

OpcMethodDelegateCommand Members

Namespace: Opc.UaFx

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

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

Constructors

OpcMethodDelegateCommand(OpcMethodExecuteCallback)

Initializes a new instance of the [OpcMethodDelegateCommand](#) class using the [executeCallback](#) specified.

C#

```
public OpcMethodDelegateCommand(OpcMethodExecuteCallback executeCallback)
```

Parameters

[executeCallback](#) [OpcMethodExecuteCallback](#)

The [OpcMethodExecuteCallback](#) which is invoked whenever the method of this command is called.

Exceptions

[ArgumentNullException](#)

The [executeCallback](#) is a null reference (Nothing in Visual Basic).

OpcMethodDelegateCommand(OpcMethodExecuteCallback, OpcMethodCanExecuteCallback)

Initializes a new instance of the [OpcMethodDelegateCommand](#) class using the [executeCallback](#) and [canExecuteCallback](#) specified.

C#

```
public OpcMethodDelegateCommand(OpcMethodExecuteCallback executeCallback,  
OpcMethodCanExecuteCallback canExecuteCallback)
```

Parameters

[executeCallback](#) [OpcMethodExecuteCallback](#)

The [OpcMethodExecuteCallback](#) which is invoked whenever the [canExecuteCallback](#) evaluates to the value true as soon as the method of this command is called.

`canExecuteCallback` `OpcMethodCanExecuteCallback`

The `OpcMethodCanExecuteCallback` which is used to evaluate whether the method of this command can be called in general and by the current user.

Exceptions

`ArgumentNullException`

The `executeCallback` or `canExecuteCallback` is a null reference (Nothing in Visual Basic).

OpcMethodDelegateCommand(OpcMethodExecuteCallback, OpcMethodCanExecuteCallback, OpcMethodCanExecuteCallback)

Initializes a new instance of the `OpcMethodDelegateCommand` class using the `executeCallback`, `canExecuteCallback` and `canUserExecuteCallback` specified.

C#

```
public OpcMethodDelegateCommand(OpcMethodExecuteCallback executeCallback,
OpcMethodCanExecuteCallback canExecuteCallback, OpcMethodCanExecuteCallback
canUserExecuteCallback)
```

Parameters

`executeCallback` `OpcMethodExecuteCallback`

The `OpcMethodExecuteCallback` which is invoked whenever the `canExecuteCallback` and `canUserExecuteCallback` evaluate to the value true as soon as the method of this command is called.

`canExecuteCallback` `OpcMethodCanExecuteCallback`

The `OpcMethodCanExecuteCallback` which is used to evaluate whether the method of this command can be called in general.

`canUserExecuteCallback` `OpcMethodCanExecuteCallback`

The `OpcMethodCanExecuteCallback` which is used to evaluate whether the method of this command can be called by the current user.

Exceptions

`ArgumentNullException`

The `executeCallback`, `canExecuteCallback` or `canUserExecuteCallback` is a null reference (Nothing in Visual Basic).

Methods

CanExecute(OpcContext)

Determines the value of the [Executable](#) attribute of the method ([CanExecute](#)) and therefore decides whether the [Execute\(OpcMethodContext, IList, IList\)](#) method can be called in general to execute the method implementation.

C#

```
public bool CanExecute(OpcContext context)
```

Parameters

context [OpcContext](#)

The [OpcReadAttributeValueContext](#) in case there the [Executable](#) attribute is read or the [OpcMethodContext](#) in case there the method call is to be processed.

Returns

[Boolean](#)

The value true if the method can be called; otherwise the value false.

CanUserExecute(OpcContext)

Determines the value of the [UserExecutable](#) attribute of the method ([CanUserExecute](#)) and therefore decides whether the [Execute\(OpcMethodContext, IList, IList\)](#) method can be called by the current user to execute the method implementation.

C#

```
public bool CanUserExecute(OpcContext context)
```

Parameters

context [OpcContext](#)

The [OpcReadAttributeValueContext](#) in case there the [UserExecutable](#) attribute is read or the [OpcMethodContext](#) in case there the method call is to be processed.

Returns

[Boolean](#)

The value true if the method can be called; otherwise the value false.

Execute(OpcMethodContext, IList<Object>, IList<Object>)

Performs the tasks necessary to implement the logic associated with a specific method using the **context** and **inputArguments** specified.

C#

```
public void Execute(OpcMethodContext context, IList<object> inputArguments, IList<object> outputArguments)
```

Parameters

context [OpcMethodContext](#)

The [OpcMethodContext](#) which provides the contextual information required to execute the method.

inputArguments [IList<Object>](#)

The generic list of [Object](#) values to use / respect during the execution of the method.

outputArguments [IList<Object>](#)

The generic list of [Object](#) values to use for out, ref (in Visual Basic ByRef) and returned values.

Remarks

Any OPC UA related result information is to be set using the [Result](#) property of the **context**.

This method is (to be) only called after the call to [CanExecute\(OpcContext\)](#) and [CanUserExecute\(OpcContext\)](#) evaluates to the value true.

Table of Contents

Constructors	1
OpcMethodDelegateCommand(OpcMethodExecuteCallback)	1
OpcMethodDelegateCommand(OpcMethodExecuteCallback, OpcMethodCanExecuteCallback)	1
OpcMethodDelegateCommand(OpcMethodExecuteCallback, OpcMethodCanExecuteCallback, OpcMethodCanExecuteCallback)	2
Methods	2
CanExecute(OpcContext)	3
CanUserExecute(OpcContext)	3
Execute(OpcMethodContext, IList<Object>, IList<Object>)	3

