

OPC Client (RLOPCC)



OPC Client (RLOPCC)

OPC Client (RLOPCC)

Table of Contents

INSTALLATION	3
LICENSING	3
DOCUMENTATION	4
FUNCTIONS.....	4
<i>Add a group to an OPC Server.....</i>	4
<i>Add an item to the group with a given handle</i>	5
<i>Add items to the group with a given handle.....</i>	6
<i>Asynchronously read a value of the tag with a given handle</i>	7
<i>Asynchronously read the value of multiple tags.....</i>	8
<i>Asynchronously write a value to the tag with a given handle</i>	9
<i>Asynchronously write values to multiple tags</i>	10
<i>Apply clamping to specified item.....</i>	11
<i>Apply clamping to specified items</i>	12
<i>Connect to an OPC server with a given ProgID.....</i>	13
<i>Disconnect from an OPC server with a given handle</i>	14
<i>Disconnect from all OPC servers</i>	15
<i>Refresh values of the items of the group with a given handle.....</i>	16
<i>Remove the group with a given handle</i>	17
<i>Remove item</i>	18
<i>Remove items.....</i>	19
<i>Set precision of specified item.....</i>	20
<i>Set precision of specified items.....</i>	21
<i>Apply linear scaling to specified item.....</i>	22
<i>Apply linear scaling to specified items.....</i>	23
<i>Synchronously read a value of the tag with a given handle</i>	24
<i>Synchronously read the value of multiple tags.....</i>	25
<i>Synchronously write a value to the tag with a given handle</i>	27
<i>Synchronously write values to multiple tags.....</i>	28
PROPERTIES.....	29
<i>Make the group with a given handle active/inactive</i>	29
<i>Enable/disable updates from the group with a given handle.....</i>	30
<i>Set update rate for the group with a given handle.....</i>	31
EVENTS.....	32
<i>Fired when the value(s) of added item(s) change(s).....</i>	32
<i>Fired when the asynchronous read is completed.....</i>	33
<i>Fired when the asynchronous write is completed.....</i>	34

OPC Client (RLOPCC)

Installation

ReLab Software Development toolkit is part of ReLab OPC Suite. In order to use the toolkit, you need to download ReLab OPC Suite and install it on your computer. The toolkit is implemented in RLOPCC.dll. During the Installation process the RLOPCC.dll will be properly registered in your system.

Licensing

In order to use the toolkit, you need to receive from ReLab the license file and the Licensing code. The code has to be incorporated into the project and the file has to be copied into ReLab's OPC products "bin" directory.

OPC Client (RLOPCC)

Documentation

The documentation includes description, input parameters and return values of RLOPCC.dll functions, properties and events.

Functions

Add a group to an OPC Server

Call AddGroup function to add a group to the specified OPC Server

VB

```
AddGroup( hServer As Long, pwszGroupName As String, bActive As Long, dwUpdateRate As Long ) As Long
```

C++

```
HRESULT AddGroup( long hServer, BSTR pwszGroupName, BOOL bActive, long dwUpdateRate, long *phGroup );
```

Parameters:

- *hServer* [in] – A handle to the OPC Server, previously obtained with **Connect** method.
- *pwszGroupName* [in] – The name of the OPC group to be created.
- *bActive* [in] – The state of the OPC group to be created.
- *dwUpdateRate* [in] – The desired update rate (in milliseconds) of the OPC group to be created.
- *phGroup* [out, retval] – A pointer to a variable to receive the OPC group handle.

Return values:

Result	Description
S_OK	The operation succeeded
E_FAIL	The operation failed
E_OUTOFMEMORY	Not enough memory
E_INVALIDARG	One or more arguments of the method are invalid

Comments:

If succeeded the method returns a handle to a new OPC group to be used in subsequent operations.

OPC Client (RLOPCC)

Add an item to the group with a given handle

Call **AddItem** function to add a single item to the specified OPC group

VB

```
AddItem( hGroup As Long, phItem As Long, pwszItemID As String, fDeadband As Single ) As Long
```

C++

```
HRESULT AddItem( long hGroup, long *phItem, BSTR pwszItemID, float fDeadband, long *pError );
```

Parameters:

- *hGroup* [in] – A handle to the OPC Group, previously created with **AddGroup** method.
- *phItem* [out] – A pointer to a variable receiving a handle to the new item.
- *pwszItemID* [in] – The identifier of the OPC item to be created.
- *fDeadband* [in] – The deadband value for the OPC item to be created.
- *pError* [out, retval] – A pointer to a variable to receive the result of the operation.

Possible results:

Result	Description
S_OK	The operation succeeded
E_FAIL	The operation failed
E_OUTOFMEMORY	Not enough memory

Return values:

Result	Description
S_OK	The operation succeeded
E_FAIL	The operation failed
E_OUTOFMEMORY	Not enough memory
E_INVALIDARG	One or more arguments of the method are invalid

Comments:

If succeeded the method returns a handle to a new OPC item to be used in subsequent operations. AddItems method is much more effective when client adds several OPC items.

OPC Client (RLOPCC)

Add items to the group with a given handle

Call **AddItems** function to add several items to the specified OPC group

VB

```
AddItems( hGroup As Long, nCount As Long, phItem() As Long, ppwszItemID() As String, pfDeadband() As Single, pError()  
As Long )
```

C++

```
HRESULT AddItems( long hGroup, SAFEARRAY(long) *phItem, SAFEARRAY(BSTR) *ppwszItemID, SAFEARRAY(float)  
*pfDeadband, SAFEARRAY(long) *pError );
```

Parameters:

- *hGroup* [in] – A handle to the OPC Group, previously created with **AddGroup** method.
- *phItem* [out] – A pointer to an array of variables receiving handles of the new items.
- *ppwszItemID* [in] – The array of identifiers of the OPC items to be created.
- *pfDeadband* [in] – The array of deadband values for the OPC items to be created.
- *pError* [out, retval] – A pointer to an array of variables to receive the results of the operation.

Possible results:

Result	Description
S_OK	The operation succeeded
E_FAIL	The operation failed
E_OUTOFMEMORY	Not enough memory

Return values:

Result	Description
S_OK	The operation succeeded
S_FALSE	The operation failed for one or more items
E_FAIL	The operation failed
E_OUTOFMEMORY	Not enough memory
E_INVALIDARG	One or more arguments of the method are invalid

Comments:

If succeeded the method returns a handles of new OPC items to be used in subsequent operations. AddItems method is more convenient when client adds a few OPC items.

OPC Client (RLOPCC)

Asynchronously read a value of the tag with a given handle

Call **AsyncRead** to asynchronously read the value, quality and timestamp information of a single item of the OPC group.

VB

```
AsyncRead( hGroup As Long, nTransaction As Long, pnCancel As Long, hItem As Long )
```

C++

```
HRESULT AsyncRead( long hGroup, long nTransaction, long *pnCancel, long hItem );
```

Parameters:

- *hGroup* [in] – A handle to the group to be read from. The group handle is returned by **AddGroup** method.
- *nTransaction* [in] – The identifier (nonzero 32 bits numeric value) of the transaction.
- *pnCancel* [out] – The “cancel” identifier of the transaction.
- *hItem* [in] – The handle of the item to be read. The item handle is returned by either **AddItem** or **AddItems** methods.

C++ Return values:

Result	Description
S_OK	The read operation succeeded
S_FALSE	The read operation succeeded but there are one or more items could not be read. Refer to individual error returns in the ppErrors array for more information.
E_FAIL	The read operation failed
E_OUTOFMEMORY	Not enough memory
E_INVALIDARG	One or more arguments to the method were invalid

Comments:

The result of asynchronous read operation is delivered to a client by **ReadComplete** event.

OPC Client (RLOPCC)

Asynchronously read the value of multiple tags

Call **AsyncReadData** to asynchronously read the value, quality and timestamp information of one or more items of the OPC group.

VB

```
AsyncReadData( hGroup As Long, nTransaction As Long, pnCancel As Long, Count As Long, phItems() As Long )
```

C++

```
HRESULT AsyncReadData( long hGroup, long nTransaction, long *pnCancel, long Count, SAFEARRAY(long) *phItems );
```

Parameters:

- *hGroup* [in] – A handle to the group to be read from. The group handle is returned by **AddGroup** method.
- *nTransaction* [in] – The identifier (nonzero 32 bits numeric value) of the transaction.
- *pnCancel* [out] – The “cancel” identifier of the transaction.
- *Count* [in] – The number of items to be read.
- *phItems* [in] – A pointer to an array of item handles to be read. It may contain the multiple copies of the same handle. The item handle is returned by either **AddItem** or **AddItems** methods.

C++ Return values:

Result	Description
S_OK	The read operation succeeded
S_FALSE	The read operation succeeded but there are one or more items could not be read. Refer to individual error returns in the ppErrors array for more information.
E_FAIL	The read operation failed
E_OUTOFMEMORY	Not enough memory
E_INVALIDARG	One or more arguments to the method were invalid

Comments:

The result of asynchronous read operation is delivered to a client by **ReadComplete** event.

OPC Client (RLOPCC)

Asynchronously write a value to the tag with a given handle

Call **AsyncWrite** to asynchronously read the value, quality and timestamp information of a single item of the OPC group.

VB

```
AsyncWrite( hGroup As Long, nTransaction As Long, pnCancel As Long, hItem As Long, Value As Variant )
```

C++

```
HRESULT AsyncWrite( long hGroup, long nTransaction, long *pnCancel, long hItem, VARIANT Value );
```

Parameters:

- *hGroup* [in] – A handle to the group to be written to. The group handle is returned by **AddGroup** method.
- *nTransaction* [in] – The identifier (nonzero 32 bits numeric value) of the transaction.
- *pnCancel* [out] – The “cancel” identifier of the transaction.
- *hItem* [in] – The handle of the item to be written. The item handle is returned by either **AddItem** or **AddItems** methods.
- *Value* [in] – The value of the item.

C++ Return values:

Result	Description
S_OK	The read operation succeeded
S_FALSE	The read operation succeeded but there are one or more items could not be read. Refer to individual error returns in the ppErrors array for more information.
E_FAIL	The read operation failed
E_OUTOFMEMORY	Not enough memory
E_INVALIDARG	One or more arguments to the method were invalid

Comments:

The result of asynchronous write operation is delivered to a client by **WriteComplete** event.

OPC Client (RLOPCC)

Asynchronously write values to multiple tags

Call **AsyncWriteData** to asynchronously write the value of one or more items of the OPC group.

VB

AsyncWriteData(hGroup As Long, nTransaction As Long, pnCancel As Long, Count As Long, phItems() As Long, pValues() As Variant)

C++

```
HRESULT AsyncWriteData( long hGroup, long nTransaction, long *pnCancel, long Count, SAFEARRAY(long) *phItems, SAFEARRAY(VARIANT) *pValues );
```

Parameters:

- *hGroup* [in] – A handle to the group to be written to. The group handle is returned by **AddGroup** method.
- *nTransaction* [in] – The identifier (nonzero 32 bits numeric value) of the transaction.
- *pnCancel* [out] – The “cancel” identifier of the transaction.
- *Count* [in] – The number of items to be written.
- *phItems* [in] – A pointer to an array of item handles to be written. It may contain the multiple copies of the same handle. The item handle is returned by either **AddItem** or **AddItems** methods.
- *pValues* [in] – A pointer to the array of values.

C++ Return values:

Result	Description
S_OK	The read operation succeeded
E_FAIL	The read operation failed
E_OUTOFMEMORY	Not enough memory
E_INVALIDARG	One or more arguments to the method were invalid

Comments:

The result of asynchronous write operation is delivered to a client by **WriteComplete** event.

OPC Client (RLOPCC)

Apply clamping to specified item

Call **ClampItem** to set parameters necessary to clamp a single OPC item.

VB

ClampItem(hItem As Long, fMin As Single, fMax As Single) As Long

C++

```
HRESULT ClampItem( long hItem, float fMin, float fMax, long *pError );
```

Parameters:

- *hItem* [in] – A handle to the item to be clamped. The item handle is returned by **AddItem** or **AddItems** method.
- *fMin* [in] – The minimal value.
- *fMax* [in] – The maximal value.
- *pError* [out] – Pointer to a variable receiving a result of clamp operation.

C++ Return values:

Result	Description
S_OK	The read operation succeeded
E_FAIL	The read operation failed
E_OUTOFMEMORY	Not enough memory
E_INVALIDARG	One or more arguments to the method were invalid

OPC Client (RLOPCC)

Apply clamping to specified items

Call **ClampItems** to set parameters necessary to clamp one or more OPC items.

VB

ClampItems(nCount As Long, phItem() As Long, pfMins() As Single, pfMaxs() As Single, pError() As Long) As Long

C++

HRESULT ClampItems(long nCount, SAFEARRAY(long) *phItem, SAFEARRAY(float) *pfMins, SAFEARRAY(float) *pfMaxs, SAFEARRAY(long) *pError);

Parameters:

- *nCount* [in] – The number of items to be clamped.
- *phItem* [in] – A pointer to an array containing handles to the items to be clamped. The item handle is returned by **AddItem** or **AddItems** method.
- *pfMins* [in] – A pointer to an array containing minimal value.
- *pfMaxs* [in] – A pointer to an array containing expected scaled value.
- *pError* [out] – Pointer to an array of variables receiving a result of clamp operation.

C++ Return values:

Result	Description
S_OK	The read operation succeeded
E_FAIL	The read operation failed
E_OUTOFMEMORY	Not enough memory
E_INVALIDARG	One or more arguments to the method were invalid

OPC Client (RLOPCC)

Connect to an OPC server with a given ProgID

Call **Connect** method to establish a connection with an instance of the specified OPC Server on the specified host

VB

```
Connect( pwszNode As String, pwszProgID As String ) As Long
```

C++

```
HRESULT Connect( BSTR pwszNode, BSTR pwszProgID, long *phServer );
```

Parameters:

- *pwszNode* [in] – The name of a host where an OPC Server is deployed
- *pwszProgID* [in] – The ProgID of an OPC Server.
- *phServer* [out, retval] – A pointer to a variable to receive the OPC Server handle.

Return values:

Result	Description
S_OK	The operation succeeded
E_FAIL	The operation failed
E_OUTOFMEMORY	Not enough memory
E_INVALIDARG	One or more arguments of the method are invalid

Comments:

If succeeded the method returns a handle of an OPC Server to be used in subsequent operations.

OPC Client (RLOPCC)

Disconnect from an OPC server with a given handle

Call **Disconnect** to close the connection with the OPC Server specified by its handle.

VB

Disconnect (hServer As Long)

C++

HRESULT Disconnect(long hServer);

Parameters:

- *hServer* [in] – The handle of the OPC Server previously obtained with **Connect** method.

Return values:

Result	Description
S_OK	The operation succeeded

Comments:

The method closes existing connection and frees the memory associated with this connection.

OPC Client (RLOPCC)

Disconnect from all OPC servers

Call **DisconnectAll** to close all previously established connections.

VB

```
DisconnectAll()
```

C++

```
HRESULT DisconnectAll();
```

Parameters:

- None

Return values:

Result	Description
S_OK	The operation succeeded

Comments:

The method closes all existing connections and frees the memory associated with them.

OPC Client (RLOPCC)

Refresh values of the items of the group with a given handle

Call **Refresh** method to generate DataChange callback containing information about all OPC items containing in the OPC group specified by its handle.

VB

```
Refresh( hGroup As Long )
```

C++

```
HRESULT Refresh( long hGroup );
```

Parameters:

- *hGroup* [in] – The handle of the OPC group previously obtained with **AddGroup** method.

Return values:

Result	Description
S_OK	The operation succeeded
E_FAIL	The operation failed

Comments:

The method **IOPCAsyncIO2::Refresh2** is used to trigger **DataChange** event. Usually this method is used when complete OPC configuration is created on client side.

OPC Client (RLOPCC)

Remove the group with a given handle

Call **RemoveGroup** to remove the OPC group specified by its handle.

VB

```
RemoveGroup( hGroup As Long )
```

C++

```
HRESULT RemoveGroup( long hGroup );
```

Parameters:

- *hGroup* [in] – The handle of the OPC group previously obtained with **AddGroup** method.

Return values:

Result	Description
S_OK	The operation succeeded

Comments:

The method removes existing OPC group and frees the memory associated with it.

OPC Client (RLOPCC)

Remove item

Call **RemoveItem** function to remove a single item from the specified OPC group

VB

RemoveItem(hGroup As Long, hItem As Long) As Long

C++

```
HRESULT RemoveItem( long hGroup, long hItem, long *pError );
```

Parameters:

- *hGroup* [in] – A handle to the OPC Group, previously created with **AddGroup** method.
- *hItem* [in] – The handle to the item to be removed.
- *pError* [out, retval] – A pointer to a variable to receive the result of the operation.

Return values:

Result	Description
S_OK	The operation succeeded
E_FAIL	The operation failed
E_OUTOFMEMORY	Not enough memory
E_INVALIDARG	One or more arguments of the method are invalid

OPC Client (RLOPCC)

Remove items

Call **RemoveItems** function to remove one or more items from the specified OPC group

VB

```
RemoveItems( hGroup As Long, nCount As Long, phItem() As Long, pError() As Long )
```

C++

```
HRESULT RemoveItems( long hGroup, long nCount, SAFEARRAY(long) *phItem, SAFEARRAY(long) *pError );
```

Parameters:

- *hGroup* [in] – A handle to the OPC Group, previously created with **AddGroup** method.
- *nCount* [in] – The number of items to be removed.
- *phItem* [in] – A pointer to an array of the handles to the items to be removed.
- *pError* [out, retval] – A pointer to an array of variables to receive the results of the operation.

Return values:

Result	Description
S_OK	The operation succeeded
E_FAIL	The operation failed
E_OUTOFMEMORY	Not enough memory
E_INVALIDARG	One or more arguments of the method are invalid

OPC Client (RLOPCC)

Set precision of specified item

Call **RoundItem** to set parameters necessary to round a single OPC item.

VB

RoundItem(hItem As Long, nDigits As Long) As Long

C++

```
HRESULT RoundItem( long hItem, int nDigits, long *pError );
```

Parameters:

- *hItem* [in] – A handle to the item to be rounded. The item handle is returned by **AddItem** or **AddItems** method.
- *nDigits* [in] – The maximal number of digits after decimal.
- *pError* [out] – Pointer to a variable receiving a result of round operation.

C++ Return values:

Result	Description
S_OK	The read operation succeeded
E_FAIL	The read operation failed
E_OUTOFMEMORY	Not enough memory
E_INVALIDARG	One or more arguments to the method were invalid

Comments:

The round operation can be applied to float or double types of items only.

OPC Client (RLOPCC)

Set precision of specified items

Call **RoundItems** to set parameters necessary to round one or more OPC items.

VB

```
RoundItems( nCount As Long, phItem() As Long, pnDigits() As Long, pError() As Long )
```

C++

```
HRESULT RoundItems( long nCount, SAFEARRAY(long) *phItem, SAFEARRAY(int) *pnDigits, SAFEARRAY(long) *pError );
```

Parameters:

- *nCount* [in] – The number of items to be rounded.
- *phItem* [in] – A pointer to an array of handles to the items to be rounded. The item handle is returned by **AddItem** or **AddItems** method.
- *pnDigits* [in] – A pointer to an array of maximal numbers of digits after decimal.
- *pError* [out] – A pointer to an array of variables receiving results of round operation.

C++ Return values:

Result	Description
S_OK	The read operation succeeded
E_FAIL	The read operation failed
E_OUTOFMEMORY	Not enough memory
E_INVALIDARG	One or more arguments to the method were invalid

Comments:

The round operation can be applied to float or double types of items only.

OPC Client (RLOPCC)

Apply linear scaling to specified item

Call **ScaleItem** to set parameters necessary to scale a single OPC item.

VB

ScaleItem(hItem As Long, fRawMin As Single, fRawMax As Single, fMin As Single, fMax As Single) As Long

C++

HRESULT ScaleItem(long hItem, float fRawMin, float fRawMax, float fMin, float fMax, long *pError);

Parameters:

- *hItem* [in] – A handle to the item to be scaled. The item handle is returned by **AddItem** or **AddItems** method.
- *fRawMin* [in] – The expected minimal raw value.
- *fRawMax* [in] – The expected maximal raw value.
- *fMin* [in] – The minimal scaled value.
- *fMax* [in] – The maximal scaled value.
- *pError* [out] – Pointer to a variable receiving a result of scale operation.

C++ Return values:

Result	Description
S_OK	The read operation succeeded
E_FAIL	The read operation failed
E_OUTOFMEMORY	Not enough memory
E_INVALIDARG	One or more arguments to the method were invalid

Comments:

The raw and scaled values are used to calculate scaling coefficients, thus a value coming from an OPC Server can be less than *fMin* or greater than *fMax*.

OPC Client (RLOPCC)

Apply linear scaling to specified items

Call **ScaleItems** to set parameters necessary to scale one or more OPC items.

VB

ScaleItems(nCount As Long, phItem() As Long, pfRawMins() As Single, pfRawMaxs() As Single, pfMins() As Single, pfMaxs() As Single, pError() AsLong) As Long

C++

HRESULT ScaleItems(long nCount, SAFEARRAY(long) *phItem, SAFEARRAY(float) *pfRawMins, SAFEARRAY(float) *pfRawMaxs, SAFEARRAY(float) *pfMins, SAFEARRAY(float) *pfMaxs, SAFEARRAY(long) *pError);

Parameters:

- *nCount* [in] – The number of items to be scaled.
- *phItem* [in] – A pointer to an array containing handles to the items to be scaled. The item handle is returned by **AddItem** or **AddItems** method.
- *pfRawMins* [in] – A pointer to an array containing expected minimal raw values.
- *pfRawMaxs* [in] – A pointer to an array containing expected maximal raw value.
- *pfMins* [in] – A pointer to an array containing minimal scaled value.
- *pfMaxs* [in] – A pointer to an array containing expected maximal scaled value.
- *pError* [out] – Pointer to an array of variables receiving a result of scale operation.

C++ Return values:

Result	Description
S_OK	The read operation succeeded
E_FAIL	The read operation failed
E_OUTOFMEMORY	Not enough memory
E_INVALIDARG	One or more arguments to the method were invalid

Comments:

The raw and scaled values are used to calculate scaling coefficients, thus a value coming from an OPC Server can be less than fMin or greater than fMax.

OPC Client (RLOPCC)

Synchronously read a value of the tag with a given handle

Call **SyncRead** to synchronously read the value, quality and timestamp information of single item of the OPC group.

VB

```
SyncRead( hGroup As Long, bDevice As Long, hItem As Long, pValue As Variant, pQuality As Long, pTimeStamp As Date )
```

C++

```
HRESULT SyncRead( long hGroup, BOOL bDevice, long hItem, VARIANT *pValue, long *pQuality, DATE *pTimeStamp );
```

Parameters:

- *hGroup* [in] – A handle to the group to be read from. The group handle is returned by **AddGroup** method.
- *bDevice* [in] – A selector of the data source. If this parameter is TRUE, the data will be read from device(s) connected to the OPC Server. If this parameter is FALSE, the data will be read from the OPC Server cache.
- *hItem* [in] – The handle to the item to be read. The item handle is returned by either **AddItem** or **AddItems** methods.
- *pValue* [out] – A pointer to a variant receiving value of the item.
- *pQuality* [out] – A pointer to a variable receiving quality of the item.
- *pTimeStamp* [out] – A pointer to a variable receiving timestamp of the item.

C++ Return values:

Result	Description
S_OK	The read operation succeeded
E_FAIL	The read operation failed
E_OUTOFMEMORY	Not enough memory
E_INVALIDARG	One or more arguments to the method were invalid

Comments:

Usually a read from an OPC Server cache will be completed very quickly (within milliseconds). A read from a device connected to an OPC Server may take a very long time (many seconds or more). Depending on the details of the implementation (e.g. which threading model is used) the synchronous read operation may also prevent any other operations from being performed on an OPC Server by any other clients.

OPC Client (RLOPCC)

Synchronously read the value of multiple tags

Call **SyncReadData** to synchronously read the value, quality and timestamp information of one or more items of the OPC group.

VB

```
SyncReadData( hGroup As Long, bDevice As Long, nCount As Long, phItems() As Long, pValues() As Variant, pQualities()  
As Long, pErrors() As Long, pTimeStamps() As Date )
```

C++

```
HRESULT SyncReadData( long hGroup, BOOL bDevice, long nCount, SAFEARRAY(long) *phItems, SAFEARRAY(VARIANT)  
*pValues, SAFEARRAY(long) *pQualities, SAFEARRAY(long) *pErrors, SAFEARRAY(DATE) *pTimeStamps );
```

Parameters:

- *hGroup* [in] – A handle to the group to be read from. The group handle is returned by **AddGroup** method.
- *bDevice* [in] – A selector of the data source. If this parameter is TRUE, the data will be read from device(s) connected to the OPC Server. If this parameter is FALSE, the data will be read from the OPC Server cache.
- *Count* [in] – The number of items to be read.
- *phItems* [in] – A pointer to an array of item handles to be read. It may contain the multiple copies of the same handle. The item handle is returned by either **AddItem** or **AddItems** methods.
- *pValues* [out] – A pointer to an array of variants receiving values of the items.
- *pQualities* [out] – A pointer to an array of variables receiving qualities of the items.
- *pErrors* [out] – A pointer to an array of variables receiving read operation results for the items.
- *pTimeStamps* [out] – A pointer to an array of variables receiving timestamps of the items.

Possible read results:

Result	Description
S_OK	The item is successfully read
E_FAIL	The read operation fails for the item
OPC_E_BADRIGHTS	The item is not readable
OPC_E_INVALIDHANDLE	The passed item handle is not valid
OPC_E_UNKNOWNITEMID	The item is no longer available in the OPC Server address space
S_xxx E_xxx	The OPC Server vendor specific results

C++ Return values:

Result	Description
S_OK	The read operation succeeded
S_FALSE	The read operation succeeded but there are one or more items that could not be read. Refer to individual error returns in the pErrors array for more information.
E_FAIL	The read operation failed
E_OUTOFMEMORY	Not enough memory
E_INVALIDARG	One or more arguments of the method are invalid

OPC Client (RLOPCC)

Comments:

Usually a read from an OPC Server cache will be completed very quickly (within milliseconds). A read from a device connected to an OPC Server may take a very long time (many seconds or more). Depending on the details of the implementation (e.g. which threading model is used) the synchronous read operation may also prevent any other operations from being performed on an OPC Server by any other clients.

OPC Client (RLOPCC)

Synchronously write a value to the tag with a given handle

Call **SyncWrite** to synchronously write the value of single item of the OPC group.

VB

```
SyncWrite( hGroup As Long, hItem As Long, Value As Variant )
```

C++

```
HRESULT SyncWrite( long hGroup, long hItem, VARIANT Value, long *pError );
```

Parameters:

- *hGroup* [in] – A handle to the group to be written to. The group handle is returned by **AddGroup** method.
- *hItem* [in] – The handle to the item to be read. The item handle is returned by either **AddItem** or **AddItems** methods.
- *Value* [in] – The variant containing value of the item.
- *pError* [out] – A pointer to a variable receiving result of the operation.

C++ Return values:

Result	Description
S_OK	The write operation succeeded
E_FAIL	The write operation failed
E_OUTOFMEMORY	Not enough memory
E_INVALIDARG	One or more arguments to the method were invalid

Comments:

A write to a device connected to an OPC Server may take a very long time (many seconds or more). Depending on the details of the implementation (e.g. which threading model is used) the synchronous write operation may also prevent any other operations from being performed on an OPC Server by any other clients.

OPC Client (RLOPCC)

Synchronously write values to multiple tags

Call **SyncWriteData** to synchronously write the value of one or more items of the OPC group.

VB

SyncWriteData(hGroup As Long, Count As Long, phItems() As Long, pValues() As Variant, pErrors() As Long)

C++

HRESULT SyncWriteData(long hGroup, long Count, SAFEARRAY(long) *phItems, SAFEARRAY(VARIANT) *pValues, SAFEARRAY(long) *pErrors);

Parameters:

- *hGroup* [in] – A handle to the group to be written to. The group handle is returned by **AddGroup** method.
- *Count* [in] – The number of items to be read.
- *phItems* [in] – A pointer to an array of item handles to be written. It may contain the multiple copies of the same handle. The item handle is returned by either **AddItem** or **AddItems** methods.
- *pValues* [in] – The pointer to the array of variants containing values of the items.
- *pErrors* [out] – A pointer to an array of variables receiving read operation results for the items.

Possible read results:

Result	Description
S_OK	The item is successfully written
E_FAIL	The write operation fails for the item
OPC_E_BADRIGHTS	The item is not writable
OPC_E_INVALIDHANDLE	The passed item handle is not valid
OPC_E_UNKNOWNITEMID	The item is no longer available in the OPC Server address space
S_xxx E_xxx	The OPC Server vendor specific results

C++ Return values:

Result	Description
S_OK	The write operation succeeded
S_FALSE	The write operation succeeded but there are one or more items that could not be written. Refer to individual error returns in the pErrors array for more information.
E_FAIL	The write operation failed
E_OUTOFMEMORY	Not enough memory
E_INVALIDARG	One or more arguments of the method are invalid

Comments:

A write to a device connected to an OPC Server may take a very long time (many seconds or more). Depending on the details of the implementation (e.g. which threading model is used) the synchronous write operation may also prevent any other operations from being performed on an OPC Server by any other clients.

OPC Client (RLOPCC)

Properties

Make the group with a given handle active/inactive

Use **Active** property to check current state of the OPC group and to make it active or inactive the OPC group.

VB

Active(hGroup As Long) As Long

C++

```
HRESULT put_Active( long hGroup, BOOL bActive );  
HRESULT get_Active( long hGroup, BOOL *pbActive );
```

Parameters:

- *hGroup* [in] – A handle to the OPC group to be inspected, activated or deactivated. The handle is returned by **AddGroup** method.
- *bActive* [in] – The new state of the OPC group. If this parameter is TRUE, the OPC Server hosting this group activates the OPC group, otherwise – deactivates it.
- **pbActive* [out, retval] – A pointer to a variable to receive the OPC group state.

Return values:

Result	Description
S_OK	The operation succeeded
E_INVALIDARG	One or more arguments of the method are invalid
E_FAIL	The operation failed

Comments:

This method behaves similar to Enable method (in particular inactive group will not raise **DataChange** event).

OPC Client (RLOPCC)

Enable/disable updates from the group with a given handle

Use **Enable** property to check current state of the OPC group and to enable or disable the OPC group.

VB

```
Enable( hGroup As Long ) As Long
```

C++

```
HRESULT put_Enable( long hGroup, BOOL bEnable );  
HRESULT get_Enable( long hGroup, BOOL *pbEnable );
```

Parameters:

- *hGroup* [in] – The handle of the OPC group to be inspected, enabled or disabled. The handle is returned by **AddGroup** method.
- *bEnable* [in] – The new state of the OPC group. If this parameter is TRUE, the OPC Server hosting this group will call **DataChange** event using the update rate previously set for this group. If this parameter is FALSE, the OPC Server will not raise **DataChange** event.
- **pbEnable* [out, retval] – A pointer to a variable to receive the OPC group state.

Return values:

Result	Description
S_OK	The operation succeeded
E_INVALIDARG	One or more arguments of the method are invalid
E_FAIL	The operation failed

Comments:

This method is used when DataCallback event has to be temporarily suspended. For instance, usual practice is disabling the callback when OPC items are being added to or removed from the OPC group.

OPC Client (RLOPCC)

Set update rate for the group with a given handle

Use **UpdateRate** property to get or put current value of update rate for the OPC group.

VB

UpdateRate(hGroup As Long) As Long

C++

```
HRESULT put_UpdateRate( long hGroup, long nRate );  
HRESULT get_UpdateRate( long hGroup, long *pnRate );
```

Parameters:

- *hGroup* [in] – The handle of the OPC group to be inspected or modified. The handle is returned by **AddGroup** method.
- *nRate* [in] – The desired update rate of the OPC group. The actual update rate depends on current load of the OPC Server hosting the group.
- **pnRate* [out, retval] – A pointer to a variable to receive the update rate of the OPC group.

Return values:

Result	Description
S_OK	The operation succeeded
E_INVALIDARG	One or more arguments of the method are invalid
E_FAIL	The operation failed

Comments:

The actual update rate depends on current load of the OPC Server hosting the group and therefore can be different from the one requested by a client.

OPC Client (RLOPCC)

Events

Fired when the value(s) of added item(s) change(s)

Event **DataChange** is raised as a result of previous call to Refresh method and periodically according to the OPC group update rate and content.

VB

DataChange(hGroup As Long, nCount As Long, phItems() As Long, pValues() As Variant, pQualities() As Long, pErrors() As Long, pTS() As Date)

C++

```
HRESULT DataChange( long hGroup, long nCount, SAFEARRAY(long) *phItems, SAFEARRAY(VARIANT) *pValues, SAFEARRAY(long) *pQualities, SAFEARRAY(long) *pErrors, SAFEARRAY(DATE) *pTS );
```

Parameters:

- *hGroup* [in] – A handle to the group generated the event.
- *Count* [in] – The number of items delivered by the event.
- *phItems* [in] – A pointer to an array of item handles. It may contain the multiple copies of the same handle.
- *pValues* [in] – A pointer to an array of variants containing the values of the items.
- *pQualities* [in] – A pointer to an array of variables containing the qualities of the items.
- *pErrors* [in] – A pointer to an array of variables containing the results for the items.
- *pTS* [in] – A pointer to an array of variables containing the timestamps of the items.

Possible results:

Result	Description
S_OK	The item is successfully accessed
E_FAIL	The item can not be reliably accessed
OPC_E_UNKNOWNTITEMID	The item is no longer available in the OPC Server address space
S_xxx E_xxx	The OPC Server vendor specific results

C++ Return values:

Result	Description
S_OK	The operation succeeded
S_FALSE	The operation succeeded but there are one or more items that could not be accessed. Refer to individual error returns in the pErrors array for more information.
E_FAIL	The operation failed
E_OUTOFMEMORY	Not enough memory
E_INVALIDARG	One or more arguments of the method are invalid

OPC Client (RLOPCC)

Fired when the asynchronous read is completed

Event **ReadComplete** is raised as a result of previous calls to **AsyncRead** or **AsyncReadData** methods.

VB

ReadComplete(nTransaction As Long, hGroup As Long, nCount As Long, phItems() As Long, pValues() As Variant, pQualities() As Long, pErrors() As Long, pTS() As Date)

C++

HRESULT ReadComplete(long nTransaction, long hGroup, long nCount, SAFEARRAY(long) *phItems, SAFEARRAY(VARIANT) *pValues, SAFEARRAY(long) *pQualities, SAFEARRAY(long) *pErrors, SAFEARRAY(DATE) *pTS);

Parameters:

- *nTransaction* [in] – The identifier of the read transaction.
- *hGroup* [in] – A handle to the group generated the event.
- *Count* [in] – The number of items delivered by the event.
- *phItems* [in] – A pointer to an array of item handles. It may contain the multiple copies of the same handle.
- *pValues* [in] – A pointer to an array of variants containing the values of the items.
- *pQualities* [in] – A pointer to an array of variables containing the qualities of the items.
- *pErrors* [in] – A pointer to an array of variables containing the results for the items.
- *pTS* [in] – A pointer to an array of variables containing the timestamps of the items.

Possible results:

Result	Description
S_OK	The item is successfully read
E_FAIL	The read operation fails for the item
OPC_E_UNKNOWNITEMID	The item is no longer available in the OPC Server address space
S_xxx E_xxx	The OPC Server vendor specific results

C++ Return values:

Result	Description
S_OK	The read operation succeeded
S_FALSE	The read operation succeeded but there are one or more items that could not be accessed. Refer to individual error returns in the pErrors array for more information.
E_FAIL	The read operation failed
E_OUTOFMEMORY	Not enough memory
E_INVALIDARG	One or more arguments of the method are invalid

OPC Client (RLOPCC)

Fired when the asynchronous write is completed

Event **WriteComplete** is raised as a result of previous calls to **AsyncWrite** or **AsyncWriteData** methods.

VB

WriteComplete(nTransaction As Long, hGroup As Long, nCount As Long, phItems() As Long, pErrors() As Long)

C++

HRESULT WriteComplete(long nTransaction, long hGroup, long nCount, SAFEARRAY(long) *phItems, SAFEARRAY(long) *pErrors);

Parameters:

- *nTransaction* [in] – The identifier of the write transaction.
- *hGroup* [in] – A handle to the group generated the event.
- *nCount* [in] – The number of items delivered by the event.
- *phItems* [in] – A pointer to an array of item handles. It may contain the multiple copies of the same handle.
- *pErrors* [in] – A pointer to an array of variables containing the results of the write transaction for the items.

Possible results:

Result	Description
S_OK	The item is successfully written
E_FAIL	The write operation fails for the item
OPC_E_UNKNOWNITEMID	The item is no longer available in the OPC Server address space
S_xxx E_xxx	The OPC Server vendor specific results

C++ Return values:

Result	Description
S_OK	The write operation succeeded
S_FALSE	The write operation succeeded but there are one or more items that could not be accessed. Refer to individual error returns in the pErrors array for more information.
E_FAIL	The read operation failed
E_OUTOFMEMORY	Not enough memory
E_INVALIDARG	One or more arguments of the method are invalid