

# IOpcApplicationInstance Members

**Namespace:** Opc.UaFx

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

The [IOpcApplicationInstance](#) interface defines the following members.

## Events

### CertificateRequested

Occurs when the certificate of this application instance could not automatically determined using the information of this [IOpcApplicationInstance](#). This event is when used to query the certificate to use for the application instance. Additionally use this event to determine the case that an appropriate instance certificate is missing and in case there [AutoCreateCertificate](#) of [CertificateStores](#) is equals true a new certificate is being created automatically after the event has been processed (and this may take some time).

**C#**

```
event OpcCertificateRequestedEventHandler CertificateRequested
```

### CertificateValidationFailed

Occurs when the validation of a opponent application instance certificate does not fulfill the requirements for valid and trusted certificates.

**C#**

```
event OpcCertificateValidationFailedEventHandler CertificateValidationFailed
```

### Remarks

This event can be used to complement the default validation of opponent application instance certificates. Using the event arguments it is possible to accept the certificate using different/custom circumstances.

## Properties

### ApplicationName

Gets or sets the name of the application instance.

**C#**

```
string ApplicationName { get; set; }
```

## Property Value

### String

The name of the application. The default value is "OPC Application" in case there is no entry assembly available; otherwise the value of the [AssemblyTitleAttribute](#) of the entry assembly.

## Remarks

The name does not have to be unique. The value of this property is also used as the [Cryptography.X509Certificates.X509Certificate2.SubjectName](#) of the application certificate whenever an application certificate is to be created/looked-up (from the application certificates store).

# ApplicationType

Gets the type of application the instance represents.

## C#

```
OpcApplicationType ApplicationType { get; }
```

## Property Value

### OpcApplicationType

One of the members defined by the [OpcApplicationType](#) enumeration. Each member identifies a specific type of OPC UA application and its usage within the OPC UA environment.

# ApplicationUri

Gets or sets the URI of the application instance.

## C#

```
Uri ApplicationUri { get; set; }
```

## Property Value

### Uri

The [Uri](#) or a null reference (Nothing in Visual Basic) which uniquely identifies the application instance.

## Remarks

In case there the value of this property is a null reference (Nothing in Visual Basic) the URI of the application is determined by the subject alternate name of the application certificate used. In case there an application certificate is to be created, this value is used to define the certificates subject alternate name.

# Certificate

Gets or sets the application certificate representing the applications identity.

## C#

```
X509Certificate2 Certificate { get; set; }
```

## Property Value

[Cryptography.X509Certificates.X509Certificate2](#)

An instance of the [Cryptography.X509Certificates.X509Certificate2](#) class used to identify the application instance.

# CertificateStores

Gets the [OpcCertificateStores](#) instance used to maintain the different certificate stores used by the application instance.

## C#

```
OpcCertificateStores CertificateStores { get; }
```

## Property Value

[OpcCertificateStores](#)

An instance of the [OpcCertificateStores](#) class used by the instance to setup and maintain the certificate stores and their certificates used by the application instance.

# Configuration

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

## C#

```
OpcApplicationConfiguration Configuration { get; set; }
```

## Property Value

[OpcApplicationConfiguration](#)

An instance of the [OpcApplicationConfiguration](#) class configured with the instance 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 the type of application instance.

# Namespaces

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

## C#

```
OpcReadOnlyNamespaceCollection Namespaces { get; }
```

## Property Value

[OpcReadOnlyNamespaceCollection](#)

An instance of the [OpcReadOnlyNamespaceCollection](#) class with [OpcNamespace](#) items for each 'NamespaceUri' known/used.

# ProductUri

Gets or sets the URI of the product.

## C#

```
Uri ProductUri { get; set; }
```

## Property Value

[Uri](#)

The [Uri](#) or a null reference (Nothing in Visual Basic) which uniquely identifies the product.

## Remarks

In case there the value of this property is a null reference (Nothing in Visual Basic) the URI of the product is determined by the [AssemblyCompanyAttribute](#) and [AssemblyProductAttribute](#) used.

# Security

Gets the [OpcSecurity](#) instance used to maintain the different security options used by the application instance.

## C#

```
OpcSecurity Security { get; }
```

## Property Value

[OpcSecurity](#)

An instance of the [OpcSecurity](#) class used by the [IOpcApplicationInstance](#) to setup and maintain the different security options.

# SystemContext

Gets the [OpcContext](#) which is used to store the context sensitive data and configuration used by the current application instance.

## C#

```
OpcContext SystemContext { get; }
```

## Property Value

### [OpcContext](#)

An instance of the [OpcContext](#) class which might be at least [Empty](#).

# Transport

Gets the [OpcTransport](#) instance used to maintain the different transport options used by the application instance.

## C#

```
OpcTransport Transport { get; }
```

## Property Value

### [OpcTransport](#)

An instance of the [OpcTransport](#) class used by the [IOpcApplicationInstance](#) to setup and maintain the different transport options.

# Methods

## QueryCertificate(OpcCertificateValidationResult)

Requests the certificate to use for this application instance in case there the certificate could not automatically determined using the information of this application instance or there are any issues with the [Certificate](#).

## C#

```
X509Certificate2 QueryCertificate(OpcCertificateValidationResult reason)
```

## Parameters

**reason** [OpcCertificateValidationResult](#)

One of the members defined by the [OpcCertificateValidationResult](#) enumeration indicating the reason upon that the certificate could not automatically determined or indicating which issue with the [Certificate](#) exists.

## Returns

### [Cryptography.X509Certificates.X509Certificate2](#)

An instance of the [Cryptography.X509Certificates.X509Certificate2](#) class indicating the application instance certificate to use or a null reference (Nothing in Visual Basic) in case there the framework shall create a new certificate by its own using the application information of this instance.

# Table of Contents

<b>Events</b> .....	1
CertificateRequested .....	1
CertificateValidationFailed .....	1
<b>Properties</b> .....	1
ApplicationName .....	1
ApplicationType .....	2
ApplicationUri .....	2
Certificate .....	3
CertificateStores .....	3
Configuration .....	3
Namespaces .....	4
ProductUri .....	4
Security .....	4
SystemContext .....	5
Transport .....	5
<b>Methods</b> .....	5
QueryCertificate(OpcCertificateValidationResult) .....	5

