OpcFileMethods Members

Namespace: Opc.UaFx.Client
Assemblies: Opc.UaFx.Advanced.dll, Opc.UaFx.Advanced.dll
The OpcFileMethods type exposes the following members.

Methods

Close(OpcClient, OpcNodeId, UInt32)

Closes the file maintained by the file node referenced by the `fileNodeId` specified using the further specified OpcClient.

C#

```
[CLSCompliant(false)]
public static void Close(OpcClient client, OpcNodeId fileNodeId, uint fileHandle)
```

Parameters

`client` OpcClient

The OpcClient to use to call the method.

`fileNodeId` OpcNodeId

The OpcNodeId of the file node its 'Close' method is to be called.

`fileHandle` UInt32

The numeric file handle acquired by one of the Open methods.

Exceptions

ArgumentNullException

The `client` or `fileNodeId` is a null reference (Nothing in Visual Basic).

InvalidOperationException

The file node does not exist.

NotSupportedException

The 'Close' method node does not exist or the method node signature does not define a valid method node to close a file handle.

OpcException

The call of the method node failed (see exception details for more information).

Close(OpcClient, OpcNodeId, UInt32, OpcCallMethodOptions)

Closes the file maintained by the file node referenced by the `fileNodeId` specified using the further specified OpcClient.

C#

```
[CLSCompliant(false)]
public static void Close(OpcClient client, OpcNodeId fileNodeId, uint fileHandle,
OpcCallMethodOptions options)
```

Parameters

`client` OpcClient

The OpcClient to use to call the method.

`fileNodeId` OpcNodeId

The OpcNodeId of the file node its 'Close' method is to be called.

`fileHandle` UInt32

The numeric file handle acquired by one of the Open methods.

`options` OpcCallMethodOptions

The OpcCallMethodOptions to apply before and after calling the method node to close the file handle.

Exceptions

ArgumentNullException

The `client` or `fileNodeId` is a null reference (Nothing in Visual Basic).

InvalidOperationException

The file node does not exist.

NotSupportedException

The 'Close' method node does not exist or the method node signature does not define a valid method node to close a file handle.

OpcException

The call of the method node failed (see exception details for more information).

Close(OpcFileNodeContext, UInt32)

Closes the file maintained by the file node referenced by the `context` specified.

C#

```
[CLSCompliant(false)]
public static void Close(OpcFileNodeContext context, uint fileHandle)
```

Parameters

`context` OpcFileNodeContext

The OpcFileNodeContext to use to access the file node.

`fileHandle` UInt32

The numeric file handle acquired by one of the Open methods.

Exceptions

ArgumentNullException

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

InvalidOperationException

The file node does not exist.

NotSupportedException

The 'Close' method node does not exist or the method node signature does not define a valid method node to close a file handle.

OpcException

The call of the method node failed (see exception details for more information).

Close(OpcFileNodeContext, UInt32, OpcCallMethodOptions)

Closes the file maintained by the file node referenced by the `context` specified.

C#

```
[CLSCompliant(false)]
public static void Close(OpcFileNodeContext context, uint fileHandle, OpcCallMethodOptions
options)
```

Parameters

`context` OpcFileNodeContext

The OpcFileNodeContext to use to access the file node.

`fileHandle` UInt32

The numeric file handle acquired by one of the Open methods.

`options` OpcCallMethodOptions

The OpcCallMethodOptions to apply before and after calling the method node to close the file handle.

Exceptions

ArgumentNullException

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

InvalidOperationException

The file node does not exist.

NotSupportedException

The 'Close' method node does not exist or the method node signature does not define a valid method node to close a file handle.

OpcException

The call of the method node failed (see exception details for more information).

GetPosition(OpcClient, OpcNodeId, UInt32)

Retrieves the position within the file maintained by the file node referenced by the `fileNodeId` specified using the further specified OpcClient.

C#

```
[CLSCompliant(false)]
public static ulong GetPosition(OpcClient client, OpcNodeId fileNodeId, uint fileHandle)
```

Parameters

`client` OpcClient

The OpcClient to use to call the method.

`fileNodeId` OpcNodeId

The OpcNodeId of the file node its 'GetPosition' method node is to be called.

`fileHandle` UInt32

The numeric file handle acquired by one of the Open methods.

Returns

UInt64

The current position in the file.

Exceptions

ArgumentNullException

The `client` or `fileNodeId` is a null reference (Nothing in Visual Basic).

InvalidOperationException

The file node does not exist.

NotSupportedException

The 'GetPosition' method node does not exist or the method node signature does not define a valid method node to query the position in a file.

OpcException

The call of the method node failed (see exception details for more information).

GetPosition(OpcClient, OpcNodeId, UInt32, OpcCallMethodOptions)

Retrieves the position within the file maintained by the file node referenced by the `fileNodeId` specified using the further specified OpcClient.

C#

```
[CLSCompliant(false)]
public static ulong GetPosition(OpcClient client, OpcNodeId fileNodeId, uint fileHandle,
OpcCallMethodOptions options)
```

Parameters

`client` OpcClient

The OpcClient to use to call the method.

`fileNodeId` OpcNodeId

The OpcNodeId of the file node its 'GetPosition' method node is to be called.

`fileHandle` UInt32

The numeric file handle acquired by one of the Open methods.

`options` OpcCallMethodOptions

The OpcCallMethodOptions to apply before and after calling the method node to query the position in a file.

Returns

UInt64

The current position in the file.

Exceptions

ArgumentNullException

The `client` or `fileNodeId` is a null reference (Nothing in Visual Basic).

InvalidOperationException

The file node does not exist.

NotSupportedException

The 'GetPosition' method node does not exist or the method node signature does not define a valid method node to query the position in a file.

OpcException

The call of the method node failed (see exception details for more information).

GetPosition(OpcFileNodeContext, UInt32)

Retrieves the position within the file maintained by the file node referenced by the `context` specified.

C#

```
[CLSCompliant(false)]
public static ulong GetPosition(OpcFileNodeContext context, uint fileHandle)
```

Parameters

`context` OpcFileNodeContext

The OpcFileNodeContext to use to access the file node.

`fileHandle` UInt32

The numeric file handle acquired by one of the Open methods.

Returns

UInt64

The current position in the file.

Exceptions

ArgumentNullException

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

InvalidOperationException

The file node does not exist.

NotSupportedException

The 'GetPosition' method node does not exist or the method node signature does not define a valid method node to query the position in a file.

OpcException

The call of the method node failed (see exception details for more information).

GetPosition(OpcFileNodeContext, UInt32, OpcCallMethodOptions)

Retrieves the position within the file maintained by the file node referenced by the `context` specified.

C#

```
[CLSCompliant(false)]
public static ulong GetPosition(OpcFileNodeContext context, uint fileHandle,
OpcCallMethodOptions options)
```

Parameters

`context` OpcFileNodeContext

The OpcFileNodeContext to use to access the file node.

`fileHandle` UInt32

The numeric file handle acquired by one of the Open methods.

`options` OpcCallMethodOptions

The OpcCallMethodOptions to apply before and after calling the method node to query the position in a file.

Returns

UInt64

The current position in the file.

Exceptions

ArgumentNullException

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

InvalidOperationException

The file node does not exist.

NotSupportedException

The 'GetPosition' method node does not exist or the method node signature does not define a valid method node to query the position in a file.

OpcException

The call of the method node failed (see exception details for more information).

IsFileNode(OpcClient, OpcNodeId)

Inspects the file node referenced by the `fileNodeId` specified using the further specified OpcClient whether the node fulfills the requirements of accessing it in the manner of a OPC UA FileType.

C#

```
public static bool IsFileNode(OpcClient client, OpcNodeId fileNodeId)
```

Parameters

`client` OpcClient

The OpcClient to use to call the method.

`fileNodeId` OpcNodeId

The OpcNodeId of the file node its 'Open' method node is to be called.

Returns

Boolean

The value true if the supposed file node is concrete file node; otherwise the value false.

Exceptions

ArgumentNullException

The `client` or `fileNodeId` is a null reference (Nothing in Visual Basic).

IsFileNode(OpcFileNodeContext)

Inspects the file node referenced by the `context` specified whether the node fulfills the requirements of accessing it in the manner of a OPC UA FileType.

C#

```
public static bool IsFileNode(OpcFileNodeContext context)
```

Parameters

`context` OpcFileNodeContext

The OpcFileNodeContext to use to inspect the supposed file node.

Returns

Boolean

The value true if the supposed file node is concrete file node; otherwise the value false.

Exceptions

ArgumentNullException

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

Open(OpcClient, OpcNodeId, OpcFileMode)

Opens the file maintained by the file node referenced by the `fileNodeId` specified using the further specified OpcClient.

C#

```
[CLSCompliant(false)]
public static uint Open(OpcClient client, OpcNodeId fileNodeId, OpcFileMode mode)
```

Parameters

`client` OpcClient

The OpcClient to use to call the method.

`fileNodeId` OpcNodeId

The OpcNodeId of the file node its 'Open' method node is to be called.

`mode` OpcFileMode

One of the OpcFileMode members or a combination of them. The value defines the type of requested access and how the system should open an existing file.

Returns

UInt32

The session based file handle to use to identify the file access requested for subsequent file method calls. To release the acquired file handle call one of the Close methods.

Exceptions

ArgumentNullException

The `client` or `fileNodeId` is a null reference (Nothing in Visual Basic).

InvalidOperationException

The file node does not exist.

NotSupportedException

The 'Open' method node does not exist or the method node signature does not define a valid method node to open a file handle.

OpcException

The call of the method node failed (see exception details for more information).

Open(OpcClient, OpcNodeId, OpcFileMode, OpcCallMethodOptions)

Opens the file maintained by the file node referenced by the `fileNodeId` specified using the further specified OpcClient.

C#

```
[CLSCompliant(false)]
public static uint Open(OpcClient client, OpcNodeId fileNodeId, OpcFileMode mode,
OpcCallMethodOptions options)
```

Parameters

`client` OpcClient

The OpcClient to use to call the method.

`fileNodeId` OpcNodeId

The OpcNodeId of the file node its 'Open' method node is to be called.

`mode` OpcFileMode

One of the OpcFileMode members or a combination of them. The value defines the type of requested access and how the system should open an existing file.

`options` OpcCallMethodOptions

The OpcCallMethodOptions to apply before and after calling the method node to open a file handle.

Returns

UInt32

The session based file handle to use to identify the file access requested for subsequent file method calls. To release the acquired file handle call one of the Close methods.

Exceptions

ArgumentNullException

The `client` or `fileNodeId` is a null reference (Nothing in Visual Basic).

InvalidOperationException

The file node does not exist.

NotSupportedException

The 'Open' method node does not exist or the method node signature does not define a valid method node to open a file handle.

OpcException

The call of the method node failed (see exception details for more information).

Open(OpcFileNodeContext, OpcFileMode)

Opens the file maintained by the file node referenced by the `context` specified.

C#

```
[CLSCompliant(false)]
public static uint Open(OpcFileNodeContext context, OpcFileMode mode)
```

Parameters

`context` OpcFileNodeContext

The OpcFileNodeContext to use to access the file node.

`mode` OpcFileMode

One of the OpcFileMode members or a combination of them. The value defines the type of requested access and how the system should open an existing file.

Returns

UInt32

The session based file handle to use to identify the file access requested for subsequent file method calls. To release the acquired file handle call one of the Close methods.

Exceptions

ArgumentNullException

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

InvalidOperationException

The file node does not exist.

NotSupportedException

The 'Open' method node does not exist or the method node signature does not define a valid method node to open a file handle.

OpcException

The call of the method node failed (see exception details for more information).

Open(OpcFileNodeContext, OpcFileMode, OpcCallMethodOptions)

Opens the file maintained by the file node referenced by the `context` specified.

C#

```
[CLSCompliant(false)]
public static uint Open(OpcFileNodeContext context, OpcFileMode mode, OpcCallMethodOptions
options)
```

Parameters

`context` OpcFileNodeContext

The OpcFileNodeContext to use to access the file node.

`mode` OpcFileMode

One of the OpcFileMode members or a combination of them. The value defines the type of requested access and how the system should open an existing file.

`options` OpcCallMethodOptions

The OpcCallMethodOptions to apply before and after calling the method node to open a file handle.

Returns

UInt32

The session based file handle to use to identify the file access requested for subsequent file method calls. To release the acquired file handle call one of the Close methods.

Exceptions

ArgumentNullException

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

InvalidOperationException

The file node does not exist.

NotSupportedException

The 'Open' method node does not exist or the method node signature does not define a valid method node to open a file handle.

OpcException

The call of the method node failed (see exception details for more information).

Read(OpcClient, OpcNodeId, UInt32, Int32)

Reads the number of bytes specified by `length` from the file maintained by the file node referenced by the `fileNodeId` specified using the further specified OpcClient.

C#

```
[CLSCompliant(false)]
public static byte[] Read(OpcClient client, OpcNodeId fileNodeId, uint fileHandle, int
length)
```

Parameters

`client` OpcClient

The OpcClient to use to call the method.

`fileNodeId` OpcNodeId

The OpcNodeId of the file node its 'Read' method node is to be called.

`fileHandle` UInt32

The numeric file handle acquired by one of the Open methods.

`length` Int32

The number of bytes to read starting from the current position in the file.

Returns

Byte[]

The data read from the file. If the end of file is reached only the data till the end of the file is read. If the specified length is longer than the maximum allowed message size of the communication, only those data fitting into the message size is read.

Exceptions

ArgumentNullException

The `client` or `fileNodeId` is a null reference (Nothing in Visual Basic).

InvalidOperationException

The file node does not exist.

NotSupportedException

The 'Read' method node does not exist or the method node signature does not define a valid method node to read data from a file.

OpcException

The call of the method node failed (see exception details for more information).

Read(OpcClient, OpcNodeId, UInt32, Int32, OpcCallMethodOptions)

Reads the number of bytes specified by `length` from the file maintained by the file node referenced by the `fileNodeId` specified using the further specified OpcClient.

C#

```
[CLSCompliant(false)]
public static byte[] Read(OpcClient client, OpcNodeId fileNodeId, uint fileHandle, int
length, OpcCallMethodOptions options)
```

Parameters

`client` OpcClient

The OpcClient to use to call the method.

`fileNodeId` OpcNodeId

The OpcNodeId of the file node its 'Read' method node is to be called.

`fileHandle` UInt32

The numeric file handle acquired by one of the Open methods.

`length` Int32

The number of bytes to read starting from the current position in the file.

`options` OpcCallMethodOptions

The OpcCallMethodOptions to apply before and after calling the method node to read data from a file.

Returns

Byte[]

The data read from the file. If the end of file is reached only the data till the end of the file is read. If the specified length is longer than the maximum allowed message size of the communication, only those data fitting into the message size is read.

Exceptions

ArgumentNullException

The `client` or `fileNodeId` is a null reference (Nothing in Visual Basic).

InvalidOperationException

The file node does not exist.

NotSupportedException

The 'Read' method node does not exist or the method node signature does not define a valid method node to read data from a file.

OpcException

The call of the method node failed (see exception details for more information).

Read(OpcFileNodeContext, UInt32, Int32)

Reads the number of bytes specified by `length` from the file maintained by the file node referenced by the `context` specified.

C#

```
[CLSCompliant(false)]
public static byte[] Read(OpcFileNodeContext context, uint fileHandle, int length)
```

Parameters

`context` OpcFileNodeContext

The OpcFileNodeContext to use to access the file node.

`fileHandle` UInt32

The numeric file handle acquired by one of the Open methods.

`length` Int32

The number of bytes to read starting from the current position in the file.

Returns

Byte[]

The data read from the file. If the end of file is reached only the data till the end of the file is read. If the specified length is longer than the maximum allowed message size of the communication, only those data fitting into the message size is read.

Exceptions

ArgumentNullException

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

InvalidOperationException

The file node does not exist.

NotSupportedException

The 'Read' method node does not exist or the method node signature does not define a valid method node to read data from a file.

OpcException

The call of the method node failed (see exception details for more information).

Read(OpcFileNodeContext, UInt32, Int32, OpcCallMethodOptions)

Reads the number of bytes specified by `length` from the file maintained by the file node referenced by the `context` specified.

C#

```
[CLSCompliant(false)]
public static byte[] Read(OpcFileNodeContext context, uint fileHandle, int length,
OpcCallMethodOptions options)
```

Parameters

`context` OpcFileNodeContext

The OpcFileNodeContext to use to access the file node.

`fileHandle` UInt32

The numeric file handle acquired by one of the Open methods.

`length` Int32

The number of bytes to read starting from the current position in the file.

`options` OpcCallMethodOptions

The OpcCallMethodOptions to apply before and after calling the method node to read data from a file.

Returns

Byte[]

The data read from the file. If the end of file is reached only the data till the end of the file is read. If the specified length is longer than the maximum allowed message size of the communication, only those data fitting into the message size is read.

Exceptions

ArgumentNullException

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

InvalidOperationException

The file node does not exist.

NotSupportedException

The 'Read' method node does not exist or the method node signature does not define a valid method node to read data from a file.

OpcException

The call of the method node failed (see exception details for more information).

SecureClose(OpcClient, OpcNodeId, SafeOpcFileHandle)

Closes the file maintained by the file node referenced by the `nodeId` specified using the further specified OpcClient.

C#

```
public static void SecureClose(OpcClient client, OpcNodeId nodeId, SafeOpcFileHandle
fileHandle)
```

Parameters

`client` OpcClient

The OpcClient to use to call the method.

`nodeId` OpcNodeId

The OpcNodeId of the file node its 'Close' method is to be called.

`fileHandle` SafeOpcFileHandle

The SafeOpcFileHandle acquired by one of the SecureOpen methods.

Exceptions

ArgumentNullException

The `client`, `nodeId` or `fileHandle` is a null reference (Nothing in Visual Basic).

InvalidOperationException

The file node does not exist.

NotSupportedException

The 'Close' method node does not exist or the method node signature does not define a valid method node to close a file handle.

OpcException

The call of the method node failed (see exception details for more information).

SecureClose(OpcClient, OpcNodeId, SafeOpcFileHandle, OpcCallMethodOptions)

Closes the file maintained by the file node referenced by the `nodeId` specified using the further specified OpcClient.

C#

```
public static void SecureClose(OpcClient client, OpcNodeId nodeId, SafeOpcFileHandle
fileHandle, OpcCallMethodOptions options)
```

Parameters

`client` OpcClient

The OpcClient to use to call the method.

`nodeId` OpcNodeId

The OpcNodeId of the file node its 'Close' method is to be called.

`fileHandle` SafeOpcFileHandle

The SafeOpcFileHandle acquired by one of the SecureOpen methods.

`options` OpcCallMethodOptions

The OpcCallMethodOptions to apply before and after calling the method node to close the file handle.

Exceptions

ArgumentNullException

The `client`, `nodeId` or `fileHandle` is a null reference (Nothing in Visual Basic).

InvalidOperationException

The file node does not exist.

NotSupportedException

The 'Close' method node does not exist or the method node signature does not define a valid method node to close a file handle.

OpcException

The call of the method node failed (see exception details for more information).

SecureClose(OpcFileNodeContext, SafeOpcFileHandle)

Closes the file maintained by the file node referenced by the `context` specified.

C#

```
public static void SecureClose(OpcFileNodeContext context, SafeOpcFileHandle fileHandle)
```

Parameters

`context` OpcFileNodeContext

The OpcFileNodeContext to use to access the file node.

`fileHandle` SafeOpcFileHandle

The SafeOpcFileHandle acquired by one of the SecureOpen methods.

Exceptions

ArgumentNullException

The `context` or `fileHandle` is a null reference (Nothing in Visual Basic).

InvalidOperationException

The file node does not exist.

NotSupportedException

The 'Close' method node does not exist or the method node signature does not define a valid method node to close a file handle.

OpcException

The call of the method node failed (see exception details for more information).

SecureClose(OpcFileNodeContext, SafeOpcFileHandle, OpcCallMethodOptions)

Closes the file maintained by the file node referenced by the `context` specified.

C#

```
public static void SecureClose(OpcFileNodeContext context, SafeOpcFileHandle fileHandle,
OpcCallMethodOptions options)
```

Parameters

`context` OpcFileNodeContext

The OpcFileNodeContext to use to access the file node.

`fileHandle` SafeOpcFileHandle

The SafeOpcFileHandle acquired by one of the SecureOpen methods.

`options` OpcCallMethodOptions

The OpcCallMethodOptions to apply before and after calling the method node to close the file handle.

Exceptions

ArgumentNullException

The `context` or `fileHandle` is a null reference (Nothing in Visual Basic).

InvalidOperationException

The file node does not exist.

NotSupportedException

The 'Close' method node does not exist or the method node signature does not define a valid method node to close a file handle.

OpcException

The call of the method node failed (see exception details for more information).

SecureClose(SafeOpcFileHandle)

Closes the file `fileHandle` specified.

C#

```
public static void SecureClose(SafeOpcFileHandle fileHandle)
```

Parameters

`fileHandle` SafeOpcFileHandle

The SafeOpcFileHandle acquired by one of the SecureOpen methods.

Exceptions

ArgumentNullException

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

InvalidOperationException

The file node does not exist.

NotSupportedException

The 'Close' method node does not exist or the method node signature does not define a valid method node to close a file handle.

OpcException

The call of the method node failed (see exception details for more information).

SecureClose(SafeOpcFileHandle, OpcCallMethodOptions)

Closes the file `fileHandle` specified.

C#

```
public static void SecureClose(SafeOpcFileHandle fileHandle, OpcCallMethodOptions options)
```

Parameters

`fileHandle` SafeOpcFileHandle

The SafeOpcFileHandle acquired by one of the SecureOpen methods.

`options` OpcCallMethodOptions

The OpcCallMethodOptions to apply before and after calling the method node to close the file handle.

Exceptions

ArgumentNullException

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

InvalidOperationException

The file node does not exist.

NotSupportedException

The 'Close' method node does not exist or the method node signature does not define a valid method node to close a file handle.

OpcException

The call of the method node failed (see exception details for more information).

SecureGetPosition(OpcClient, OpcNodeId, SafeOpcFileHandle)

Retrieves the position within the file maintained by the file node referenced by the `fileNodeId` specified using the further specified OpcClient.

C#

```
public static long SecureGetPosition(OpcClient client, OpcNodeId fileNodeId,
SafeOpcFileHandle fileHandle)
```

Parameters

`client` OpcClient

The OpcClient to use to call the method.

`fileNodeId` OpcNodeId

The OpcNodeId of the file node its 'GetPosition' method node is to be called.

`fileHandle` SafeOpcFileHandle

The SafeOpcFileHandle acquired by one of the SecureOpen methods.

Returns

Int64

The current position in the file.

Exceptions

ArgumentNullException

The `client`, `fileNodeId` or `fileHandle` is a null reference (Nothing in Visual Basic).

InvalidOperationException

The file node does not exist.

NotSupportedException

The 'GetPosition' method node does not exist or the method node signature does not define a valid method node to query the position in a file.

OpcException

The call of the method node failed (see exception details for more information).

SecureGetPosition(OpcClient, OpcNodeId, SafeOpcFileHandle, OpcCallMethodOptions)

Retrieves the position within the file maintained by the file node referenced by the `fileNodeId` specified using the further specified OpcClient.

C#

```
public static long SecureGetPosition(OpcClient client, OpcNodeId fileNodeId,
SafeOpcFileHandle fileHandle, OpcCallMethodOptions options)
```

Parameters

`client` OpcClient

The OpcClient to use to call the method.

`fileNodeId` OpcNodeId

The OpcNodeId of the file node its 'GetPosition' method node is to be called.

`fileHandle` SafeOpcFileHandle

The SafeOpcFileHandle acquired by one of the SecureOpen methods.

`options` OpcCallMethodOptions

The OpcCallMethodOptions to apply before and after calling the method node to query the position in a file.

Returns

Int64

The current position in the file.

Exceptions

ArgumentNullException

The `client`, `fileNodeId` or `fileHandle` is a null reference (Nothing in Visual Basic).

InvalidOperationException

The file node does not exist.

NotSupportedException

The 'GetPosition' method node does not exist or the method node signature does not define a valid method node to query the position in a file.

OpcException

The call of the method node failed (see exception details for more information).

SecureGetPosition(OpcFileNodeContext, SafeOpcFileHandle)

Retrieves the position within the file maintained by the file node referenced by the `context` specified.

C#

```
public static long SecureGetPosition(OpcFileNodeContext context, SafeOpcFileHandle
fileHandle)
```

Parameters

`context` OpcFileNodeContext

The OpcFileNodeContext to use to access the file node.

`fileHandle` SafeOpcFileHandle

The SafeOpcFileHandle acquired by one of the SecureOpen methods.

Returns

Int64

The current position in the file.

Exceptions

ArgumentNullException

The `context` or `fileHandle` is a null reference (Nothing in Visual Basic).

InvalidOperationException

The file node does not exist.

NotSupportedException

The 'GetPosition' method node does not exist or the method node signature does not define a valid method node to query the position in a file.

OpcException

The call of the method node failed (see exception details for more information).

SecureGetPosition(OpcFileNodeContext, SafeOpcFileHandle, OpcCallMethodOptions)

Retrieves the position within the file maintained by the file node referenced by the `context` specified.

C#

```
public static long SecureGetPosition(OpcFileNodeContext context, SafeOpcFileHandle
fileHandle, OpcCallMethodOptions options)
```

Parameters

`context` OpcFileNodeContext

The OpcFileNodeContext to use to access the file node.

`fileHandle` SafeOpcFileHandle

The SafeOpcFileHandle acquired by one of the SecureOpen methods.

`options` OpcCallMethodOptions

The OpcCallMethodOptions to apply before and after calling the method node to query the position in a file.

Returns

Int64

The current position in the file.

Exceptions

ArgumentNullException

The `context` or `fileHandle` is a null reference (Nothing in Visual Basic).

InvalidOperationException

The file node does not exist.

NotSupportedException

The 'GetPosition' method node does not exist or the method node signature does not define a valid method node to query the position in a file.

OpcException

The call of the method node failed (see exception details for more information).

SecureGetPosition(SafeOpcFileHandle)

Retrieves the position within the file identified by the `fileHandle` specified.

C#

```
public static long SecureGetPosition(SafeOpcFileHandle fileHandle)
```

Parameters

`fileHandle` SafeOpcFileHandle

The SafeOpcFileHandle acquired by one of the SecureOpen methods.

Returns

Int64

The current position in the file.

Exceptions

ArgumentNullException

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

InvalidOperationException

The file node does not exist.

NotSupportedException

The 'GetPosition' method node does not exist or the method node signature does not define a valid method node to query the position in a file.

OpcException

The call of the method node failed (see exception details for more information).

SecureGetPosition(SafeOpcFileHandle, OpcCallMethodOptions)

Retrieves the position within the file identified by the `fileHandle` specified.

C#

```
public static long SecureGetPosition(SafeOpcFileHandle fileHandle, OpcCallMethodOptions
options)
```

Parameters

`fileHandle` SafeOpcFileHandle

The SafeOpcFileHandle acquired by one of the SecureOpen methods.

`options` OpcCallMethodOptions

The OpcCallMethodOptions to apply before and after calling the method node to query the position in a file.

Returns

Int64

The current position in the file.

Exceptions

ArgumentNullException

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

InvalidOperationException

The file node does not exist.

NotSupportedException

The 'GetPosition' method node does not exist or the method node signature does not define a valid method node to query the position in a file.

OpcException

The call of the method node failed (see exception details for more information).

SecureOpen(OpcClient, OpcNodeId, OpcFileMode)

Opens the file maintained by the file node referenced by the `fileNodeId` specified using the further specified OpcClient.

C#

```
public static SafeOpcFileHandle SecureOpen(OpcClient client, OpcNodeId fileNodeId,
OpcFileMode mode)
```

Parameters

`client` OpcClient

The OpcClient to use to call the method.

`fileNodeId` OpcNodeId

The OpcNodeId of the file node its 'Open' method node is to be called.

`mode` OpcFileMode

One of the OpcFileMode members or a combination of them. The value defines the type of requested access and how the system should open an existing file.

Returns

SafeOpcFileHandle

The session based SafeOpcFileHandle to use to identify the file access requested for subsequent file method calls. To release the acquired file handle call one of the SecureClose methods or call the 'Close'

method of the handle provided.

Exceptions

ArgumentNullException

The `client` or `fileNodeId` is a null reference (Nothing in Visual Basic).

InvalidOperationException

The file node does not exist.

NotSupportedException

The 'Open' method node does not exist or the method node signature does not define a valid method node to open a file handle.

OpcException

The call of the method node failed (see exception details for more information).

SecureOpen(OpcClient, OpcNodeId, OpcFileMode, OpcCallMethodOptions)

Opens the file maintained by the file node referenced by the `fileNodeId` specified using the further specified OpcClient.

C#

```
public static SafeOpcFileHandle SecureOpen(OpcClient client, OpcNodeId fileNodeId,
OpcFileMode mode, OpcCallMethodOptions options)
```

Parameters

`client` OpcClient

The OpcClient to use to call the method.

`fileNodeId` OpcNodeId

The OpcNodeId of the file node its 'Open' method node is to be called.

`mode` OpcFileMode

One of the OpcFileMode members or a combination of them. The value defines the type of requested access and how the system should open an existing file.

`options` OpcCallMethodOptions

The OpcCallMethodOptions to apply before and after calling the method node to open a file handle.

Returns

SafeOpcFileHandle

The session based SafeOpcFileHandle to use to identify the file access requested for subsequent file method calls. To release the acquired file handle call one of the SecureClose methods or call the 'Close' method of the handle provided.

Exceptions

ArgumentNullException

The `client` or `fileNodeId` is a null reference (Nothing in Visual Basic).

InvalidOperationException

The file node does not exist.

NotSupportedException

The 'Open' method node does not exist or the method node signature does not define a valid method node to open a file handle.

OpcException

The call of the method node failed (see exception details for more information).

SecureOpen(OpcFileNodeContext, OpcFileMode)

Opens the file maintained by the file node referenced by the `context` specified.

C#

```
public static SafeOpcFileHandle SecureOpen(OpcFileNodeContext context, OpcFileMode mode)
```

Parameters

`context` OpcFileNodeContext

The OpcFileNodeContext to use to access the file node.

`mode` OpcFileMode

One of the OpcFileMode members or a combination of them. The value defines the type of requested access and how the system should open an existing file.

Returns

SafeOpcFileHandle

The session based SafeOpcFileHandle to use to identify the file access requested for subsequent file method calls. To release the acquired file handle call one of the SecureClose methods or call the 'Close' method of the handle provided.

Exceptions

ArgumentNullException

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

InvalidOperationException

The file node does not exist.

NotSupportedException

The 'Open' method node does not exist or the method node signature does not define a valid method node to open a file handle.

OpcException

The call of the method node failed (see exception details for more information).

SecureOpen(OpcFileNodeContext, OpcFileMode, OpcCallMethodOptions)

Opens the file maintained by the file node referenced by the `context` specified.

C#

```
public static SafeOpcFileHandle SecureOpen(OpcFileNodeContext context, OpcFileMode mode,
OpcCallMethodOptions options)
```

Parameters

`context` OpcFileNodeContext

The OpcFileNodeContext to use to access the file node.

`mode` OpcFileMode

One of the OpcFileMode members or a combination of them. The value defines the type of requested access and how the system should open an existing file.

`options` OpcCallMethodOptions

The OpcCallMethodOptions to apply before and after calling the method node to open a file handle.

Returns

SafeOpcFileHandle

The session based SafeOpcFileHandle to use to identify the file access requested for subsequent file method calls. To release the acquired file handle call one of the SecureClose methods or call the 'Close' method of the handle provided.

Exceptions

ArgumentNullException

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

InvalidOperationException

The file node does not exist.

NotSupportedException

The 'Open' method node does not exist or the method node signature does not define a valid method node to open a file handle.

OpcException

The call of the method node failed (see exception details for more information).

SecureRead(OpcClient, OpcNodeId, SafeOpcFileHandle, Int32)

Reads the number of bytes specified by `length` from the file maintained by the file node referenced by the `fileNodeId` specified using the further specified OpcClient.

C#

```
public static byte[] SecureRead(OpcClient client, OpcNodeId fileNodeId, SafeOpcFileHandle
fileHandle, int length)
```

Parameters

`client` OpcClient

The OpcClient to use to call the method.

`fileNodeId` OpcNodeId

The OpcNodeId of the file node its 'Read' method node is to be called.

`fileHandle` SafeOpcFileHandle

The SafeOpcFileHandle acquired by one of the SecureOpen methods.

`length` Int32

The number of bytes to read starting from the current position in the file.

Returns

Byte[]

The data read from the file. If the end of file is reached only the data till the end of the file is read. If the specified length is longer than the maximum allowed message size of the communication, only those data fitting into the message size is read.

Exceptions

ArgumentNullException

The `client`, `fileNodeId` or `fileHandle` is a null reference (Nothing in Visual Basic).

InvalidOperationException

The file node does not exist.

NotSupportedException

The 'Read' method node does not exist or the method node signature does not define a valid method node to read data from a file.

OpcException

The call of the method node failed (see exception details for more information).

SecureRead(OpcClient, OpcNodeId, SafeOpcFileHandle, Int32, OpcCallMethodOptions)

Reads the number of bytes specified by `length` from the file maintained by the file node referenced by the `fileNodeId` specified using the further specified OpcClient.

C#

```
public static byte[] SecureRead(OpcClient client, OpcNodeId fileNodeId, SafeOpcFileHandle
fileHandle, int length, OpcCallMethodOptions options)
```

Parameters

`client` OpcClient

The OpcClient to use to call the method.

`fileNodeId` OpcNodeId

The OpcNodeId of the file node its 'Read' method node is to be called.

`fileHandle` SafeOpcFileHandle

The SafeOpcFileHandle acquired by one of the SecureOpen methods.

`length` Int32

The number of bytes to read starting from the current position in the file.

`options` OpcCallMethodOptions

The OpcCallMethodOptions to apply before and after calling the method node to read data from a file.

Returns

Byte[]

The data read from the file. If the end of file is reached only the data till the end of the file is read. If the specified length is longer than the maximum allowed message size of the communication, only those data fitting into the message size is read.

Exceptions

ArgumentNullException

The `client`, `fileNodeId` or `fileHandle` is a null reference (Nothing in Visual Basic).

InvalidOperationException

The file node does not exist.

NotSupportedException

The 'Read' method node does not exist or the method node signature does not define a valid method node to read data from a file.

OpcException

The call of the method node failed (see exception details for more information).

SecureRead(OpcFileNodeContext, SafeOpcFileHandle, Int32)

Reads the number of bytes specified by `length` from the file maintained by the file node referenced by the `context` specified.

C#

```
public static byte[] SecureRead(OpcFileNodeContext context, SafeOpcFileHandle fileHandle,
int length)
```

Parameters

`context` OpcFileNodeContext

The OpcFileNodeContext to use to access the file node.

`fileHandle` SafeOpcFileHandle

The SafeOpcFileHandle acquired by one of the SecureOpen methods.

`length` Int32

The number of bytes to read starting from the current position in the file.

Returns

Byte[]

The data read from the file. If the end of file is reached only the data till the end of the file is read. If the specified length is longer than the maximum allowed message size of the communication, only those data fitting into the message size is read.

Exceptions

ArgumentNullException

The `context` or `fileHandle` is a null reference (Nothing in Visual Basic).

InvalidOperationException

The file node does not exist.

NotSupportedException

The 'Read' method node does not exist or the method node signature does not define a valid method node to read data from a file.

OpcException

The call of the method node failed (see exception details for more information).

SecureRead(OpcFileNodeContext, SafeOpcFileHandle, Int32, OpcCallMethodOptions)

Reads the number of bytes specified by `length` from the file maintained by the file node referenced by the `context` specified.

C#

```
public static byte[] SecureRead(OpcFileNodeContext context, SafeOpcFileHandle fileHandle,
int length, OpcCallMethodOptions options)
```

Parameters

`context` OpcFileNodeContext

The OpcFileNodeContext to use to access the file node.

`fileHandle` SafeOpcFileHandle

The SafeOpcFileHandle acquired by one of the SecureOpen methods.

`length` Int32

The number of bytes to read starting from the current position in the file.

`options` OpcCallMethodOptions

The OpcCallMethodOptions to apply before and after calling the method node to read data from a file.

Returns

Byte[]

The data read from the file. If the end of file is reached only the data till the end of the file is read. If the specified length is longer than the maximum allowed message size of the communication, only those data fitting into the message size is read.

Exceptions

ArgumentNullException

The `context` or `fileHandle` is a null reference (Nothing in Visual Basic).

InvalidOperationException

The file node does not exist.

NotSupportedException

The 'Read' method node does not exist or the method node signature does not define a valid method node to read data from a file.

OpcException

The call of the method node failed (see exception details for more information).

SecureRead(SafeOpcFileHandle, Int32)

Reads the number of bytes specified by `length` from the file identified by the `fileHandle` specified.

C#

```
public static byte[] SecureRead(SafeOpcFileHandle fileHandle, int length)
```

Parameters

`fileHandle` SafeOpcFileHandle

The SafeOpcFileHandle acquired by one of the SecureOpen methods.

`length` Int32

The number of bytes to read starting from the current position in the file.

Returns

Byte[]

The data read from the file. If the end of file is reached only the data till the end of the file is read. If the specified length is longer than the maximum allowed message size of the communication, only those data fitting into the message size is read.

Exceptions

ArgumentNullException

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

InvalidOperationException

The file node does not exist.

NotSupportedException

The 'Read' method node does not exist or the method node signature does not define a valid method node to read data from a file.

OpcException

The call of the method node failed (see exception details for more information).

SecureRead(SafeOpcFileHandle, Int32, OpcCallMethodOptions)

Reads the number of bytes specified by `length` from the file identified by the `fileHandle` specified.

C#

```
public static byte[] SecureRead(SafeOpcFileHandle fileHandle, int length,
OpcCallMethodOptions options)
```

Parameters

`fileHandle` SafeOpcFileHandle

The SafeOpcFileHandle acquired by one of the SecureOpen methods.

`length` Int32

The number of bytes to read starting from the current position in the file.

`options` OpcCallMethodOptions

The OpcCallMethodOptions to apply before and after calling the method node to read data from a file.

Returns

Byte[]

The data read from the file. If the end of file is reached only the data till the end of the file is read. If the specified length is longer than the maximum allowed message size of the communication, only those data fitting into the message size is read.

Exceptions

ArgumentNullException

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

InvalidOperationException

The file node does not exist.

NotSupportedException

The 'Read' method node does not exist or the method node signature does not define a valid method node to read data from a file.

OpcException

The call of the method node failed (see exception details for more information).

SecureSetPosition(OpcClient, OpcNodeId, SafeOpcFileHandle, Int64)

Updates the position in the file maintained by the file node referenced by the `fileNodeId` specified using the further specified OpcClient using `position`.

C#

```
public static void SecureSetPosition(OpcClient client, OpcNodeId fileNodeId,
SafeOpcFileHandle fileHandle, long position)
```

Parameters

`client` OpcClient

The OpcClient to use to call the method.

`fileNodeId` OpcNodeId

The OpcNodeId of the file node its 'SetPosition' method node is to be called.

`fileHandle` SafeOpcFileHandle

The SafeOpcFileHandle acquired by one of the SecureOpen methods.

`position` Int64

The new position in the file. If `position` is greater than the file size the position is set to the end of the file.

Exceptions

ArgumentNullException

The `client`, `fileNodeId` or `fileHandle` is a null reference (Nothing in Visual Basic).

InvalidOperationException

The file node does not exist.

NotSupportedException

The 'SetPosition' method node does not exist or the method node signature does not define a valid method node to update the position in a file.

OpcException

The call of the method node failed (see exception details for more information).

SecureSetPosition(OpcClient, OpcNodeId, SafeOpcFileHandle, Int64, OpcCallMethodOptions)

Updates the position in the file maintained by the file node referenced by the `fileNodeId` specified using the further specified OpcClient using `position`.

C#

```
public static void SecureSetPosition(OpcClient client, OpcNodeId fileNodeId,
SafeOpcFileHandle fileHandle, long position, OpcCallMethodOptions options)
```

Parameters

`client` OpcClient

The OpcClient to use to call the method.

`fileNodeId` OpcNodeId

The OpcNodeId of the file node its 'SetPosition' method node is to be called.

`fileHandle` SafeOpcFileHandle

The SafeOpcFileHandle acquired by one of the SecureOpen methods.

`position` Int64

The new position in the file. If `position` is greater than the file size the position is set to the end of the file.

`options` OpcCallMethodOptions

The OpcCallMethodOptions to apply before and after calling the method node to update the position in a file.

Exceptions

ArgumentNullException

The `client`, `fileNodeId` or `fileHandle` is a null reference (Nothing in Visual Basic).

InvalidOperationException

The file node does not exist.

NotSupportedException

The 'SetPosition' method node does not exist or the method node signature does not define a valid method node to update the position in a file.

OpcException

The call of the method node failed (see exception details for more information).

SecureSetPosition(OpcFileNodeContext, SafeOpcFileHandle, Int64)

Updates the position in the file maintained by the file node referenced by the `context` specified using `position`.

C#

```
public static void SecureSetPosition(OpcFileNodeContext context, SafeOpcFileHandle
fileHandle, long position)
```

Parameters

`context` OpcFileNodeContext

The OpcFileNodeContext to use to access the file node.

`fileHandle` SafeOpcFileHandle

The SafeOpcFileHandle acquired by one of the SecureOpen methods.

`position` Int64

The new position in the file. If `position` is greater than the file size the position is set to the end of the file.

Exceptions

ArgumentNullException

The `context` or `fileHandle` is a null reference (Nothing in Visual Basic).

InvalidOperationException

The file node does not exist.

NotSupportedException

The 'SetPosition' method node does not exist or the method node signature does not define a valid method node to update the position in a file.

OpcException

The call of the method node failed (see exception details for more information).

SecureSetPosition(OpcFileNodeContext, SafeOpcFileHandle, Int64, OpcCallMethodOptions)

Updates the position in the file maintained by the file node referenced by the `context` specified using `position`.

C#

```
public static void SecureSetPosition(OpcFileNodeContext context, SafeOpcFileHandle
fileHandle, long position, OpcCallMethodOptions options)
```

Parameters

`context` OpcFileNodeContext

The OpcFileNodeContext to use to access the file node.

`fileHandle` SafeOpcFileHandle

The SafeOpcFileHandle acquired by one of the SecureOpen methods.

`position` Int64

The new position in the file. If `position` is greater than the file size the position is set to the end of the file.

`options` OpcCallMethodOptions

The OpcCallMethodOptions to apply before and after calling the method node to update the position in a file.

Exceptions

ArgumentNullException

The `context` or `fileHandle` is a null reference (Nothing in Visual Basic).

InvalidOperationException

The file node does not exist.

NotSupportedException

The 'SetPosition' method node does not exist or the method node signature does not define a valid method node to update the position in a file.

OpcException

The call of the method node failed (see exception details for more information).

SecureSetPosition(SafeOpcFileHandle, Int64)

Updates the position in the file identified by the `fileHandle` specified using `position`.

C#

```
public static void SecureSetPosition(SafeOpcFileHandle fileHandle, long position)
```

Parameters

`fileHandle` SafeOpcFileHandle

The SafeOpcFileHandle acquired by one of the SecureOpen methods.

`position` Int64

The new position in the file. If `position` is greater than the file size the position is set to the end of the file.

Exceptions

ArgumentNullException

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

InvalidOperationException

The file node does not exist.

NotSupportedException

The 'SetPosition' method node does not exist or the method node signature does not define a valid method

node to update the position in a file.

OpcException

The call of the method node failed (see exception details for more information).

SecureSetPosition(SafeOpcFileHandle, Int64, OpcCallMethodOptions)

Updates the position in the file identified by the `fileHandle` specified using `position`.

C#

```
public static void SecureSetPosition(SafeOpcFileHandle fileHandle, long position,
OpcCallMethodOptions options)
```

Parameters

`fileHandle` SafeOpcFileHandle

The SafeOpcFileHandle acquired by one of the SecureOpen methods.

`position` Int64

The new position in the file. If `position` is greater than the file size the position is set to the end of the file.

`options` OpcCallMethodOptions

The OpcCallMethodOptions to apply before and after calling the method node to update the position in a file.

Exceptions

ArgumentNullException

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

InvalidOperationException

The file node does not exist.

NotSupportedException

The 'SetPosition' method node does not exist or the method node signature does not define a valid method node to update the position in a file.

OpcException

The call of the method node failed (see exception details for more information).

SecureWrite(OpcClient, OpcNodeId, SafeOpcFileHandle,

Byte[])

Writes the `data` to the file maintained by the file node referenced by the `fileNodeId` specified using the further specified OpcClient.

C#

```
public static void SecureWrite(OpcClient client, OpcNodeId fileNodeId, SafeOpcFileHandle
fileHandle, byte[] data)
```

Parameters

`client` OpcClient

The OpcClient to use to call the method.

`fileNodeId` OpcNodeId

The OpcNodeId of the file node its 'Write' method node is to be called.

`fileHandle` SafeOpcFileHandle

The SafeOpcFileHandle acquired by one of the SecureOpen methods.

`data` Byte[]

The data to write to the file at the current position.

Exceptions

ArgumentNullException

The `client`, `fileNodeId`, `fileHandle` or `data` is a null reference (Nothing in Visual Basic).

InvalidOperationException

The file node does not exist.

NotSupportedException

The 'Write' method node does not exist or the method node signature does not define a valid method node to write data to a file.

OpcException

The call of the method node failed (see exception details for more information).

Remarks

It is server-dependent whether the written data is persistently stored if the session is ended without calling the Close method.

SecureWrite(OpcClient, OpcNodeId, SafeOpcFileHandle,

Byte[], OpcCallMethodOptions)

Writes the `data` to the file maintained by the file node referenced by the `fileNodeId` specified using the further specified OpcClient.

C#

```
public static void SecureWrite(OpcClient client, OpcNodeId fileNodeId, SafeOpcFileHandle
fileHandle, byte[] data, OpcCallMethodOptions options)
```

Parameters

`client` OpcClient

The OpcClient to use to call the method.

`fileNodeId` OpcNodeId

The OpcNodeId of the file node its 'Write' method node is to be called.

`fileHandle` SafeOpcFileHandle

The SafeOpcFileHandle acquired by one of the SecureOpen methods.

`data` Byte[]

The data to write to the file at the current position.

`options` OpcCallMethodOptions

The OpcCallMethodOptions to apply before and after calling the method node to write data to a file.

Exceptions

ArgumentNullException

The `client`, `fileNodeId`, `fileHandle` or `data` is a null reference (Nothing in Visual Basic).

InvalidOperationException

The file node does not exist.

NotSupportedException

The 'Write' method node does not exist or the method node signature does not define a valid method node to write data to a file.

OpcException

The call of the method node failed (see exception details for more information).

Remarks

It is server-dependent whether the written data is persistently stored if the session is ended without calling the Close method.

SecureWrite(OpcFileNodeContext, SafeOpcFileHandle, Byte[])

Writes the `data` to the file maintained by the file node referenced by the `context` specified.

C#

```
public static void SecureWrite(OpcFileNodeContext context, SafeOpcFileHandle fileHandle,
byte[] data)
```

Parameters

`context` OpcFileNodeContext

The OpcFileNodeContext to use to access the file node.

`fileHandle` SafeOpcFileHandle

The SafeOpcFileHandle acquired by one of the SecureOpen methods.

`data` Byte[]

The data to write to the file at the current position.

Exceptions

ArgumentNullException

The `context`, `fileHandle` or `data` is a null reference (Nothing in Visual Basic).

InvalidOperationException

The file node does not exist.

NotSupportedException

The 'Write' method node does not exist or the method node signature does not define a valid method node to write data to a file.

OpcException

The call of the method node failed (see exception details for more information).

Remarks

It is server-dependent whether the written data is persistently stored if the session is ended without calling the Close method.

SecureWrite(OpcFileNodeContext, SafeOpcFileHandle, Byte[], OpcCallMethodOptions)

Writes the `data` to the file maintained by the file node referenced by the `context` specified.

C#

```
public static void SecureWrite(OpcFileNodeContext context, SafeOpcFileHandle fileHandle,
byte[] data, OpcCallMethodOptions options)
```

Parameters

`context` OpcFileNodeContext

The OpcFileNodeContext to use to access the file node.

`fileHandle` SafeOpcFileHandle

The SafeOpcFileHandle acquired by one of the SecureOpen methods.

`data` Byte[]

The data to write to the file at the current position.

`options` OpcCallMethodOptions

The OpcCallMethodOptions to apply before and after calling the method node to write data to a file.

Exceptions

ArgumentNullException

The `context`, `fileHandle` or `data` is a null reference (Nothing in Visual Basic).

InvalidOperationException

The file node does not exist.

NotSupportedException

The 'Write' method node does not exist or the method node signature does not define a valid method node to write data to a file.

OpcException

The call of the method node failed (see exception details for more information).

Remarks

It is server-dependent whether the written data is persistently stored if the session is ended without calling the Close method.

SecureWrite(SafeOpcFileHandle, Byte[])

Writes the `data` to the file identified by the `fileHandle` specified.

C#

```
public static void SecureWrite(SafeOpcFileHandle fileHandle, byte[] data)
```

Parameters

`fileHandle` SafeOpcFileHandle

The SafeOpcFileHandle acquired by one of the SecureOpen methods.

`data` Byte[]

The data to write to the file at the current position.

Exceptions

ArgumentNullException

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

InvalidOperationException

The file node does not exist.

NotSupportedException

The 'Write' method node does not exist or the method node signature does not define a valid method node to write data to a file.

OpcException

The call of the method node failed (see exception details for more information).

Remarks

It is server-dependent whether the written data is persistently stored if the session is ended without calling the Close method.

SecureWrite(SafeOpcFileHandle, Byte[], OpcCallMethodOptions)

Writes the `data` to the file identified by the `fileHandle` specified.

C#

```
public static void SecureWrite(SafeOpcFileHandle fileHandle, byte[] data,
OpcCallMethodOptions options)
```

Parameters

`fileHandle` SafeOpcFileHandle

The SafeOpcFileHandle acquired by one of the SecureOpen methods.

`data` Byte[]

The data to write to the file at the current position.

`options` OpcCallMethodOptions

The OpcCallMethodOptions to apply before and after calling the method node to write data to a file.

Exceptions

ArgumentNullException

The `fileHandle` or `data` is a null reference (Nothing in Visual Basic).

InvalidOperationException

The file node does not exist.

NotSupportedException

OpcException

The call of the method node failed (see exception details for more information).

Remarks

It is server-dependent whether the written data is persistently stored if the session is ended without calling the Close method.

SetPosition(OpcClient, OpcNodeId, UInt32, UInt64)

Updates the position in the file maintained by the file node referenced by the `fileNodeId` specified using the further specified OpcClient using `position`.

C#

```
[CLSCompliant(false)]
public static void SetPosition(OpcClient client, OpcNodeId fileNodeId, uint fileHandle,
ulong position)
```

Parameters

`client` OpcClient

The OpcClient to use to call the method.

`fileNodeId` OpcNodeId

The OpcNodeId of the file node its 'SetPosition' method node is to be called.

`fileHandle` UInt32

The numeric file handle acquired by one of the Open methods.

`position` UInt64

The new position in the file. If `position` is greater than the file size the position is set to the end of the file.

Exceptions

ArgumentNullException

The `client` or `fileNodeId` is a null reference (Nothing in Visual Basic).

InvalidOperationException

The file node does not exist.

NotSupportedException

The 'SetPosition' method node does not exist or the method node signature does not define a valid method node to update the position in a file.

OpcException

The call of the method node failed (see exception details for more information).

SetPosition(OpcClient, OpcNodeId, UInt32, UInt64, OpcCallMethodOptions)

Updates the position in the file maintained by the file node referenced by the `fileNodeId` specified using the further specified OpcClient using `position`.

C#

```
[CLSCompliant(false)]
public static void SetPosition(OpcClient client, OpcNodeId fileNodeId, uint fileHandle,
ulong position, OpcCallMethodOptions options)
```

Parameters

`client` OpcClient

The OpcClient to use to call the method.

`fileNodeId` OpcNodeId

The OpcNodeId of the file node its 'SetPosition' method node is to be called.

`fileHandle` UInt32

The numeric file handle acquired by one of the Open methods.

`position` UInt64

The new position in the file. If `position` is greater than the file size the position is set to the end of the file.

`options` OpcCallMethodOptions

The OpcCallMethodOptions to apply before and after calling the method node to update the position in a file.

Exceptions

ArgumentNullException

The `client` or `fileNodeId` is a null reference (Nothing in Visual Basic).

InvalidOperationException

The file node does not exist.

NotSupportedException

The 'SetPosition' method node does not exist or the method node signature does not define a valid method node to update the position in a file.

OpcException

The call of the method node failed (see exception details for more information).

SetPosition(OpcFileNodeContext, UInt32, UInt64)

Updates the position in the file maintained by the file node referenced by the `context` specified using `position`.

C#

```
[CLSCompliant(false)]
public static void SetPosition(OpcFileNodeContext context, uint fileHandle, ulong position)
```

Parameters

`context` OpcFileNodeContext

The OpcFileNodeContext to use to access the file node.

`fileHandle` UInt32

The numeric file handle acquired by one of the Open methods.

`position` UInt64

The new position in the file. If `position` is greater than the file size the position is set to the end of the file.

Exceptions

ArgumentNullException

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

InvalidOperationException

The file node does not exist.

NotSupportedException

The 'SetPosition' method node does not exist or the method node signature does not define a valid method node to update the position in a file.

OpcException

The call of the method node failed (see exception details for more information).

SetPosition(OpcFileNodeContext, UInt32, UInt64, OpcCallMethodOptions)

Updates the position in the file maintained by the file node referenced by the `context` specified using `position`.

C#

```
[CLSCompliant(false)]
public static void SetPosition(OpcFileNodeContext context, uint fileHandle, ulong position,
OpcCallMethodOptions options)
```

Parameters

`context` OpcFileNodeContext

The OpcFileNodeContext to use to access the file node.

`fileHandle` UInt32

The numeric file handle acquired by one of the Open methods.

`position` UInt64

The new position in the file. If `position` is greater than the file size the position is set to the end of the file.

`options` OpcCallMethodOptions

The OpcCallMethodOptions to apply before and after calling the method node to update the position in a file.

Exceptions

ArgumentNullException

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

InvalidOperationException

The file node does not exist.

NotSupportedException

The 'SetPosition' method node does not exist or the method node signature does not define a valid method node to update the position in a file.

OpcException

The call of the method node failed (see exception details for more information).

Write(OpcClient, OpcNodeId, UInt32, Byte[])

Writes the `data` to the file maintained by the file node referenced by the `fileNodeId` specified using the further specified OpcClient.

C#

```
[CLSCompliant(false)]
public static void Write(OpcClient client, OpcNodeId fileNodeId, uint fileHandle, byte[]
data)
```

Parameters

`client` OpcClient

The OpcClient to use to call the method.

`fileNodeId` OpcNodeId

The OpcNodeId of the file node its 'Write' method node is to be called.

`fileHandle` UInt32

The numeric file handle acquired by one of the Open methods.

`data` Byte[]

The data to write to the file at the current position.

Exceptions

ArgumentNullException

The `client`, `fileNodeId` or `data` is a null reference (Nothing in Visual Basic).

InvalidOperationException

The file node does not exist.

NotSupportedException

The 'Write' method node does not exist or the method node signature does not define a valid method node to write data to a file.

OpcException

The call of the method node failed (see exception details for more information).

Remarks

It is server-dependent whether the written data is persistently stored if the session is ended without calling the Close method.

Write(OpcClient, OpcNodeId, UInt32, Byte[], OpcCallMethodOptions)

Writes the `data` to the file maintained by the file node referenced by the `fileNodeId` specified using the further specified OpcClient.

C#

```
[CLSCompliant(false)]
public static void Write(OpcClient client, OpcNodeId fileNodeId, uint fileHandle, byte[]
data, OpcCallMethodOptions options)
```

Parameters

`client` OpcClient

The OpcClient to use to call the method.

`fileNodeId` OpcNodeId

The OpcNodeId of the file node its 'Write' method node is to be called.

`fileHandle` UInt32

The numeric file handle acquired by one of the Open methods.

`data` Byte[]

The data to write to the file at the current position.

`options` OpcCallMethodOptions

The OpcCallMethodOptions to apply before and after calling the method node to write data to a file.

Exceptions

ArgumentNullException

The `client`, `fileNodeId` or `data` is a null reference (Nothing in Visual Basic).

InvalidOperationException

The file node does not exist.

NotSupportedException

The 'Write' method node does not exist or the method node signature does not define a valid method node to write data to a file.

OpcException

The call of the method node failed (see exception details for more information).

Remarks

It is server-dependent whether the written data is persistently stored if the session is ended without calling the Close method.

Write(OpcFileNodeContext, UInt32, Byte[])

Writes the `data` to the file maintained by the file node referenced by the `context` specified.

C#

```
[CLSCompliant(false)]
public static void Write(OpcFileNodeContext context, uint fileHandle, byte[] data)
```

Parameters

`context` OpcFileNodeContext

The OpcFileNodeContext to use to access the file node.

`fileHandle` UInt32

The numeric file handle acquired by one of the Open methods.

`data` Byte[]

The data to write to the file at the current position.

Exceptions

ArgumentNullException

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

InvalidOperationException

The file node does not exist.

NotSupportedException

The 'Write' method node does not exist or the method node signature does not define a valid method node to write data to a file.

OpcException

The call of the method node failed (see exception details for more information).

Remarks

It is server-dependent whether the written data is persistently stored if the session is ended without calling the Close method.

Write(OpcFileNodeContext, UInt32, Byte[], OpcCallMethodOptions)

Writes the `data` to the file maintained by the file node referenced by the `context` specified.

C#

```
[CLSCompliant(false)]
public static void Write(OpcFileNodeContext context, uint fileHandle, byte[] data,
OpcCallMethodOptions options)
```

Parameters

`context` OpcFileNodeContext

The OpcFileNodeContext to use to access the file node.

`fileHandle` UInt32

The numeric file handle acquired by one of the Open methods.

`data` Byte[]

The data to write to the file at the current position.

`options` OpcCallMethodOptions

The OpcCallMethodOptions to apply before and after calling the method node to write data to a file.

Exceptions

ArgumentNullException

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

InvalidOperationException

The file node does not exist.

NotSupportedException

The 'Write' method node does not exist or the method node signature does not define a valid method node to write data to a file.

OpcException

The call of the method node failed (see exception details for more information).

Remarks

It is server-dependent whether the written data is persistently stored if the session is ended without calling the Close method.

Table of Contents

Methods 1
Close(OpcClient, OpcNodeId, UInt32) 1
Close(OpcClient, OpcNodeId, UInt32, OpcCallMethodOptions) 2
Close(OpcFileNodeContext, UInt32) 2
Close(OpcFileNodeContext, UInt32, OpcCallMethodOptions) 3
GetPosition(OpcClient, OpcNodeId, UInt32) 4
GetPosition(OpcClient, OpcNodeId, UInt32, OpcCallMethodOptions) 5
GetPosition(OpcFileNodeContext, UInt32) 6
GetPosition(OpcFileNodeContext, UInt32, OpcCallMethodOptions) 7
IsFileNode(OpcClient, OpcNodeId) 8
IsFileNode(OpcFileNodeContext) 8
Open(OpcClient, OpcNodeId, OpcFileMode) 9
Open(OpcClient, OpcNodeId, OpcFileMode, OpcCallMethodOptions) 10
Open(OpcFileNodeContext, OpcFileMode) 11
Open(OpcFileNodeContext, OpcFileMode, OpcCallMethodOptions) 11
Read(OpcClient, OpcNodeId, UInt32, Int32) 12
Read(OpcClient, OpcNodeId, UInt32, Int32, OpcCallMethodOptions) 13
Read(OpcFileNodeContext, UInt32, Int32) 14
Read(OpcFileNodeContext, UInt32, Int32, OpcCallMethodOptions) 15
SecureClose(OpcClient, OpcNodeId, SafeOpcFileHandle) 16
SecureClose(OpcClient, OpcNodeId, SafeOpcFileHandle, OpcCallMethodOptions) 17
SecureClose(OpcFileNodeContext, SafeOpcFileHandle) 18
SecureClose(OpcFileNodeContext, SafeOpcFileHandle, OpcCallMethodOptions) 19
SecureClose(SafeOpcFileHandle) 20
SecureClose(SafeOpcFileHandle, OpcCallMethodOptions) 20
SecureGetPosition(OpcClient, OpcNodeId, SafeOpcFileHandle) 21
SecureGetPosition(OpcClient, OpcNodeId, SafeOpcFileHandle, OpcCallMethodOptions) 22
SecureGetPosition(OpcFileNodeContext, SafeOpcFileHandle) 23
SecureGetPosition(OpcFileNodeContext, SafeOpcFileHandle, OpcCallMethodOptions) 24
SecureGetPosition(SafeOpcFileHandle) 24
SecureGetPosition(SafeOpcFileHandle, OpcCallMethodOptions) 25
SecureOpen(OpcClient, OpcNodeId, OpcFileMode) 26
SecureOpen(OpcClient, OpcNodeId, OpcFileMode, OpcCallMethodOptions) 27
SecureOpen(OpcFileNodeContext, OpcFileMode) 28
SecureOpen(OpcFileNodeContext, OpcFileMode, OpcCallMethodOptions) 29
SecureRead(OpcClient, OpcNodeId, SafeOpcFileHandle, Int32) 30
SecureRead(OpcClient, OpcNodeId, SafeOpcFileHandle, Int32, OpcCallMethodOptions) 31
SecureRead(OpcFileNodeContext, SafeOpcFileHandle, Int32) 32
SecureRead(OpcFileNodeContext, SafeOpcFileHandle, Int32, OpcCallMethodOptions) 33
SecureRead(SafeOpcFileHandle, Int32) 34
SecureRead(SafeOpcFileHandle, Int32, OpcCallMethodOptions) 35
SecureSetPosition(OpcClient, OpcNodeId, SafeOpcFileHandle, Int64) 36
SecureSetPosition(OpcClient, OpcNodeId, SafeOpcFileHandle, Int64, OpcCallMethodOptions) 36
SecureSetPosition(OpcFileNodeContext, SafeOpcFileHandle, Int64) 37
SecureSetPosition(OpcFileNodeContext, SafeOpcFileHandle, Int64, OpcCallMethodOptions) 38
SecureSetPosition(SafeOpcFileHandle, Int64) 39
SecureSetPosition(SafeOpcFileHandle, Int64, OpcCallMethodOptions) 40
SecureWrite(OpcClient, OpcNodeId, SafeOpcFileHandle, Byte[]) 40
SecureWrite(OpcClient, OpcNodeId, SafeOpcFileHandle, Byte[], OpcCallMethodOptions) 41