink		
---------------------	--	--

VirtualSymbolicLink		
---------------------	--	--

VirtualSymbolicLink		
---------------------	--	--

VirtualSymbolicLink		
---------------------	--	--

VirtualSymbolicLink		
---------------------	--	--

VirtualSymbolicLink		
---------------------	--	--

VirtualSymbolicLink		
---------------------	--	--

VirtualSymbolicLink		
---------------------	--	--

VirtualSymbolicLink		
---------------------	--	--

VirtualSymbolicLink		
---------------------	--	--

VirtualSymbolicLink		
---------------------	--	--

VirtualSymbolicLink		
---------------------	--	--

VirtualSymbolicLink		
---------------------	--	--

VirtualSymbolicLink		
---------------------	--	--

VirtualSymbolicLink		
---------------------	--	--

VirtualSymbolicLink		
---------------------	--	--

VirtualSymbolicLink		
---------------------	--	--

VirtualSymbolicLink		
---------------------	--	--

VirtualSymbolicLink		
---------------------	--	--

VirtualSymbolicLink		
---------------------	--	--

VirtualSymbolicLink		
---------------------	--	--

VirtualSymbolicLink		
---------------------	--	--

VirtualSymbolicLink		
---------------------	--	--

VirtualSymbolicLink		
---------------------	--	--

VirtualSymbolicLink		
---------------------	--	--

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 (AkiraNetwork.VirtualStorageLibrary)		

string ( <a href="https://learn.microsoft.com/dotnet/api/system.string">https://learn.microsoft.com/dotnet/api/system.string</a> )	nodeName	The node name to convert.
--	----------	---------------------------

### Returns

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

VirtualSymbolicLink (AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink.html)	
---	--

#### VirtualID

Type	Description
VirtualItem<T> (AkiraNetwork.VirtualStorageLibrary)	

VirtualSymbolicLink ((VirtualNodeName nodeName, VirtualPath? targetPath)) (AkiraNetwork.VirtualStorageLibrary)	
---	--

#### VirtualItem<T>

Performs an implicit conversion from a tuple to VirtualSymbolicLink  
 (AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink.html).  
 (AkiraNetwork.VirtualStorageLibrary.VirtualSymbolicLink.html).

#### VirtualPath

Type	Description
VirtualPath (AkiraNetwork.VirtualStorageLibrary)	

Type	Description
VirtualPath (AkiraNetwork.VirtualStorageLibrary)	

Type	Description
VirtualPath (AkiraNetwork.VirtualStorageLibrary)	

Type	Description
VirtualPath (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 (AkiraNetwork.VirtualStorageLibrary.VirtualPath.html)		<i>basePath</i>	The base path from which to start.
bool (https://learn.microsoft.com/dotnet/api/system.boolean) ▼		<i>recursive</i>	Whether to list subdirectories recursively.
bool (https://learn.microsoft.com/dotnet/api/system.boolean)		<i>followLinks</i>	Whether to follow symbolic links.

## + AkiraNetwork.VirtualStorage

### Library

Type	Description
-string (https://learn.microsoft.com/dotnet/api/system.string)	The generated debug text of the tree structure.

### Library.Utilities

#### Type AkiraNetwork.VirtualStorage

Name	Description
<i>T</i> (AkiraNetwork.VirtualStorageLibrary)	The type of data in the storage.

## + AkiraNetwork.VirtualStorage

### Library.WildcardMatchers

The generated text represents the tree structure starting from the specified base path. Each node is represented by the result of the `ToString` method overridden in the specific derived class of the node.

When `recursive` is true, subdirectories are listed recursively. If false, subdirectories are not included in the output.

When `followLinks` is true, symbolic links are replaced with their target paths in the output. If false, the symbolic links themselves are displayed.

The output format includes indents and special characters to denote directory hierarchies and symbolic links.  
Example:

```

/
├dir1/
│├subdir1/
││└item3
│└item2
└link-to-dir -> /dir1
└item1
└link-to-item -> /item1

```

# Namespace AkiraNetwork.VirtualStorage Library.WildcardMatchers

▼ Filter by title

## Classes

+ **AkiraNetwork.VirtualStorage**

**DefaultWildcardMatcher**

        (AkiraNetwork.VirtualStorageLibrary.WildcardMatchers.DefaultWildcardMatcher.html)

- **AkiraNetwork.VirtualStorage**

**LibraryUtilities**

        Implements default wildcard matching using regular expression patterns. Provides a straightforward mapping of

        regular expression symbols.

        (AkiraNetwork.VirtualStorageLibrary.WildcardMatchers.DefaultWildcardMatcher.html)

- **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.WildcardMatchers.PowerShellWildcardMatcher.html)

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

## (AkiraNetwork.VirtualStorageLibrary)

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

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 ( <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> ) <string ( <a href="https://learn.microsoft.com/dotnet/api/system.string">https://learn.microsoft.com/dotnet/api/system.string</a> ), 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 ( <a href="https://learn.microsoft.com/dotnet/api/system.collections.generic.ienumerable-1">https://learn.microsoft.com/dotnet/api/system.collections.generic.ienumerable-1</a> ) <string ( <a href="https://learn.microsoft.com/dotnet/api/system.string">https://learn.microsoft.com/dotnet/api/system.string</a> )>	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 ( <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 ( <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 ( <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))

### (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](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 ( <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> ) <string ( <a href="https://learn.microsoft.com/dotnet/api/system.string">https://learn.microsoft.com/dotnet/api/system.string</a> ), 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 ( <a href="https://learn.microsoft.com/dotnet/api/system.collections.generic.ienumerable-1">https://learn.microsoft.com/dotnet/api/system.collections.generic.ienumerable-1</a> ) <string ( <a href="https://learn.microsoft.com/dotnet/api/system.string">https://learn.microsoft.com/dotnet/api/system.string</a> )>	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 ( <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 ( <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 ( <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)