

OpcServerBase Members

Namespace: Opc.UaFx.Server

Assemblies: Opc.UaFx.Advanced.dll, Opc.UaFx.Advanced.dll

The [OpcServerBase](#) type exposes the following members.

Constructors

OpcServerBase(OpcServerType)

C#

```
protected OpcServerBase(OpcServerType type)
```

Parameters

type [OpcServerType](#)

OpcServerBase(OpcServerType, String)

C#

```
protected OpcServerBase(OpcServerType type, string address)
```

Parameters

type [OpcServerType](#)

address [String](#)

Exceptions

[ArgumentException](#)

The **address** format / scheme can not be used as a server address.

[ArgumentNullException](#)

[UriFormatException](#)

OpcServerBase(OpcServerType, Uri)

C#

```
protected OpcServerBase(OpcServerType type, Uri address)
```

Parameters

type [OpcServerType](#)

address [Uri](#)

Exceptions

[ArgumentException](#)

The [address](#) format / scheme can not be used as a server address.

[ArgumentNullException](#)

Events

Started

C#

```
public event EventHandler Started
```

Starting

C#

```
public event EventHandler Starting
```

StateChanged

Occurs when ever the [State](#) property has been changed to a different value.

C#

```
public event EventHandler<OpcServerStateChangedEventArgs> StateChanged
```

Stopped

C#

```
public event EventHandler Stopped
```

Stopping

C#

```
public event EventHandler Stopping
```

Properties

Address

Gets or sets the (first) base address of the server.

C#

```
public Uri Address { get; set; }
```

Property Value

[Uri](#)

The first [Uri](#) used by the server to listen and response to OPC UA requests or a null reference (Nothing in Visual Basic) in case there is no base address defined.

Exceptions

[ArgumentException](#)

The address format / scheme can not be used as a server address.

[ArgumentNullException](#)

The value is a null reference (Nothing in Visual Basic).

Addresses

Gets a read-only collection of [Uri](#) instances representing the base addresses used by the server.

C#

```
public ReadOnlyCollection<Uri> Addresses { get; }
```

Property Value

ReadOnlyCollection<Uri>

A [ReadOnlyCollection](#) of [Uri](#) instances used by the server to listen and response to OPC UA requests.

Remarks

To manage the different base addresses used use the [RegisterAddress\(String\)/RegisterAddress\(Uri\)](#) method to add an additional base address or use the [UnregisterAddress\(String\)/ UnregisterAddress\(Uri\)](#) method to remove a base address.

Binding

Gets the [OpcServerBinding](#) instance used to maintain the different binding options used by the server application.

C#

```
public OpcServerBinding Binding { get; }
```

Property Value

[OpcServerBinding](#)

An instance of the [OpcServerBinding](#) class used by the [OpcServer](#) to setup and maintain the different binding options.

Configuration

Gets or sets an instance of the [OpcApplicationConfiguration](#) class which is used as a low-level representation of the code/file based server application configuration.

C#

```
[CLSCompliant(false)]  
public override OpcApplicationConfiguration Configuration { get; set; }
```

Property Value

[OpcApplicationConfiguration](#)

An instance of the [OpcApplicationConfiguration](#) class configured with the server specific setup to use. An assignment of a null reference (Nothing in Visual Basic) will lead to the creation of a new instance of the [OpcApplicationConfiguration](#) class with all defaults typically used by server applications.

Remarks

A file based [OpcApplicationConfiguration](#) needs to be loaded manually either using a specific configuration section in its App.config (see [LoadServerConfig\(String\)](#)) or using a custom file using [LoadServerConfigFile\(String\)](#). A low-level code based server configuration can be directly performed using

an instance of the [OpcApplicationConfiguration](#) like the instance provided by this property. In case there is a different instance applied to this property the property will ensure that the [Address](#) property will represent the first base address configured in the base addresses of the passed configuration instance.

Name

C#

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

Property Value

[String](#)

Namespaces

Gets a read-only collection of namespaces used by the server.

C#

```
public override OpcReadOnlyNamespaceCollection Namespaces { get; }
```

Property Value

[OpcReadOnlyNamespaceCollection](#)

An instance of the [OpcReadOnlyNamespaceCollection](#) class with [OpcNamespace](#) items for each 'NamespaceUri' used by the server to organize the nodes in its address space.

Remarks

The items in the [OpcReadOnlyNamespaceCollection](#) provided are created during the startup of the server.

State

Gets a value that indicates the current state of the server.

C#

```
public OpcServerState State { get; }
```

Property Value

[OpcServerState](#)

One of the members defined by the [OpcServerState](#) enumeration. Which specifies the current state of the server.

Methods

CreateConfiguration()

C#

```
[CLSCompliant(false)]
protected virtual OpcApplicationConfiguration CreateConfiguration()
```

Returns

[OpcApplicationConfiguration](#)

CreateSecurity()

Creates a new instance of the [OpcServerSecurity](#) class.

C#

```
protected override sealed OpcServerSecurity CreateSecurity()
```

Returns

[OpcServerSecurity](#)

A new instance of the [OpcServerSecurity](#) class.

CreateTransport()

Creates a new instance of the [OpcTransport](#) class.

C#

```
protected override sealed OpcTransport CreateTransport()
```

Returns

[OpcTransport](#)

A new instance of the [OpcTransport](#) class.

DenyIfNotStarted()

Verifies whether the server is in started state.

C#

```
protected void DenyIfNotStarted()
```

Exceptions

InvalidOperationException

The [State](#) is not equals [Started](#).

Dispose(Boolean)

Releases the unmanaged resources used by the [OpcServerBase](#) and optionally releases the managed resources.

C#

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

Parameters

disposing Boolean

The value true to release both managed and unmanaged resources; otherwise the value false to release only unmanaged resources.

GetInfo()

C#

```
public OpcServerInfo GetInfo()
```

Returns

[OpcServerInfo](#)

GetInfoCore()

C#

```
protected virtual OpcServerInfo GetInfoCore()
```

Returns

[OpcServerInfo](#)

OnStarted(EventArgs)

C#

```
protected virtual void OnStarted(EventArgs e)
```

Parameters

e EventArgs

OnStarting(EventArgs)

C#

```
protected virtual void OnStarting(EventArgs e)
```

Parameters

e EventArgs

OnStateChanged(OpcServerStateChangedEventArgs)

C#

```
protected virtual void OnStateChanged(OpcServerStateChangedEventArgs e)
```

Parameters

e OpcServerStateChangedEventArgs

OnStopped(EventArgs)

C#

```
protected virtual void OnStopped(EventArgs e)
```

Parameters

e EventArgs

OnStopping(EventArgs)

C#

```
protected virtual void OnStopping(EventArgs e)
```

Parameters

e EventArgs

RegisterAddress(String)

Registers the **address** specified as one of the base addresses used by the server.

C#

```
public void RegisterAddress(string address)
```

Parameters

address [String](#)

The uniform resource identifier which defines the communication schema, the host, the port and optionally the path to listen to.

Exceptions

[ArgumentNullException](#)

The **address** is a null reference (Nothing in Visual Basic).

[ArgumentException](#)

The **address** is an empty string or the address format / scheme can not be used as a server address.

[FormatException](#)

The **address** is not a valid URI format.

Remarks

In case there the **address** is already in use as a base address, it will not be added to the list of base addresses a second time. In case there the [Address](#) properties is a null reference (Nothing in Visual Basic), the **address** will be used.

RegisterAddress(Uri)

Registers the **address** specified as one of the base addresses used by the server.

C#

```
public void RegisterAddress(Uri address)
```

Parameters

address [Uri](#)

The [Uri](#) which defines the communication schema, the host, the port and optionally the path to listen to.

Exceptions

ArgumentException

The **address** format / scheme can not be used as a server address.

ArgumentNullException

The **address** is a null reference (Nothing in Visual Basic).

Remarks

In case there the **address** is already in use as a base address, it will not be added to the list of base addresses a second time. In case there the **Address** properties is a null reference (Nothing in Visual Basic), the **address** will be used.

Resume()

C#

```
public void Resume()
```

Exceptions

[InvalidOperationException](#)

[OpcException](#)

[ObjectDisposedException](#)

[NotSupportedException](#)

ResumeCore()

C#

```
protected virtual void ResumeCore()
```

Setup(OpcApplicationConfiguration)

C#

```
[CLSCompliant(false)]  
protected override void Setup(OpcApplicationConfiguration configuration)
```

Parameters

configuration [OpcApplicationConfiguration](#)

Start()

C#

```
public void Start()
```

Exceptions

[InvalidOperationException](#)

The [Address](#) is not specified.

[OpcException](#)

The configuration is invalid.

[ObjectDisposedException](#)

StartCore(OpcApplicationConfiguration)

C#

```
[CLSCompliant(false)]  
protected virtual void StartCore(OpcApplicationConfiguration configuration)
```

Parameters

configuration [OpcApplicationConfiguration](#)

Stop()

C#

```
public void Stop()
```

Exceptions

[InvalidOperationException](#)

[OpcException](#)

[ObjectDisposedException](#)

StopCore()

C#

```
protected abstract void StopCore()
```

Suspend()

C#

```
public void Suspend()
```

Exceptions

[InvalidOperationException](#)

[OpcException](#)

[ObjectDisposedException](#)

[NotSupportedException](#)

SuspendCore()

C#

```
protected virtual void SuspendCore()
```

UnregisterAddress(String)

Unregisters the **address** specified from the list of base addresses used by the server.

C#

```
public void UnregisterAddress(string address)
```

Parameters

address String

The uniform resource identifier which defines the communication schema, the host, the port and optionally the path to remove.

Exceptions

ArgumentNullException

The **address** is a null reference (Nothing in Visual Basic).

ArgumentException

The **address** is an empty string.

FormatException

The **address** is not a valid URI format.

Remarks

In case there **address** is equals to **Address** the **Address** property will be set to a null reference (Nothing in Visual Basic) if there is no other address registered; otherwise the first entry in **Addresses** will be used for **Address**.

UnregisterAddress(Uri)

Unregisters the **address** specified from the list of base addresses used by the server.

C#

```
public void UnregisterAddress(Uri address)
```

Parameters

address Uri

The **Uri** which defines the communication schema, the host, the port and optionally the path to remove.

Exceptions

ArgumentNullException

The **address** is a null reference (Nothing in Visual Basic).

Remarks

In case there **address** is equals to **Address** the **Address** property will be set to a null reference (Nothing in Visual Basic) if there is no other address registered; otherwise the first entry in **Addresses** will be used for

Address.

Table of Contents

Constructors	1
OpcServerBase(OpcServerType)	1
OpcServerBase(OpcServerType, String)	1
OpcServerBase(OpcServerType, Uri)	2
Events	2
Started	2
Starting	2
StateChanged	2
Stopped	3
Stopping	3
Properties	3
Address	3
Addresses	3
Binding	4
Configuration	4
Name	5
Namespaces	5
State	5
Methods	6
CreateConfiguration()	6
CreateSecurity()	6
CreateTransport()	6
DenyIfNotStarted()	6
Dispose(Boolean)	7
GetInfo()	7
GetInfoCore()	7
OnStarted(EventArgs)	7
OnStarting(EventArgs)	8
OnStateChanged(OpcServerStateChangedEventArgs)	8
OnStopped(EventArgs)	8
OnStopping(EventArgs)	8
RegisterAddress(String)	9
RegisterAddress(Uri)	9
Resume()	10
ResumeCore()	10
Setup(OpcApplicationConfiguration)	10
Start()	11
StartCore(OpcApplicationConfiguration)	11
Stop()	11
StopCore()	12
Suspend()	12
SuspendCore()	12
UnregisterAddress(String)	12
UnregisterAddress(Uri)	13

