GE VERNOVA

PROFICY®SOFTWARE & SERVICES

PROFICY iFIX HMI/SCADA

OSDK Type Library

GE VERNOVA

Proprietary Notice
The information contained in this publication is believed to be accurate and reliable. However, GE Vernova assumes no responsibilities for any errors, omissions or inaccuracies. Information contained in the publication is subject to change without notice.

Trademark Notices
"GE VERNOVA" is a registered trademark of GE Vernova. The terms "GE" and the GE Monogram are trademarks of the General Electric Company, and are used with permission.

Microsoft® is a registered trademark of Microsoft Corporation, in the United States and/or other countries.

All other trademarks are the property of their respective owners.

We want to hear from you. If you have any comments, questions, or suggestions about our documentation, send them to the following email address:
doc@ge.com

Table of Contents

OSDK Type Library Reference

The OSDK Type Library Reference is intended for programmers who are proficient in the Microsoft® Visual Basic® programming language.

The following sections provide more details on how to use the methods and properties associated with the OSDK Type Library.

- Method Summary
- Property Summary
- Obtaining an Interface Pointer to the Server Object
- OPC Client Only

Method Summary

The following list contains the OSDK Type Library methods that are available.

AddChannel

AddDataBlock

AddDevice

CopyDevice

DebugMessage

ErrorMessage

FileNew

FileOpen

FileSave

FileSaveAs

GetChannels

GetDevices

GetDatablocks

GetNameSpace

GetProperties

GetPropertyData

InfoMessage

IsBrowseSupported

Poll

Quit

ReadData

RemoveChannel

AddChannel Method

Adds a channel object to the server configuration, optionally setting channel properties.

Syntax

Handle = *object*.**AddChannel** *Properties PropertyData Errors*

The **AddChannel** method syntax has these parts:

Part	Description
Handle	A long integer value specifying the channel object handle assigned by the server.
object	An object expression that evaluates to the Driver object.
Properties	A string array specifying channel properties to set.
PropertyData	An array specifying corresponding channel property data.
Errors	A returned array specifying corresponding property setting errors.

Remarks

Any channel properties not specified will be set to a default value.

AddDatablock Method

Adds a datablock to the specified device's configuration, optionally setting datablock properties.

Syntax

Handle = *object*.**AddDataBlock** *DeviceHandleProperties PropertyData Errors*

The **AddDataBlock** method syntax has these parts:

Part	Description
Handle	A long integer value specifying the channel object handle assigned by the server.
object	An object expression that evaluates to the Driver object.
DeviceHandle	A long integer expression specifying the handle of the device of which to add the device.
Properties	A string array specifying channel properties to set.
PropertyData	An array specifying corresponding channel property data.
Errors	A returned array specifying corresponding property setting errors.

Remarks

Any datablock properties not specified will be set to a default value.

Add Device Method

Returns the path specification for the application's executable file. Read only.

Syntax

Handle = *object*.**AddDevice** *ChannelHandleProperties PropertyData Errors*

The **AddDevice** method syntax has these parts:

Part	Description
Handle	A long integer value specifying the channel object handle assigned by the server.
object	An object expression that evaluates to the Driver object.
ChannelHandle	A long integer expression specifying the handle of the channel to which to add the device.
Properties	A string array specifying channel properties to set.
PropertyData	An array specifying corresponding channel property data.
Errors	A returned array specifying corresponding property setting errors.

Remarks

Any device properties not specified will be set to a default value.

CopyDevice Method

Copies or moves the specified device to the specified channel's configuration.

Syntax

Error=*object*.**CopyDevice** *DeviceHandle*, *bstrName*, *ChannelHandlebMove*

The **CopyDevice** method syntax has these parts:

Part	Description
Error	A long integer value specifying success or failure.
object	An object expression that evaluates to the Driver object.
DeviceHandle	A long integer expression specifying the handle of the device to copy or move.
bstrName	A string expression that specifies the name of the device to copy or move.
ChannelHandle	A long integer value specifying the handle of the channel to add the device to.
bMove	A boolean expression specifying whether the device should be moved.

Settings

The settings for *bMove* are:

Setting	Description
True	If the device is to be moved.
False	If the Device is to be copied.

The settings for *Error* are:

Setting	Description
0	The device was copied or moved.
1	The device was not copied or moved; the method failed.

DebugMessage Method

Sends a debug message to the I/O Server event window.

Syntax

object.**DebugMessage** *Message*

The **DebugMessage** method syntax has these parts:

Part	Description
object	An object expression that evaluates to the Driver object.
Message	A string expression that specifies the text you want to display in the I/O Server event window as a debug message.

ErrorMessage Method

Sends an error message to the I/O Server event window.

Syntax

object.**ErrorMessage** *Message*

The **ErrorMessage** method syntax has these parts:

Part	Description
object	An object expression that evaluates to the Driver object.
Message	A string expression that specifies the text you want to display in the I/O Server event window as an error message.

FileNew Method

Creates an empty configuration.

Syntax

object.**FileNew**

The **FileNew** method syntax has these parts:

Part	Description
object	An object expression that evaluates to the Driver object.

FileOpen Method

Opens and loads the specified configuration file.

Syntax

object.**FileOpen** [= *string*]

The **FileOpen** method syntax has these parts:

Part	Description
object	An object expression that evaluates to the Driver object.
string	A string expression that specifies a configuration file name.

Remarks

If the specified file does not exist, a new configuration file is created with the specified file name. If an existing .csv file is specified, the file is imported.

FileSave Method

Saves changes to the file specified in the FullConfigFileName property.

Syntax

object.**FileSave**

The **FileSave** method syntax has these parts:

Part	Description
object	An object expression that evaluates to the Driver object.

FileSaveAs Method

Saves changes to the specified file.

Syntax

object.**FileSaveAs** [= *string*]

The **FileSaveAs** method syntax has these parts:

Part	Description
object	An object expression that evaluates to the Driver object.

string	A string expression that specifies a configuration file name.

Remarks

A configuration file specified with the I/O Driver's acronym as the extension will be saved as a reloadable binary file, a ".csv" extension will be saved as a comma separated value text file, and a ".txt" extension will result in the server window's log being written to a text file.

GetChannels Method

Returns all configured channels as handles and names.

Syntax

NumChannels = *object*.**GetChannels** *ChannelHandles ChannelNames*

The **GetChannels** method syntax has these parts:

Part	Description
NumChannels	A long integer expression specifying the number of channels returned
object	An object expression that evaluates to the Driver object.
ChannelHandles	A long integer array specifying the returned channel handles.
ChannelNames	A string array specifying the returned channel names

Remarks

The handles and names retrieved with this method can be used in an application to populate a tree view. The handles are used in methods such as SetProperyData and GetPropertyData.

GetDataBlocks Method

Returns all configured datablocks on the specified device as handles and names.

Syntax

NumDataBlocks = *object*.**GetDataBlocks** *DeviceHandle DataBlockHandles DataBlockNames*

The **GetDataBlocks** method syntax has these parts:

Part	Description
NumDataBlocks	A long integer expression specifying the number of datablocks returned.
object	An object expression that evaluates to the Driver object.
DeviceHandle	A long integer expression specifying the handle of the device from which to retrieve the datablocks.
DataBlockHandles	A long integer array specifying the returned datablock handles.
DataBlockNames	A string array specifying the returned datablock names.

Remarks

The handles and names retrieved with this method can be used in an application to populate a tree view. The handles are used in methods such as SetProperyData and GetPropertyData.

GetDevices Method

Returns all configured devices on the specified channel as handles and names.

Syntax

NumDevices = *object*.**GetDevices** *ChannelHandle DeviceHandles DeviceNames*

The **GetDevices** method syntax has these parts:

Part	Description
NumDevices	A long integer expression specifying the number of devices returned.
object	An object expression that evaluates to the Driver object.
ChannelHandle	A long integer expression specifying the handle of the channel from which to retrieve the devices.
DeviceHandles	A long integer array specifying the returned device handles.
DeviceNames	A string array specifying the returned device names.

Remarks

The handles and names retrieved with this method can be used in an application to populate a tree view. The handles are used in methods such as SetProperyData and GetPropertyData.

GetNameSpace Method

Returns the namespace type for the OPC server. Namespaces can be hierarchical or flat. The OPC Client Tree Browser is an example of a hierarchical namespace; it has three levels off the root, server, group, and item. A flat namespace is one that has only one level off the root.

Syntax

NameSpace = *object*.**GetNameSpace**(*IChannelHandle*)

The **GetNameSpace** method syntax has these parts:

Part	Description
NameSpace	An integer value that evaluates to a namespace.
object	An object expression that evaluates to the Driver object.
IChannelHandle	A long integer expression specifying the handle of the OPC server from which to retrieve the namespace.

Settings

The settings for *NameSpace* are:

Setting	Description
0	Hierarchical namespace
1	Flat namespace

GetProperties Method

Returns all properties for the specified type of object.

Syntax

NumProperties = *object*.**GetProperties** *ObjectType Properties*

The **GetProperties** method syntax has these parts:

Part	Description
NumProperties	A long integer expression specifying the number of properties returned in the *Properties* array.
object	An object expression that evaluates to the Driver object.
ObjectType	An integer expression specifying the type of object (channel, device or datablock) to return properties for.

Properties	A string array specifying the returned properties.

Settings

The settings for *ObjectType* are:

Setting	Description
1	Driver object
2	Channel object
3	Device object
4	DataBlock object

Remarks

This method allows an OLE Automation controller to retrieve all exposed properties for each server object. These property strings are used as properties in methods such as SetProperyData and GetPropertyData.

GetPropertyData Method

Returns data for each specified property on the specified object.

Syntax

Errors = *object*.**GetPropertyData** *ObjectHandleProperties PropertyData*

The **GetPropertyData** method syntax has these parts:

Part	Description
Errors	A returned array specifying corresponding property retrieving errors.
object	An object expression that evaluates to the Driver object.
ObjectHandle	A long integer expression specifying the handle of the object (channel, device or datablock) to return property data for.
Properties	An string array specifying device properties to return.
PropertyData	An array specifying corresponding device property data.

Remarks

When retrieving data for a single property, the property string may be specified as a string instead of a one-dimensional array. In this case, *PropertyData* will be returned as a single VARIANT and not an array.

InfoMessage Method

Sends an information message to the I/O Server event window.

Syntax

object.**InfoMessage** *Message*

The **InfoMessage** method syntax has these parts:

Part	Description
object	An object expression that evaluates to the Driver object.
Message	A string expression that specifies the text you want to display in the I/O Server event window as an information message.

IsBrowseSupported Method

Returns if the OPC server supports browsing.

Syntax

Browsable = *object*.**IsBrowseSupported**(*lChannelHandle*)

The **IsBrowseSupported** method syntax has these parts:

Part	Description
Browsable	An integer value that indicates whether the OPC server supports browsing.
object	An object expression that evaluates to the Driver object.
lChannelHandle	A long integer value that evaluates to the handle of the OPC server you want to browse.

Settings

The settings for *Browsable* are:

Setting	Description

0	The server does not support browsing.
1	The server supports browsing.

Poll Method

Polls all datablocks on the specified channel, device or datablock object.

Syntax

object.**Poll** *ObjectHandle*

The **Poll** method syntax has these parts:

Part	Description
object	An object expression that evaluates to the Driver object.
ObjectHandle	A long integer value specifying a server object handle received from server methods such as AddChannel() or GetDevices().

Remarks

The **Poll** method can be used to demand poll datablocks on the server by configuring the datablocks with a disabled poll time and calling Poll() with the appropriate datablock handle.

Quit Method

Shuts down the server.

Syntax

object.**Quit**

The **Quit** method syntax has these parts:

Part	Description
object	An object expression that evaluates to the Driver object.

Remarks

The **Quit** method can be used to explicitly issue a shutdown request to the server. If the server is connected to a FIX client or clients, this method will return an error.

ReadData Method

Returns the polled data for the specified datablock.

Syntax

Data = object.**ReadData** *DataBlockHandle RequestedDataType ItemOffset SubItemOffset Count SignalConditioning HighEgu LoEgu TimeStamp Quality*

The **ReadData** method syntax has these parts:

Part	Description
Data	An array or expression specifying the datablock's polled data.
object	An object expression that evaluates to the Driver object.
DataBlockHandle	A long integer expression specifying the handle of the datablock from which to retrieve polled data.
RequestedDataType	An integer expression specifying the returned data's data type.
ItemOffset	A long integer expression specifying the offset of the desired data from the start of the datablock.
SubItemOffset	An integer expression specifying the offset of the desired data from the start of the data item.
Count	A long integer expression specifying the number of consecutive data items to return.
SignalConditioning	An integer expression specifying the signal conditioning to apply to the returned data.
HighEgu	A single expression specifying the high Engineering Units value to be used for signal conditioning.
LoEgu	A single expression specifying the low Engineering Units value to be used for signal conditioning.
TimeStamp	An array or expression specifying the returned timestamp of the data.
Quality	An array or expression specifying the returned quality of the data.

Settings

The settings for *RequestedDataType* are:

Setting	Description
1	Not specified (VT_EMPTY)

2	2-byte signed integer (VT_I2)
3	4-byte signed integer (VT_I4)
4	4-byte real (VT_R4)
5	8-byte real (VT_R8)

The settings for *SignalConditioning* are:

Setting	Description
0	None (No Signal Conditioning)
1	12BN (12 Bit Binary with No Alarming)
2	12AL (12 Bit Binary with Alarming)
3	15BN (15 Bit Binary with No Alarming)
4	15AL (15 Bit Binary with Alarming)
5	LIN (Linear Signal Conditioning)
6	3BCD (3 Digit Binary Coded Decimal)
7	4BCD (4 Digit Binary Coded Decimal)
8	13BN (13 Bit Binary with No Alarming)
9	13AL (13 Bit Binary with Alarming)
10	8BN (8 Bit Binary with No Alarming)
11	8AL (8 Bit Binary with Alarming)

Remarks

The *RequestedDataType* settings are of type VARTYPE (an enumeration type used in VARIANT, TYPEDESC, OLE property sets, and safe arrays) and represent the minimum supported for this server. See your I/O Server documentation for more details regarding this setting.

The *SignalConditioning settings* represent the minimum supported for this server. See your I/O Server documentation for more details regarding this setting.

The *TimeStamp* part is returned as a Time value. The *Quality* part is returned as an integer representing OLE for Process Control (OPC) Quality flags. Please reference the OLE for Process Control Version 1.0 Specification for more information.

RemoveChannel Method

Removes the specified channel from the server configuration.

Syntax

object.**RemoveChannel** *ChannelHandle*

The **RemoveChannel** method syntax has these parts:

Part	Description
object	An object expression that evaluates to the Driver object.
ChannelHandle	A long integer value specifying a channel object handle received from a server method such as AddChannel().

RemoveDataBlock Method

Removes the specified datablock from the specified device's configuration.

Syntax

object.**RemoveDataBlock** *DeviceHandle DataBlockHandle*

The **RemoveDataBlock** method syntax has these parts:

Part	Description
object	An object expression that evaluates to the Driver object.
DeviceHandle	A long integer expression specifying the handle of the device from which to remove the datablock.
DataBlockHandle	A long integer expression specifying the handle of the datablock to remove.

RemoveDevice Method

Removes the specified device from the specified channel's configuration.

Syntax

object.**RemoveDevice** *ChannelHandle DeviceHandle*

The **RemoveDevice** method syntax has these parts:

Part	Description

object	An object expression that evaluates to the Driver object.
ChannelHandle	A long integer value specifying a channel object handle received from a server method such as AddChannel().
DeviceHandle	A long integer expression specifying the handle of the device to remove

SetPropertyData Method

Sets data for each specified property on the specified object.

Syntax

Errors = *object*.**SetPropertyData** *ObjectHandleProperties PropertyData*

The **SetPropertyData** method syntax has these parts:

Part	Description
Errors	A returned array specifying corresponding property setting errors.
object	An object expression that evaluates to the Driver object.
ObjectHandle	A long integer expression specifying the handle of the object (channel, device or datablock) to set property data for.
Properties	A string array specifying device properties to set.
PropertyData	An array specifying corresponding device property data.

Remarks

When setting data for a single property, the property string may be specified as a string instead of a one-dimensional array, and the property data value may also be specified as a single VARIANT instead of a one-dimensional array.

Start Method

Starts server polling.

Syntax

object.**Start**

The **Start** method syntax has these parts:

Part	Description
object	An object expression that evaluates to the Driver object.

Stop Method

Stops server polling.

Syntax

object.**Stop**

The **Stop** method syntax has these parts:

Part	Description
object	An object expression that evaluates to the Driver object.

Switch Method

Switches the specified channel or device object to/from backup.

Syntax

object.**Switch** *ObjectHandle*

The **Switch** method syntax has these parts:

Part	Description
object	An object expression that evaluates to the Driver object.
ObjectHandle	A long integer value specifying a server object handle received from server methods such as AddChannel() or GetDevices().

WarnMessage Method

Sends a warning message to the I/O Server event window.

Syntax

object.**WarnMessage** *Message*

The **WarnMessage** method syntax has these parts:

Part	Description

object	An object expression that evaluates to the Driver object.
Message	A string expression that specifies the text you want to display in the I/O Server event window as a warning message.

WriteData Method

Writes the specified data to the specified datablock.

Syntax

object.**WriteData** *DataBlockHandle ItemOffset SubItemOffset SignalConditioning HighEgu LoEgu Data*

The **WriteData** method syntax has these parts:

Part	Description
object	An object expression that evaluates to the Driver object.
DataBlockHandle	A long integer expression specifying the handle of the datablock to write to.
RequestedDataType	An integer expression specifying the returned data's data type.
ItemOffset	A long integer expression specifying the offset from the start of the datablock to write to.
SubItemOffset	An integer expression specifying the offset from the start of the data item to write to.
SignalConditioning	An integer expression specifying the signal conditioning to apply to the data.
HighEgu	A single expression specifying the high Engineering Units value to be used for signal conditioning
LoEgu	A single expression specifying the high Engineering Units value to be used for signal conditioning.
Data	An expression specifying the data to write.

Settings

The settings for *SignalConditioning* are:

Setting	Description

0	None (No Signal Conditioning)
1	12BN (12 Bit Binary with No Alarming)
2	12AL (12 Bit Binary with Alarming)
3	15BN (15 Bit Binary with No Alarming)
4	15AL (15 Bit Binary with Alarming)
5	LIN (Linear Signal Conditioning)
6	3BCD (3 Digit Binary Coded Decimal)
7	4BCD (4 Digit Binary Coded Decimal)
8	13BN (13 Bit Binary with No Alarming)
9	13AL (13 Bit Binary with Alarming)
10	8BN (8 Bit Binary with No Alarming)
11	8AL (8 Bit Binary with Alarming)

Remarks

The *SignalConditioning settings* represent the minimum supported for this server. See your I/O Server documentation for more details regarding this setting.

Only one item may be written at a time with this method

Property Summary

The following list contains the OSDK Type Library properties that are available.

Application

ConfigFileName

ConfigFilePath

DefaultConfigFileName

DefaultConfigFilePath

FileSaved

FullConfigFileName

FullDefaultConfigFileName

FullName

Name

Path

Running

Visible

Application Property

Returns the Application object. Read only.

Syntax

object.**Application**

The **Application** property syntax has these parts:

Part	Description
object	An object expression that evaluates to the Driver object.

ConfigFileName Property

Returns the active configuration file name. Read only.

Syntax

object.**ConfigFileName**

The **ConfigFileName** property syntax has these parts:

Part	Description
object	An object expression that evaluates to the Driver object.

ConfigFilePath Property

Returns the path specification for the active configuration file. Read only.

Syntax

object.**ConfigFilePath**

The **ConfigFilePath** property syntax has these parts:

Part	Description
object	An object expression that evaluates to the Driver object.

DefaultConfigFileName Property

Sets or returns the default configuration file name.

Syntax

object.**DefaultConfigFileName** [= *string*]

The **DefaultConfigFileName** property syntax has these parts:

Part	Description
object	An object expression that evaluates to the Driver object.
string	A string expression that evaluates to the server default configuration file path.

DefaultConfigFilePath Property

Sets or returns the default path specification for configuration files.

Syntax

object.**DefaultConfigFilePath** [= *string*]

The **DefaultConfigFilePath** property syntax has these parts:

Part	Description
object	An object expression that evaluates to the Driver object.
string	A string expression that evaluates to the server default configuration file path.

FileSaved Property

Returns the saved state of the configuration file. Read only.

Syntax

object.**FileSaved**

The **FileSaved** property syntax has these parts:

Part	Description
object	An object expression that evaluates to the Driver object.

FullConfigFileName Property

Returns the file specification for the active configuration file, including path. Read only.

Syntax

object.**FullConfigFileName**

The **FullConfigFileName** property syntax has these parts:

Part	Description
object	An object expression that evaluates to the Driver object.

FullDefaultConfigFileName Property

Returns the file specification for the default configuration file, including path. Read only.

Syntax

object.**FullDefaultConfigFileName**

The **FullDefaultConfigFileName** property syntax has these parts:

Part	Description
object	An object expression that evaluates to the Driver object.

FullName Property

Returns the file specification for the application, including path. Read only.

Syntax

object.**FullName**

The **FullName** property syntax has these parts:

Part	Description

object	An object expression that evaluates to the Driver object.

Name Property

Returns the name of the application. Read only.

Syntax

object.**Name**

The **Name** property syntax has these parts:

Part	Description
object	An object expression that evaluates to the Driver object.

Path Property

Returns the path specification for the application's executable file. Read only.

Syntax

object.**Path**

The **Path** property syntax has these parts:

Part	Description
object	An object expression that evaluates to the Driver object.

Running Property

Returns a value indicating whether the server is started or stopped. Read only.

Syntax

object.**Running**

The **Running** property syntax has these parts:

Part	Description
object	An object expression that evaluates to the Driver object.

Visible Property

Sets or returns a value indicating whether the main server window is visible or hidden.

Syntax

object.**Visible** [= *boolean*]

The **Visible** property syntax has these parts:

Part	Description
object	An object expression that evaluates to the Driver object.
boolean	A boolean expression specifying whether the server window is visible or hidden.

Settings

The settings for *boolean* are:

Setting	Description
True	Server window is visible.
False	Server window is hidden.

Example Summary

The following list contains the OSDK Type Library methods that are available.

AddChannel Method Examples

AddDataBlock Method Examples

AddDevice Method Examples

AddProperty Method Example

ConfigFileName Property Example

ConfigFilePath Property Example

CopyDevice Method Example

DebugMessage Method Example

DefaultConfigFileName Property Example

DefaultConfigFilePath Property Example

ErrorMessage Method Example

FileNew Method Example

FileOpen Method Example

FileSave Method Example

FileSaveAs Method Example

AddChannel Method Examples

This example adds a channel object to the server, specifying some properties.

```
Dim objDriver as Object
Dim lChanHandle As Long
Dim vProperties, vPropertyData, vErrors
```

```
‘ Create and return reference to Driver object
Set objDriver = CreateObject("Intellution.ITKDrv")

‘ Set up channel properties and data
vProperties = Array(“PrimaryPort”, "BackupPort", “PrimaryBaud", "BackupBaud")
vPropertyData = Array(“COM1”, “COM3”, “9600”, “9600”)
vErrors = Empty

‘ Add the channel
lChanHandle = objDriver.AddChannel vProperties vPropertyData vErrors
This example adds a channel object to the server with default properties.
Dim objDriver as Object
Dim lChanHandle As Long
Dim vProperties, vPropertyData, vErrors

‘ Create and return reference to Driver object
Set objDriver = CreateObject("Intellution.ITKDrv")

‘ Add the channel, passing empty parameters
lChanHandle = objDriver.AddChannel vProperties vPropertyData vErrors
```

AddDataBlock Method Examples

This example adds a datablock object to the server, specifying some properties.

```
Dim objDriver as Object
Dim lChanHandle, lDevHandle, lDbHandle As Long
Dim vProperties, vPropertyData, vErrors

‘ Create and return reference to Driver object
Set objDriver = CreateObject("Intellution.ITKDrv")

‘ Add a channel
lChanHandle = objDriver.AddChannel vProperties vPropertyData vErrors
```

```
‘ Add a device
lDevHandle = objDriver.AddDevice lChanHandle vProperties vPropertyData vErrors

‘ Set up datablock properties and data
vProperties = Array(“PrimaryPollTime”, “Description")
vPropertyData = Array(“1:00:00”, “Alarms”)
vErrors = Empty

‘ Add a datablock
lDbHandle = AddDatablock lDevHandle vProperties vPropertyData vErrors
```

This example adds a device object to the server with default properties.

```
Dim objDriver as Object
Dim lChanHandle, lDevHandle, lDbHandle As Long
Dim vProperties, vPropertyData, vErrors

‘ Create and return reference to Driver object
Set objDriver = CreateObject("Intellution.ITKDrv")

‘ Add a channel
lChanHandle = objDriver.AddChannel vProperties vPropertyData vErrors

‘ Add a device
lDevHandle = objDriver.AddDevice lChanHandle vProperties vPropertyData vErrors

‘ Add a datablock
lDbHandle = objDriver.AddDatablock lDevHandle vProperties vPropertyData vErrors
```

AddDevice Method Examples

This example adds a device object to the server, specifying some properties.

```
Dim objDriver as Object
Dim lDevHandle, lChanHandle As Long
```

```
Dim vProperties, vPropertyData, vErrors

‘ Create and return reference to Driver object
Set objDriver = CreateObject("Intellution.ITKDrv")

‘ Add a channel
lChanHandle = objDriver.AddChannel vProperties vPropertyData vErrors

‘ Set up device properties and data
vProperties = Array(“PrimaryStation”, "BackupStation", “Description")
vPropertyData = Array(“1”, “11”, “Gas well #121167”)
vErrors = Empty

‘ Add a device
lDevHandle = objDriver.AddDevice lChanHandle vProperties vPropertyData vErrors
```

This example adds a device object to the server with default properties.

```
Dim objDriver as Object
Dim lDevHandle, lChanHandle As Long
Dim vProperties, vPropertyData, vErrors

‘ Create and return reference to Driver object
Set objDriver = CreateObject("Intellution.ITKDrv")

‘ Add a channel
lChanHandle = objDriver.AddChannel vProperties vPropertyData vErrors

‘ Add a device
lDevHandle = objDriver.AddDevice lChanHandle vProperties vPropertyData vErrors
```

Application Property Example

This example sets the objDriverApp variable to the Driver Application object:

```
Dim objDriver as Object
Dim objDriverApp as Object
```

```
‘ Create and return reference to Driver object
Set objDriver = CreateObject("Intellution.ITKDrv")
objDriverApp = objDriver.Application
```

ConfigFileName Property Example

This example sets the Text property of a TextBox control to the server’s configuration file name.

```
Dim objDriver as Object

‘ Create and return reference to Driver object
Set objDriver = CreateObject("Intellution.ITKDrv")

‘ Get and set text
txtText1.Text = objDriver.ConfigFileName
```

ConfigFilePath Property Example

This example sets the Text property of a TextBox control to the server s configuration file path.

```
Dim objDriver as Object

 Create and return reference to Driver object
Set objDriver = CreateObject("Intellution.ITKDrv")
 Get and set text
txtText1.Text = objDriver.ConfigFilePath
```

CopyDevice Method Example

This example copies a device object from one channel to another.

```
Dim objDriver As Object
Dim lChanHandle, lDevHandle As Long
Dim lChanHandle2, lDevHandle2 As Long
Dim vProperties, vPropertyData, vErrors
Dim Err As Long

' Create and return reference to Driver object
```

```
Set objDriver = CreateObject("Intellution.OPCDrv")

'Add a channel
lChanHandle = objDriver.AddChannel(vProperties, vPropertyData, vErrors)

'Add a device
lDevHandle = objDriver.AddDevice(lChanHandle, vProperties, vPropertyData, vErrors)

'Add second channel
lChanHandle2 = objDriver.AddChannel(vProperties, vPropertyData, vErrors)

'Copy Device object from the first channel
Err = objDriver.CopyDevice(lDevHandle, "TargetDeviceName", lChanHandle2, False)
```

DebugMessage Method Example

This example sends a debug message to the server.

```
Dim objDriver As Object
Dim objIMessage As IDriverMessage

' Create and return reference to Driver object
Set objDriver = CreateObject("Intellution.OPCDrv")

'get reference to IDriverMessage
Set objIMessage = New OPCServer

'Send a debug message to the server
objIMessage.DebugMessage ("Debug message")
```

DefaultConfigFileName Property Example

This example sets the Text property of a TextBox control to the server's default configuration file path and then sets the server's default configuration file name to a new value:

```
Dim objDriver as Object

‘ Create and return reference to Driver object
```

```
Set objDriver = CreateObject("Intellution.ITKDrv")
‘ Get text
txtText1.Text = objDriver.DefaultConfigFileName
‘ Set text
objDriver.ConfigFileName = “Test.itk”
```

DefaultConfigFilePath Property Example

This example sets the Text property of a TextBox control to the server’s default configuration file path and then sets the server’s default configuration file path to a new value:

```
Dim objDriver as Object

‘ Create and return reference to Driver object
Set objDriver = CreateObject("Intellution.ITKDrv")
‘ Get text
txtText1.Text = objDriver.DefaultConfigFilePath
‘ Set text
objDriver.ConfigFileName = “C:\ConfigurationFiles\Test”
```

ErrorMessage Method Example

This example sends an error message to the server

```
Dim objDriver As Object
Dim objIMessage As IDriverMessage

' Create and return reference to Driver object
Set objDriver = CreateObject("Intellution.OPCDrv")

'get reference to IDriverMessage
Set objIMessage = New OPCServer

'Send an Error message to the server
objIMessage.ErrorMessage ("Error message")
```

FileNew Method Example

This example creates an empty server configuration:

```
Dim objDriver as Object

‘ Create and return reference to Driver object
Set objDriver = CreateObject("Intellution.ITKDrv")

‘ Create new configuration
objDriver.FileNew
```

FileOpen Method Example

This example opens the specified server configuration file.

```
Dim objDriver as Object

‘ Create and return reference to Driver object
Set objDriver = CreateObject("Intellution.ITKDrv")

‘ Open configuration file
objDriver.FileOpen “TestConfig.itk”
```

FileSave Method Example

This example saves the current server configuration:

```
Dim objDriver as Object

‘ Create and return reference to Driver object
Set objDriver = CreateObject("Intellution.ITKDrv")

‘ Save configuration file
objDriver.FileSave
```

FileSaveAs Method Example

This example saves the server configuration as a file type determined by the value of an OptionButton control:

```
Dim objDriver as Object
```

```
‘ Create and return reference to Driver object
Set objDriver = CreateObject("Intellution.ITKDrv")
If optFileType.Value = True
 ‘ Save configuration file as binary
 objDriver.FileSaveAs “TestConfig.itk”
Else
 ‘ Save configuration file as csv
 objDriver.FileSaveAs “TestConfig.csv”
```

FileSaved Property Example

This example checks the FileSaved property and prompts the user if the file needs to be saved. This may be typical of a client application’s shutdown code.

```
‘ Check if file is dirty
If objDriver.FileSaved = False Then
 ‘ Prompt user
 If MsgBox(“Save configuration file?”, vbOKCancel) = vbOK Then
  ‘ Save config file
  objDriver.Save
```

FullConfigFileName Property Example

This example sets the Text property of a TextBox control to the server’s full configuration file name:

```
Dim objDriver as Object
‘ Create and return reference to Driver object
Set objDriver = CreateObject("Intellution.ITKDrv")
‘ Get and set text
txtText1.Text = objDriver.FullConfigFileName
```

FullDefaultConfigFileName Property Example

This example sets the Text property of a TextBox control to the server’s default configuration file specification:

```
Dim objDriver as Object
‘ Create and return reference to Driver object
Set objDriver = CreateObject("Intellution.ITKDrv")
‘ Get text
```

```
txtText1.Text = objDriver.FullDefaultConfigFileName
```

FullName Property Example

This example sets the Caption property of a form to the full application name:

```
Dim objDriver as Object
‘ Create and return reference to Driver object
Set objDriver = CreateObject("Intellution.ITKDrv")
‘ Set Caption property
frmForm1.Caption = objDriver.FullName
```

GetChannels Method Example

This example returns the configured channels for the driver object and displays them in two TextBox Control arrays.

```
Dim objDriver as Object
Dim lNumChans, i As Long
Dim vChannelHandles, vChannelNames

‘ Create and return reference to Driver object
Set objDriver = CreateObject("Intellution.ITKDrv")

‘ Get configured channels
lNumChans = objDriver.GetChannels(vChannelHandles, vChannelNames)

‘ Display them
For i = 1 To lNumChans
txtText1(i) = vChannelNames(i - 1)
txtText2(i) = vChannelHandles(i - 1)
Next i
```

GetDataBlocks Method Example

This example returns the configured datablocks for a device and displays them in two TextBox Control arrays.

```
Dim objDriver as Object
Dim lNumChans, lNumDevs, lNumDBs i As Long
```

```
Dim vChannelHandles, vChannelNames, vDeviceHandles, vDeviceNames, vDataBlockHandles,
vDataBlockNames

‘ Create and return reference to Driver object
Set objDriver = CreateObject("Intellution.ITKDrv")

‘ Get configured channels
lNumChans = objDriver.GetChannels(vChannelHandles, vChannelNames)

‘ Get configured devices on first channel
lNumDevs = objDriver.GetDevices(vChannelHandles(0), vDeviceHandles, vDeviceNames)

‘ Get configured datablocks on first device
lNumDBs = objDriver.GetDataBlocks(vDeviceHandles(0), vDataBlockHandles, vDataBlockNames)

‘ Display them
For i = 1 To lNumDBs
txtText1(i) = vDataBlockNames(i - 1)
txtText2(i) = vDataBlockHandles(i - 1)
Next i
```

GetDevices Method Example

This example returns the configured devices for a channel object and displays them in two TextBox Control arrays.

```
Dim objDriver as Object
Dim lNumChans, lNumDevs i As Long
Dim vChannelHandles, vChannelNames, vDeviceHandles, vDeviceNames

‘ Create and return reference to Driver object
Set objDriver = CreateObject("Intellution.ITKDrv")

‘ Get configured channels
lNumChans = objDriver.GetChannels(vChannelHandles, vChannelNames)

‘ Get configured devices on first channel
```

```
lNumDevs = objDriver.GetDevices(vChannelHandles(0), vDeviceHandles, vDeviceNames)

‘ Display them
For i = 1 To lNumDevs
txtText1(i) = vDeviceNames(i - 1)
txtText2(i) = vDeviceHandles(i - 1)
Next i
```

GetNameSpace Method Example

```
Dim objDriver As Object
Dim objBrowseServer As IBrowseOpcServer
Dim lNumChannels As Long
Dim pvChannelHandles, pvChannelNames
Dim nNameSpace, nIsBrowseSupported as Integer
Dim x as Integer

' Create and return reference to Driver object
Set objDriver = CreateObject("Intellution.OPCDrv")
'get reference to IBrowseOpcServer
Set objBrowseServer = New OPCServer

' Get the number of channels
lNumChannels = objDriver.GetChannels(pvChannelHandles, pvChannelNames)
If lNumChannels <> 0 Then
For x = 0 To lNumChannels - 1 Step 1

' Check if browse is supported
 nIsBrowseSupported = objBrowseServer.IsBrowseSupported(pvChannelHandles(x))

 'Get the namespace for the server
 nNameSpace = objBrowseServer.GetNameSpace (pvChannelHandles(x))

Next x
End If
```

```
pvChannelHandles = Null
pvChannelNames = Null
```

GetProperties Method Example

This example gets the exposed properties for the driver object and displays them in a TextBox Control array.

```
Dim objDriver as Object
Dim lNumProps, i As Long
Dim vProperties

' Create and return reference to Driver object
Set objDriver = CreateObject("Intellution.ITKDrv")

' Get driver object properties
lNumProps = objDriver.GetProperties(1, vProperties)

' Display them
For i = 1 To lNumProps
txtText(i) = vProperties(i - 1)
Next i
```

GetPropertyData Method Example

This example returns a property data value for a channel object.

```
Dim objDriver as Object
Dim lChanHandle As Long
Dim vProperties, vPropertyData, vErrors

' Create and return reference to Driver object
Set objDriver = CreateObject("Intellution.ITKDrv")

' Add a channel
lChanHandle = objDriver.AddChannel vProperties vPropertyData vErrors

' Setup the VARIANTs
```

```
vProperties = “Name”
vErrors = Empty

‘ Set the property data
vErrors = objDriver.SetPropertyData lChanHandle vProperties vPropertyData

‘ Display the retrieved data
txtText1.Text = vPropertyData
```

This example returns multiple property data values for a channel object.

```
Dim objDriver as Object
Dim lChanHandle As Long
Dim vProperties, vPropertyData, vErrors

‘ Create and return reference to Driver object
Set objDriver = CreateObject("Intellution.ITKDrv")

‘ Add a channel
lChanHandle = objDriver.AddChannel vProperties vPropertyData vErrors

‘ Setup the VARIANTs
vProperties = Array( “PrimaryBaud", "BackupBaud", "BackupPort")
vErrors = Empty

‘ Set the property data
vErrors = objDriver.SetPropertyData lChanHandle vProperties vPropertyData

‘ Display the retrieved data
txtText1.Text = vPropertyData(0)
txtText2.Text = vPropertyData(1)
txtText3.Text = vPropertyData(2)
```

InfoMessage Method Example

This example sends an information message to the server.

```
Dim objDriver As Object
Dim objIMessage As IDriverMessage

' Create and return reference to Driver object
Set objDriver = CreateObject("Intellution.OPCDrv")

'get reference to IDriverMessage
Set objIMessage = New OPCServer

'Send an information message to the server
objIMessage.InfoMessage ("Information message")
```

IsBrowseSupported Method Example

```
Dim objDriver As Object
Dim objBrowseServer As IBrowseOpcServer

Dim lNumChannels As Long
Dim pvChannelHandles, pvChannelNames
Dim nNameSpace, nIsBrowseSupported as Integer
Dim x as Integer


' Create and return reference to Driver object
Set objDriver = CreateObject("Intellution.OPCDrv")

'get reference to IBrowseOpcServer
Set objBrowseServer = New OPCServer

' Get the number of channels
lNumChannels = objDriver.GetChannels(pvChannelHandles, pvChannelNames)


If lNumChannels <> 0 Then
```

```
For x = 0 To lNumChannels - 1 Step 1

 ' Check if browse is supported
 nIsBrowseSupported = objBrowseServer.IsBrowseSupported(pvChannelHandles(x))

 'Get the namespace for the server
 nNameSpace = objBrowseServer.GetNameSpace (pvChannelHandles(x))

Next x
End If

 pvChannelHandles = Null
 pvChannelNames = Null
```

Name Property Example

This example sets the Caption property of a form to the application name:

```
Dim objDriver as Object

‘ Create and return reference to Driver object
Set objDriver = CreateObject("Intellution.ITKDrv")

‘ Set Caption property
frmForm1.Caption = objDriver.Name
```

Path Property Example

This example sets the Caption property of a form to the path specification of the application:

```
Dim objDriver as Object

‘ Create and return reference to Driver object
Set objDriver = CreateObject("Intellution.ITKDrv")

‘ Set Caption property
frmForm1.Caption = objDriver.Path
```

Poll Method Example

This example demand polls all datablocks on a device.

```
Dim vHandles, vNames
Dim lNumBlocks, x As Long

‘ Get datablock handles
lNumBlocks = GetDataBlocks(lDeviceHandle, vHandles, vNames)
For x = 0 To lNumBlocks
objDriver.Poll(vHandles(x))
Next x
```

Quit Method Example

This example prompts the user to shut down the server.

```
‘ Prompt user
If MsgBox(“Shutdown server?”, vbOKCancel) = vbOK Then

‘ Exit server
objDriver.Quit
```

ReadData Method Example

This example reads data from the server in response to a CommandButton click.

```
Private Sub cmdReadData_Click()

Dim i As Long
Dim lDBHdl As Long
Dim lItemOffset, lSubItemOffset, lCount As Long
Dim vTimeStamp, vQuality, vData As Variant
Dim nDataType, nSigCond As Integer
Dim sngHighEgu, sngLowEgu As Single

‘ Get the datablock handle from a TreeView
lDBHdl = VBClient.GetHdlFromKey(VBClient.TreeView1.SelectedItem.key)

‘ Get the data type from a TextBox
nDataType = CInt(txtDataType)
```

```
‘ Get the item offset from a TextBox
lItemOffset = CLng(txtItemOffset)

‘ Get the subitem offset from a TextBox
lSubItemOffset = CLng(txtSubItemOffset)

‘ Get the count from a TextBox
lCount = CLng(txtNumItems)

‘ Get the signal conditioning from a ComboBox
nSigCond = cmbSigCond.ListIndex

‘ Get the high EGU from a TextBox
sngHighEgu = CSng(txtHiEgu)

‘ Get the low EGU from a TextBox
sngLowEgu = CSng(txtLowEgu)

vData = objDriver.ReadData(lDBHdl, nDataType, lItemOffset, lSubItemOffset, lCount, _
nSigCond, sngHighEgu, sngLowEgu, vTimeStamp, vQuality)

' Display results in TextBox arrays (set up for 150 maximum)
For i = 0 To 149
 If i <= UBound(vData) Then
  txtData(i) = vData(i)
  txtTime(i) = vTimeStamp(i)
  txtQuality(i) = vQuality(i)
 Else
  txtData(i) = Empty
  txtTime(i) = Empty
  txtQuality(i) = Empty
 End If
Next i
```

```
End Sub
```

RemoveChannel Method Example

This example adds and removes a channel from the server configuration.

```
Dim objDriver as Object
Dim lChanHandle As Long
Dim vProperties, vPropertyData, vErrors

‘ Create and return reference to Driver object
Set objDriver = CreateObject("Intellution.ITKDrv")

‘ Add a channel
lChanHandle = objDriver.AddChannel vProperties vPropertyData vErrors

‘ Manipulate channel here

‘ Remove channel object
objDriver.RemoveChannel lChannelHandle
```

RemoveDataBlock Method Example

This example removes a datablock from the server configuration.

```
Dim objDriver as Object
Dim lChanHandle, lDevHandle, lDbHandle As Long
Dim vProperties, vPropertyData, vErrors

‘ Create and return reference to Driver object
Set objDriver = CreateObject("Intellution.ITKDrv")

‘ Add a channel
lChanHandle = objDriver.AddChannel vProperties vPropertyData vErrors

‘ Add a device
lDevHandle = objDriver.AddDevice lChanHandle vProperties vPropertyData vErrors

‘ Add a datablock
lDbHandle = objDriver.AddDatablock lDevHandle vProperties vPropertyData vErrors
```

```
‘ Manipulate datablock here

‘ Remove datablock object
objDriver.RemoveDataBlock lDevHandle lDbHandle
```

RemoveDevice Method Example

This example removes a device from the server configuration.

```
Dim objDriver as Object
Dim lChanHandle, lDevHandle As Long
Dim vProperties, vPropertyData, vErrors

‘ Create and return reference to Driver object
Set objDriver = CreateObject("Intellution.ITKDrv")

‘ Add a channel
lChanHandle = objDriver.AddChannel vProperties vPropertyData vErrors

‘ Add a device
lDevHandle = objDriver.AddDevice lChanHandle vProperties vPropertyData vErrors

‘ Manipulate device here

‘ Remove device object
objDriver.RemoveDevice lChannelHandle lDevHandle
```

Running Property Example

This example sets the Value property of an OptionButton control to the running state of the driver:

```
Dim objDriver as Object

‘ Create and return reference to Driver object
Set objDriver = CreateObject("Intellution.ITKDrv")
‘ Get running state
optRun.Value = objDriver.Running
```

SetPropertyData Method Example

This example sets a property data value on a channel object.

```
Dim objDriver as Object
Dim lChanHandle As Long
Dim vProperties, vPropertyData, vErrors

‘ Create and return reference to Driver object
Set objDriver = CreateObject("Intellution.ITKDrv")

‘ Add a channel
lChanHandle = objDriver.AddChannel vProperties vPropertyData vErrors

‘ Setup the VARIANTs
vProperties = “Enabled”
vPropertyData = “0”
vErrors = Empty

‘ Set the property data
vErrors = objDriver.SetPropertyData lChanHandle vProperties vPropertyData
```

This example sets multiple property data values on a channel object.

```
Dim objDriver as Object
Dim lChanHandle As Long
Dim vProperties, vPropertyData, vErrors

‘ Create and return reference to Driver object
Set objDriver = CreateObject("Intellution.ITKDrv")

‘ Add a channel
lChanHandle = objDriver.AddChannel vProperties vPropertyData vErrors

‘ Setup the VARIANTs
vProperties = Array( “PrimaryBaud", "BackupBaud", "BackupPort")
```

```
vPropertyData = Array( “9600”, “9600”, “COM3”)
vErrors = Empty

‘ Set the property data
vErrors = objDriver.SetPropertyData lChanHandle vProperties vPropertyData
```

Start Method Example

This example starts server polling in response to a CommandButton click.

```
Private Sub Command1_Click ()
‘ Start polling
objDriver.Start
End Sub
```

StopMethod Example

This example stops server polling in response to a CommandButton click.

```
Private Sub Command1_Click ()
‘ Stop polling
objDriver.Stop
End Sub
```

Switch Method Example

This example switches all channels to/from their respective backup channels.

```
Dim vHandles, vNames
Dim lNumChans, x As Long

‘ Get channel handles
lNumChans = GetChannels(vHandles, vNames)
For x = 0 To lNumChans
 objDriver.Switch(vHandles(x))
Next x
```

Visible Property Example

This example sets the Value property of an OptionButton control to the server's visibility state and then sets the server window to the visible state:

```
Dim objDriver as Object
‘ Create and return reference to Driver object
Set objDriver = CreateObject("Intellution.ITKDrv")
‘ Get visibility
optVisible.Value = objDriver.Visible
‘ Set visibility
objDriver.Visible = True
```

WarnMessage Method Example

This example sends a warning message to the server.

```
Dim objDriver As Object
Dim objIMessage As IDriverMessage
' Create and return reference to Driver object
Set objDriver = CreateObject("Intellution.OPCDrv")
'get reference to IDriverMessage
Set objIMessage = New OPCServer
'Send a Warning message to the server
objIMessage.WarnMessage ("Warning message")
```

WriteData Method Example

This example writes data to the server in response to a CommandButton click.

```
Private Sub cmdWriteData_Click()
Dim i As Long
Dim lDBHdl As Long
Dim lItemOffset, lSubItemOffset As Long
Dim vData As Variant
Dim sngHighEgu, sngLowEgu As Single
Dim nSigCond As Integer

‘ Get the datablock handle from a TreeView
lDBHdl = VBClient.GetHdlFromKey(VBClient.TreeView1.SelectedItem.key)
```

```
‘ Get the item offset from a TextBox
lItemOffset = CLng(txtItemOffset)

‘ Get the subitem offset from a TextBox
lSubItemOffset = CLng(txtSubItemOffset)

‘ Get the signal conditioning from a ComboBox
nSigCond = cmbSigCond.ListIndex

‘ Get the high EGU from a TextBox
sngHighEgu = CSng(txtHiEgu)

‘ Get the low EGU from a TextBox
sngLowEgu = CSng(txtLowEgu)

‘ Get the data from a TextBox
vData = txtData
objDriver.WriteData(lDBHdl, lItemOffset, lSubItemOffset, nSigCond, sngHighEgu, sngLowEgu,
vData)
End Sub
```

Obtaining an Interface Pointer to the Server Object

There are two different ways to obtain an interface pointer to the server object:

Early Binding – at design time

Using the Class name, declare an object as the class type. The driver class name, as shipped, is ITKServer.

Example

Public iDrv AsNew ITKServer

Late Binding – during run-time

Declare a generic object, then use the CreateObject function to obtain the interface pointer.

Public iDrv As Object

Set iDrv = CreateObject("Intellution.ITKDrv")

Early binding has better performance, but late binding allows more diversity.

OPC Client Only

The following topics apply only to the OPC Client driver.

iBrowseOPCServer Interface

iDriverMessage Interface

Sample Application

IBrowseOPCServer Interface

Lets the OPC Client determine if browsing on the OPC server. If browsing is supported, the interface lets the OPC Client browse the server for item ids and access paths.

NOTE: You can only use the IsBrowseSupported and GetNameSpace methods from this interface. The other methods are for internal use and not supported for third-party development.

IDriver Message

Lets the driver send messages to the I/O Server's event window. Visual Basic applications can also use this interface to send messages to the event window.

Sample Application

To assist you in creating a Visual Basic application, we supply source code for a sample application. This source code shows you how to:

- Add a channel, device, and datablock to the driver configuration.
- Retrieve properties from a channel, device, and datablock.
- Copy a device.
- Delete a channel.
- Poll for data.
- Handle data returned from a poll of the server.
- Write data to a datablock.
- Start and stop the server.
- Determine if the server is running.
- Show and hide the server window.
- Send messages to the server window.
- Shut down the server.
- Read and modify the default configuration file name and path.
- Open and save a driver configuration file.

To review the source code, click the See Also button.

Sample Application Source Code

```
''THIS DEMO PROVIDED THE USE OF THE FOLLOWING METHODS OR PROPERTIES OF
'THE IDRIVER INTERFACE AND IDRIVERMESSAGE INTERFACE OF THE OPCDRV:
'
'FileOpen, FileSave, AddChannel, AddDevice, AddDataBlock, GetChannels,
'GetDevices, GetDataBlocks, CopyDevice, RemoveChannel, GetPropertyData,
'SetPropertyData, Poll, Quit, ReadData, WriteData, Start,Stop,
'DefaultConfigFileName, DefaultConfigFilePath,FileSaved, Running, Visible
'InfoMessage, DebugMessage
'

Dim objDriver As Object
Dim objMessage As IDriverMessage

Dim vProperties, vPropertyData, vErrors
Dim Err As Long
Dim lNumChans, lNumDevs, lNumDataBlks As Long
Dim vChannelHandles, vChannelNames
Dim vDeviceHandles, vDeviceNames
Dim vDataBlockHandles, vDataBlockNames
```

```
Dim szFileName As String
Dim szPath As String

Dim lChanHandle1, lChanHandle2 As Long
Dim lDevHandle1, lDevHandle2 As Long
Dim lDataBlkHandle1 As Long
Dim vData As Variant
Dim vQuality As Variant
Dim i As Long
Dim szEnabled, szDisabled As String

Dim szServerName As String
Dim szProgID As String
Dim szItemID As String
Dim szChannel1, szChannel2 As String
Dim szDevice1, szDevice2 As String
Dim szDataBlk1, szDataBlk2 As String
Dim szConfigFileName As String

'Initialize the string

szEnabled = "1"
szDisabled = "0"
szServerName = "Intellution.OPCDrv"
szProgID = "SIMOPC.SIMOPC.2"
szChannel1 = "MyChannel1"
szChannel2 = "MyChannel2"
szDevice1 = "MyDevice1"
szDevice2 = "MyDevice2"
szItemID = "FLOAT_0"
szDataBlk1 = "MyDataBlock1"
szDataBlk2 = "MyDataBlock2"
szConfigFileName = "Fix.OPC"

'Create and return reference to Driver object

Set objDriver = CreateObject(szServerName)

'Get Configure file name

szFileName = objDriver.DefaultConfigFileName


'get the path to the configure file

szPath = objDriver.DefaultConfigFilePath


'check if this is the file we needed

If szFileName = szConfigFileName Then
'do nothing
```

```
Else
'open the configuration file
objDriver.FileOpen (szPath + szConfigFileName)
End If

'Display the Server Window if it is not visible
If objDriver.Visible = 0 Then
objDriver.Visible = 1
End If

'get reference to IDriverMessage
Set objMessage = New OPCServer

'Get Channel Information
lNumChans = objDriver.GetChannels(vChannelHandles, vChannelNames)

'Check if the channel exists
For i = 0 To lNumChans - 1
If vChannelNames(i) = szChannel1 Then
'Remove the channel
Err = objDriver.RemoveChannel(vChannelHandles(i))
End If
Next i

'Add a new channel to the server

'Set up channel properties and data
vProperties = Array("ProgID", "Name", "Enabled")
vPropertyData = Array(szProgID, szChannel1, szEnabled)
vErrors = Empty
lChanHandle1 = objDriver.AddChannel(vProperties, vPropertyData, vErrors)

'Send an info message to Server Window
objMessage.InfoMessage ("New Channel Added")

'Add a device to the channel
```

```
'Set up device properties and data

vProperties = Array("Name", "Enabled", "Description")
vPropertyData = Array(szDevice1, szEnabled, "My First Device")
vErrors = Empty

lDevHandle1 = objDriver.AddDevice(lChanHandle1, vProperties, vPropertyData, vErrors)


'Send an info message to Server Window

objMessage.InfoMessage ("New Device Added")


'Add a DataBlock to the device

'Set up DataBlock properties and data

vProperties = Array("Name", "ItemID")
vPropertyData = Array(szDataBlk1, szItemID)
vErrors = Empty


lDataBlkHandle1 = objDriver.AddDataBlock(lDevHandle1, vProperties, vPropertyData,vErrors)


'Send an info message to Server Window

objMessage.InfoMessage ("New DataBlock Added")


'Enable the datablock

vProperties = Array("Enabled")
vPropertyData = Array(szEnabled)
vErrors = Empty


vErrors = objDriver.SetPropertyData(lDataBlkHandle1, vProperties, vPropertyData)


objMessage.InfoMessage ("DataBlock Enabled")


'Get driver status information, if not started, start the driver

If objDriver.Running = 0 Then

objDriver.Start

End If


'Write data to the datablock
```

```
vData = "15.00"
Err = objDriver.WriteData(lDataBlkHandle1, 0, 0, 0, 0#, 0#, vData)
objMessage.DebugMessage ("WriteData() is called")

'Poll the device
objDriver.Poll (lDevHandle1)

'Read data back from the server
vData = objDriver.ReadData(lDataBlkHandle1, 0, 0, 0, 1, 0, 0, 0, 0, vQuality)
objMessage.DebugMessage ("ReadData() is called")

'Create a second channel
'Get Channel Information
lNumChans = objDriver.GetChannels(vChannelHandles, vChannelNames)

'Check if the channel exists
For i = 0 To lNumChans - 1
If vChannelNames(i) = szChannel2 Then
'Remove the channel
Err = objDriver.RemoveChannel(vChannelHandles(i))
End If
Next i

'Set up channel properties and data
vProperties = Array("ProgID", "Name", "Enabled")
vPropertyData = Array(szProgID, szChannel2, szEnabled)
vErrors = Empty

lChanHandle2 = objDriver.AddChannel(vProperties, vPropertyData, vErrors)

'Copy Device object from the first channel
Err = objDriver.CopyDevice(lDevHandle1, szDevice2, lChanHandle2, False)

'Get device infor from channel2
lNumDevs = objDriver.GetDevices(lChanHandle2, vDeviceHandles, vDeviceNames)
```

```
For i = 0 To lNumDevs - 1
If vDeviceNames(i) = szDevice2 Then
lDevHandle2 = vDeviceHandles(i)
End If
Next i

'check if the device is enabled, if not, enable it

'Enable the datablock
vProperties = Array("Enabled")
vPropertyData = Empty
vErrors = Empty

vErrors = objDriver.GetPropertyData(lDevHandle2, vProperties, vPropertyData)


If vPropertyData(0) = szEnabled Then

'disable ths device

vPropertyData = Array(szDisabled)

vErrors = objDriver.SetPropertyData(lDevHandle2, vProperties, vPropertyData)

End If

'Enable the device

vProperties = Array("Enabled")
vPropertyData = Array(szEnabled)

vErrors = objDriver.SetPropertyData(lDevHandle2, vProperties, vPropertyData)


'Get DataBlocks information

lNumDataBlks = objDriver.GetDataBlocks(lDevHandle2, vDataBlockHandles, vDataBlockNames)

For i = 0 To lNumDataBlks - 1
'Enable every datablocks

vErrors = objDriver.SetPropertyData(vDataBlockHandles(i), vProperties, vPropertyData)

Next i


'save the configuration file

If objDriver.FileSaved = 0 Then

objDriver.FileSave

objMessage.InfoMessage ("Configuration file saved")

End If
```

```
'Stop the driver
objDriver.Stop

'ShutDown
objDriver.Quit
```

Index