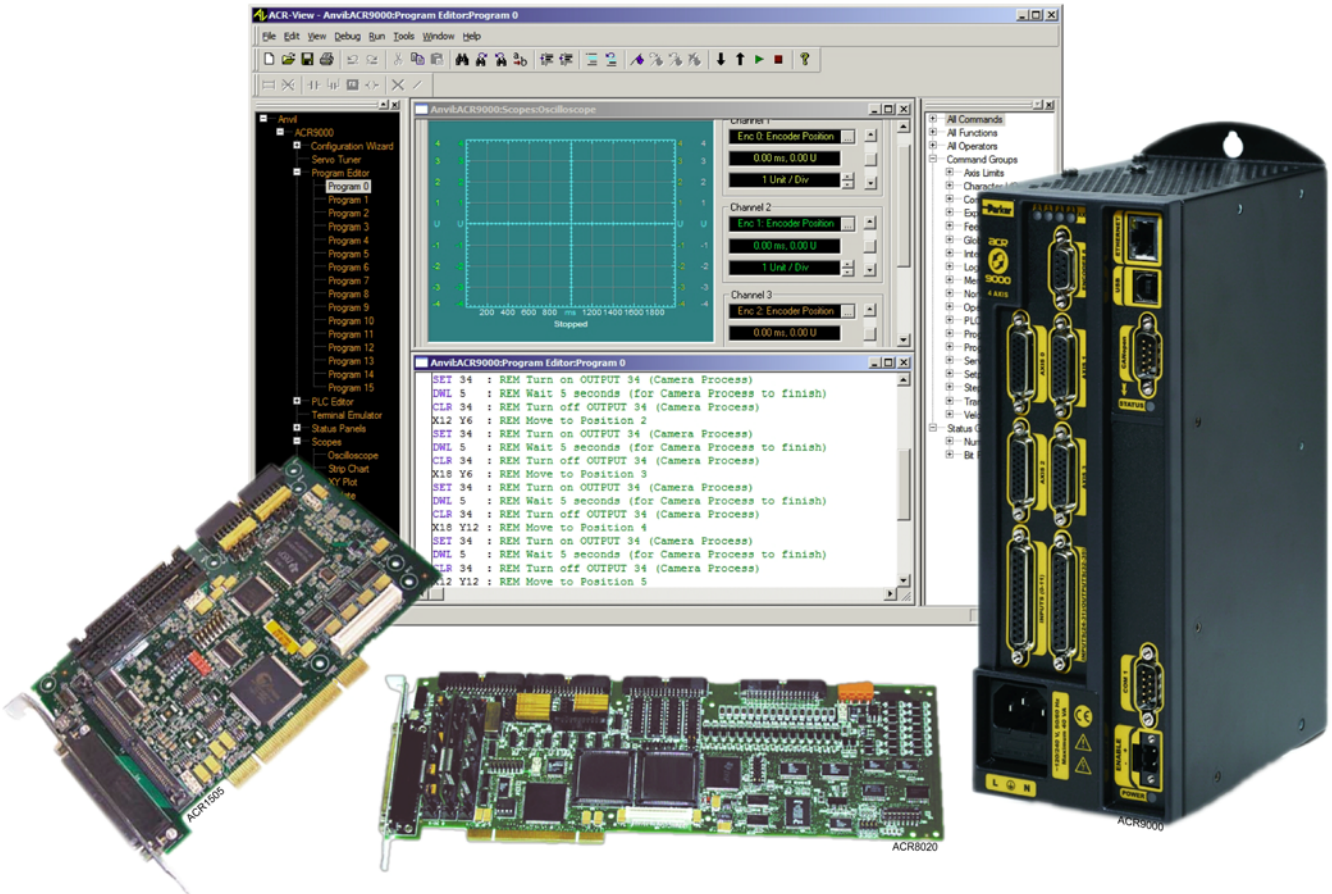
Parker Automation

p/n 88-028040-01A

Motion COMponents ActiveX User's Guide for ACR Series Products

Effective: July 2006



User Information

Warning — ACR Series products are used to control electrical and mechanical components of motion control systems. You should test your motion system for safety under all potential conditions. Failure to do so can result in damage to equipment and/or serious injury to personnel.

Technical Assistance
Contact your local automation technology center (ATC) or distributor.

North America and Asia
Parker Hannifin
5500 Business Park Drive
Rohnert Park, CA 94928
Telephone: (800) 358-9070 or (707) 584-7558
Fax: (707) 584-3793
Email: emn_support@parker.com
Internet: http://www.parkermotion.com

Germany, Austria, Switzerland
Parker Hannifin
Postfach: 77607-1720
Robert-Bosch-Str. 22
D-77656 Offenburg
Telephone: +49 (0) 781 509-0
Fax: +49 (0) 781 509-176
Email: sales.hauser@parker.com
Internet: http://www.parker-emd.com

Europe (non-German speaking)
Parker Hannifin plc
Electromechanical Automation, Europe
Arena Business Centre
Holy Rood Close
Poole
Dorset, UK
BH17 7BA
Telephone: +44 (0) 1202 606300
Fax: +44 (0) 1202 606301
Email: support.digiplan@parker.com
Internet: http://www.parker-emd.com

Italy
Parker Hannifin
20092 Cinisello Balsamo
Milan, Italy via Gounod, 1
Telephone: +39 02 6601 2478
Fax: +39 02 6601 2808
Email: sales.sbc@parker.com
Internet: http://www.parker-emd.com

Parker Automation

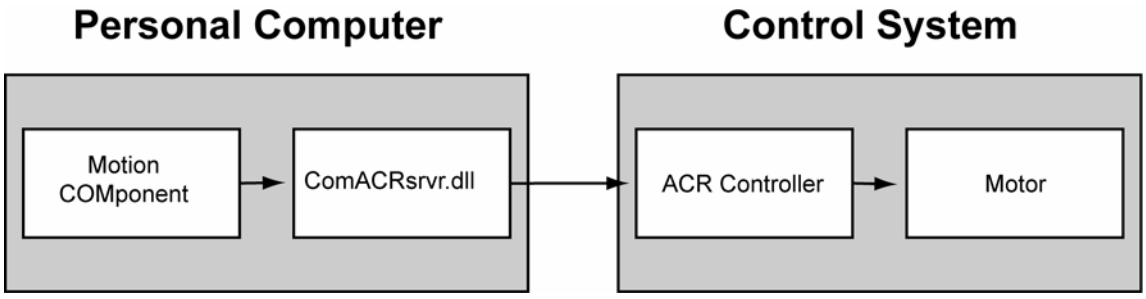
Technical Support E-mail
emn_support@parker.com

Table of Contents

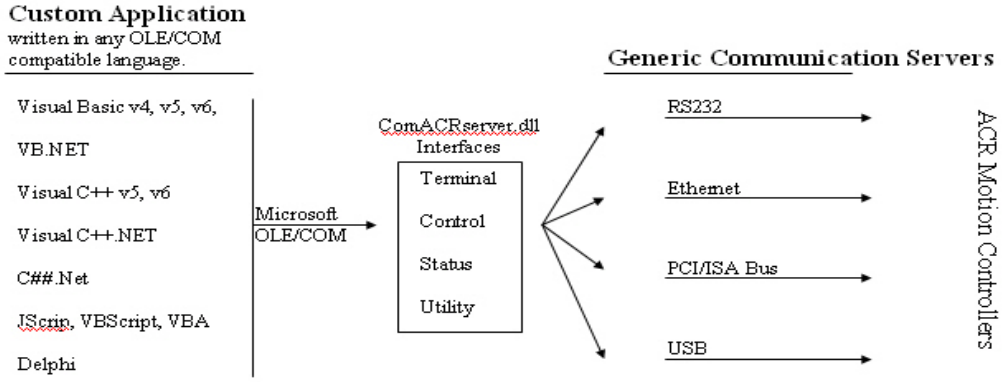
Overview

The .NET version of Motion COMponents for ACR series motion controllers acts as a wrapper for the ComACRsrvr.dll, which you can use with your motion control applications. For more information about the ComACRsrvr.dll, see the

The following illustrates the interaction between a personal computer and control system.



The Motion COMponents let you access the ComACRsrvr.dll using many different Windows based programming environments. The following diagram illustrates this concept.



Connection Control Properties and Methods

The Connection Control is the primary control responsible for creating the communication connection with the ACR controller from the application. It will have a group of properties and methods to facilitate communication.

Properties

- **Boolean** **OnConnectTest**
- **String** **ComVersion**
- **Long** **Port**
- **Long** **BPS**
- **Long** **Bus**
- **Long** **Card**
- **String** **IPAddr**
- **Boolean** **IsOffline**
- **Long** **Transport**
- **String** **USBSerialNumber**
- **Long** **nDevice**

Methods

- **Void** **Connect (nTransport as Long, nIndex as Long)**
- **Boolean** **TestConnect ()**
- **Void** **SetWatchDog (nInterval as Long, nRetries as Long)**
- **Void** **Disconnect ()**

Properties

OnConnectTest

Description A wrapper Boolean property that determines the Connection and verified after connection is established. If it is true then both Connect method and TestConnect method will be called. If it is False then only the Connect method will be called.

Property **OnConnectTest**
Return Type Boolean
Range N/A
Default True
Example ACRConnection1.OnConnectTest=True

ComVersion

Description A wrapper read only string that holds the version number of the comACRSrvr.dll file.

Property **ComVersion**
Return Type String
Range N/A
Default N/A
Example Dim strVer as String
strVer = ACRConnection1.ComVersion

Port

Description A wrapper long used to set the communications port of the computer to which the serial ACR device is connected. Only used for RS232 Serial connections.

Property **Port**
Return Type Long
Range 1 – 256
Default 1
Example ACRConnection1.Port =1 'For COM1 port

BPS

Description A wrapper long used to set the speed of the serial port in Bits per Second for the serial port specified by Port. Only used for RS232 Serial communication.

Property **BPS**
Return Type Long
Range 9600, 19200, 38400
Default 38400
Example ACRConnection1.BPS =38400

Bus

Description A wrapper long is indicating the type of Bus Card being used. A Value of 0 indicates an ACR PCI Bus Card and a value of 1 indicates an ACR ISA Bus Card. Only used for Bus communication.

Property **BUS**
Return Type Long
Range 0-1
Default 0 (PCI)
Example ACRConnection1.Bus =0 'For PCI Bus

Card

Description A wrapper long to set the index number of the ACR Controller. Only used for Bus communication.

Property **Card**
Return Type Long
Range N/A
Default 0
Example ACRConnection1.Card =1

IPAddr

Description A wrapper string representing the IP address of an ACR Controller. Only used for Ethernet communication.

Property **IPAddr**
Return Type String
Range N/A
Default 192.168.10.40
Example ACRConnection1.IPAddr = "172.20.22.10"

IsOffline

Description A wrapper read only Boolean indicating whether the connection is Offline. It will set to false once the connection is successful.

Property **IsOffline**
Return Type Boolean
Range N/A
Default TRUE
Example Dim blnIsOffline as Boolean
blnIsOffline = ACRConnection1.IsOffline

Transport

Description A long indicating the physical communication layer being used.

Property **Transport**
Return Type Long
Range 0-4
Default N/A
Example ACRConnection1.Transport = 3 ‘Ethernet connection

USBSerialNumber

Description A string holding the serial number of the USB port.

Property **USBSerialNumber**
Return Type String
Range N/A
Default N/A
Example
```
Dim strUSBSerialNumber as String
strUSBSerialNumber  = ACRConnection1.USBSerialNumber
```

---

nDevice

Description Specifies the ACR model.

Property **nDevice**
Return Type Long
Range 0-8
Default 8
Example
```
Dim lngnDevice as long
lngnDevice = ACRConnection1.nDevice
```

Methods

Connect

Description Establish a connection of type transport to an ACR Controller.

Signature **Connect** (nTransport as Long, nIndex as Long)

Return Type N/A

Parameters

nTransport — Indicates the physical communication layer being used, or no layer when offline. (0-Offline, 1- Bus, 2- Serial, 3-Ethernet, 4-USB)

nIndex — Transport type dependant data

All Interfaces initially come up with transport = Offline. Each transport type has its own data requirements for connecting.

Transport	Connection Requirements
Offline	The nIndex value can be any value.
Bus	The nIndex value must be the card index assigned during installation. To find this card number see parameter P7041, or DIP switch setting on some cards. The Bus property must be set to ISA or PCI depending on the card type.
Serial	The nIndex value is the index of the card, which is typically zero. In a daisy chain configuration, this number identifies the specific controller. The Port must be set to the PC communications port that will be used, and the BPS must be set to the desired bits per second rate.
Ethernet	The nIndex value can be any value. The IPAddrproperty must be set to the IP address of the ACR controller.
USB	The nIndex value is the unique Serial ID of the ACR device. If this is set to zero, the first ACR USB device found will be connected.

Any transport specific properties (i.e. IPAddr for Ethernet, etc.) should be set prior to calling Connect method.

Example

```
ACRConnection1.IPAddr=”192.9.200.58”
ACRConnection1.Transport=3
Call ACRConnection1.Connect (ACRConnection1.Transport, 0)
```

TestConnect

Description Verifies that an ACR Controller is connected.

Signature **TestConnect** ()

Return Type Boolean

Parameters N/A

Return A command is sent and the return value verified. If this process succeeds, an ACR's presence is presumed and TRUE is returned. Otherwise FALSE is returned. When the transport type = Offline, this method always returns FALSE.

Example

```
Dim blnTestConnect as Boolean
blnTestConnect=ACRConnection1.TestConnect
```

SetWatchDog

Description Modifies the Watchdog values.

Signature **SetWatchDog** (nInterval as Long, nRetries as Long)

Return Type N/A

Parameters

nInterval — The time in milliseconds, between sending test keep-alive strings to the ACR device.

nRetries — The number of times the keep-alive test string message is sent to the ACR device, with no valid reply, before attempting to reconnect to the ACR device.

Return The Ethernet transport currently has Watchdog functionality. The ACR controller uses a separate port to receive a coded command string (keep-alive message), and echoes it back to the sender. If the Communications Server fails to get a response to a keep-alive message in nInterval*nRetries milliseconds, the Communications Server attempts to reconnect

Example

```
Dim lngInterval as Long
Dim lngRetries as Long
lngInterval= 2000
lngRetries=4
Call ACRConnection1.SetWatchDog (lngInterval, lngRetries)
```

Disconnect

Description	A wrapper method is used to disconnect the current communication transport.
Signature	**Disconnect** ()
Return Type	N/A
Parameters	N/A
Return	Implicitly calls Connect (0,0) to switch to Offline mode.
Example	Call ACRConnection1.Disconnect

Terminal Control Properties, Methods and Events

The Terminal Control is dual pane multi-line edit control. The right pane is used as an editor and the left pane is used as a terminal where it displays the output of the ACR Controller as well as inputting and sending command to the ACR Controller. There are also buttons below the editor and terminal used to download and upload programs, downloading OS etc.

Terminal Control contains the following Properties and Methods.

Properties

- **Long DataWaitRate**
- **OLE_COLOR TerminalBackcolor**
- **OLE_COLOR TerminalForecolor**
- **OLE_COLOR EditorBackcolor**
- **OLE_COLOR EditorForecolor**
- **Object Connectioncontrol**

Methods

- **String DataRead ()**
- **Void DataWrite (send as Variant)**
- **Void DownloadFile (bstrPrg as String, bstrFile as String)**
- **Void DownloadOS (nDevice as Long, bstrFile as String)**
- **Long GetDownloadStatus (nTotal as Long, nBytes as Long)**
- **Void UploadFile (bstrPrg as String, bstrFile as String)**
- **Void StopDownload ()**

Events

- **Void DataWaiting ()**

Properties

DataWaitRate

Description	This property is a wrapper property setting the minimum time between status alerts in milliseconds. The default value is 50 ms.
Property	**DataWaitRate**
Return Type	Long
Range	N/A
Default	50 ms
Example	ACRTerminalcontrol1.DataWaitRate=100

TerminalBackColor

Description	The back color for the Terminal rich text box can be changed to user chosen color.
Property	**TerminalBackColor**
Return Type	OLE_COLOR
Range	N/A
Default	vbBlue
Example	ACRTerminalcontrol1.TerminalBackcolor =rgb (0, 0, 153)

TerminalForeColor

Description	The fore color for the Terminal rich text box can be changed to user chosen color.
Property	**TerminalForeColor**
Return Type	OLE_COLOR
Range	N/A
Default	vbWhite
Example	ACRTerminalcontrol1.TerminalForecolor =rgb (255, 255, 255)

EditorBackColor

Description	The back color for the Editor rich text box can be changed to user chosen color.
Property	**EditorBackColor**
Return Type	OLE_COLOR
Range	N/A
Default	vbWhite
Example	ACRTerminalcontrol1.EditorBackcolor =rgb (255, 255, 255)

EditorForeColor

Description	The fore color for the Editor rich text box can be changed to user chosen color.
Property	**EditorForeColor**
Return Type	OLE_COLOR
Range	N/A
Default	vbBlack
Example	ACRTerminalcontrol1.EditorForeColor =rgb (0,0,0)

Connectioncontrol

Description	This property will hold the reference of the instance of connection control.
Property	**ConnectionControl**
Return Type	Object
Range	N/A
Default	N/A
Example	Set ACRTerminalcontrol1.ConnectionControl=ACRConnection1

Methods

DataRead

Description	A wrapper method used to get any ASCII data from the ACR controller. This method wraps the method of Read available in the Terminal interface of ComACRsrvr.dll. This method is used to read the Data available in the buffer of ACR Controller. This method is mainly used to display the output of ACR Controller in the left-pane of Terminal controller.
Signature	**DataRead** ()
Return Type	String
Parameters	N/A
Return	The output of the controller will be stored in a string variable.
Example	Dim OutputData as string OutputData = ACRTerminalcontrol1.DataRead ()

DataWrite

Description	This method wraps the method of Write available in the Terminal interface of ComACRsrvr.dll. This method is used to send the command to the ACR Controller.
Signature	**DataWrite** (send as variant)
Return Type	N/A
Parameters	
send	The data used to send to the ACR Controller.
Return	N/A
Example	Dim InputData as string InputData = “AXIS0 JOG FWD X” ACRTerminalcontrol1.DataWrite (InputData)

DownloadFile

Description This method wraps the method available in the ACRTerminalcontrol1.Downloadfile. This method is used to transfer a .8K file to the controller.

Signature DownloadFile (bstrPrg as string, bstrFile as string)

Return Type N/A

Parameters

bstrPrg holds the Program location in Controller (Optional Parameter) either Blank or Program Location.

bstrFile holds the path of the downloaded text file.

Return N/A

Example

```
Dim filePath as string
filePath = “C:\DownloadData\PROG0.8K”
ACRTerminalcontrol1.DownloadFile (“”, filePath)
```

DownloadOS

Description This method wraps the method of DownloadOS available in the Utility interface of ComACRsrvr.dll. This method is used to download the new Operating system to the controller.

Signature **DownloadOS** (ndevice As Long, bstrfile As String)

Return Type N/A

Parameters

nDevice Specifies the type of ACR Controller Modal.

bstrFile Specifies the fully qualified file name of the new Operating System.

Return N/A

Example

```
Dim filePath as string
Dim nDevice as integer
nDevice = 8
filePath = “C:\DownloadData\NEWOS.OPS”
ACRTerminalcontrol1.DownloadOS (nDevice, filePath)
```

GetDownloadStatus

Description This method wraps the method of GetDownloadStatus available in the Utility interface of ComACRsrvr.dll. This method is used to get the Current Status of Active download.

Signature **GetDownloadStatus** (nTotal As Long, nBytes As Long)

Return Type Long

Parameters

nTotal	Specifies the Total number of bytes to be transferred.
nBytes	Specifies the Total number of bytes transferred so far.

Return Return value is the status of the active Download.

Example

```
Dim I as integer, j as integer, k as integer
i = 0
j = 0
k = 0
k = ACRTerminalcontrol1.GetDownloadStatus (i ,j)
Msgbox “Total Bytes to be transferred so far:” & I
Msgbox “Total number of bytes transferred so far:” &j
Msgbox “Current Active Status Download” & k
```

Uploadfile

Description This method wraps the method of “Uploadfile” available in the Utility interface of ComACRsrvr.dll. Uploads the AcroBasic Program or PLC program from the ACR to the PC.

Signature **UploadFile** (bstrPrg as string, bstrFile as string)

Return Type N/A

Parameters

bstrPrg	Specifies the location to which files are uploaded.
bstrFile	Specifies the fully qualified path and name of the file to download.

Return This method blocks any other instruction from running until the upload is complete. There is no checking of the code uploaded.

Example

```
Dim filePath as string
filePath = “C:\UploadData\PROG0.8k”
ACRTerminalcontrol1.UploadFile (“PROG 0”, filePath)
```

StopDownload

Description	This method wraps the method of “StopDownload” available in the Utility interface of ComACRsrvr.dll. This method aborts the file transfer to the Controller.
Signature	**StopDownload** ()
Return Type	N/A
Parameters	N/A
Return	N/A
Example	ACRTerminalcontrol1.StopDownload ()

Events

DataWaiting

Description	A wrapper callback method is fired whenever the Data available in the Buffer of ComACRsrvr.dll. Using this event the terminal emulator displays the result.
Signature	**DataWaiting** ()
Return Type	N/A
Parameters	N/A
Return	N/A
Example	DRead As BOXBRIDGELib.Terminal ACRTerminalcontrol1.DRead_DataWaiting ()

BitStatus Control Properties, Methods and Events

The Bit Status Control will be an array of up to 32-bit indicators with labels. The number of bit indicators is dependent on the value of the property BitMask. Bit Data are retrieved in 32-bit blocks. The Bitlabels will constantly update the values of the bits with a color for TRUE and a different color for FALSE.

BitStatus Control contains the following Properties, Methods, and Events.

Properties

- **Long** **BitSelect**
- **Long** **BitPlacement**
- **Long** **Pollrate**
- **OLE_COLOR** **TrueColor**
- **OLE_COLOR** **FalseColor**
- **Variant** **BitMask**
- **String** **BitMaskCSV**
- **Boolean** **AutoSize**
- **Object** **ConnectionControl**

Methods

- **Void** **BitLabel (strBitName as String)**
- **Void** **SetBit ()**
- **Void** **ClearBit ()**
- **Boolean** **GetValue ()**
- **Boolean** **IsFlagSet (nFlgGrp as Long, nFlgNdx as Long)**

Events

- **Void** **DataChanged ()**

Properties

BitSelect

Description Represents the user-selected bit number.

Property **BitSelect**
Return Type Long
Range N/A
Default N/A
Example ACRBitStatus1.BitSelect=550

BitPlacement

Description Represents the position of the bit within the 32-bit response. Range 0-31. This is read only property.

Property **BitPlacement**
Return Type Long
Range N/A
Default N/A
Example
```
Dim lngPlacement as Long
'If BitSelect=34 then result is 2
lngPlacement=ACRBitStatus1.BitPlacement
```

Pollrate

Description Refreshing of bits with ACR Controller will take place based on the Pollrate. Default is set to 100 ms.

Property **Pollrate**
Return Type Long
Range N/A
Default 10 ms
Example ACRBitStatus1.Pollrate=100 '100 milliseconds

TrueColor

Description OLE_Color data type has to be set for this property. If the bit is set to true then the LED color is TrueColor.

Property **TrueColor**
Return Type OLE_COLOR
Range N/A
Default VbGreen
Example ACRBitStatus1.TrueColor=VbGreen

FalseColor

Description	OLE_Color datatype has to be set for this property. If the bit is set to false then the LED color is FalseColor
Property	**FalseColor**
Return Type	OLE_COLOR
Range	N/A
Default	VbGrey
Example	ACRBitStatus1.FalseColor=VbRed ‘If Bit not set, LED color is Red

---

BitMask

Description	This is a variant value representing the mask of 32 bits. The control will show only the bits that are specified in the mask. Input should be Hexadecimal format.
Property	**BitMask**
Return Type	Variant
Range	0 - FFFFFFFF
Default	N/A
Example	ACRBitStatus1.BitMask = “FFFFFFFF” ‘it will show all 32 bits

---

BitMaskCSV

Description	This is a comma-separated value of string type. The control will show only the bits that are specified in the BitMaskCSV. Comma separated values can be any value between 0 and 31.
Property	**BitMaskCSV**
Return Type	String
Range	0-31
Default	N/A
Example	ACRBitStatus1.BitMaskCSV = “0,1,3” ‘it will show bit 0,1, and 3

AutoSize

Description	If the Autosize property is true then the size of the control has to be dynamically changed. This will be based on (i) Number of bits selected by masking the BitMask property, or (ii) Number of bits selected in BitMaskCSV property.
Property	**AutoSize**
Return Type	Boolean
Range	N/A
Default	TRUE
Example	ACRBitStatus1.AutoSize=True

ConnectionControl

Description	This property will hold the reference of the instance of connection control.
Property	**ConnectionControl**
Return Type	Object
Range	N/A
Default	N/A
Example	Set ACRBitStatus1.ConnectionControl=ACRConnection1

Methods

BitLabel

Description	Bit Label has to be captured from the user and has to be assigned to the BitSelect.
Signature	**BitLabel** (strBitName as String)
Return Type	N/A
Parameters	
strBitName	String value given by user.
Return	N/A
Example	ACRBitStatus1.BitSelect=516 Call ACRBitStatus1.BitLabel (“Bit Five One Six”)

SetBit

Description	This method will enable the corresponding bit of the property BitSelect.
Signature	**SetBit** ()
Return Type	N/A
Parameters	N/A
Return	This method will call the SetFlag() method to enable the bit.
Example	Call ACRBitStatus1.SetBit

ClearBit

Description	This method will disable the corresponding bit of the property BitSelect.
Signature	**ClearBit** ()
Return Type	N/A
Parameters	N/A
Return	This method will call the SetFlag() method to disable the bit.
Example	Call ACRBitStatus1.ClearBit

GetValue

Description This method has to get the status of the bitselect property.

Signature **GetValue** ()

Return Type Boolean

Parameters N/A

Return Returns TRUE if BitSelect bit is 1, and returns FALSE when the BitSelect bit is 0.

Example

```
Dim blnGetValue as Boolean
blnGetValue= ACRBitStatus1.GetValue()
```

IsFlagSet

Description Utility for identifying a bit in a 32-bit long.

Signature **IsFlagSet** (nFlgGrp as Long, nFlgNdx as Long)

Return Type Boolean

Parameters

nFlgGrp	A value of type Long containing flags (as bits.).
nFlgNdx	Index of the flag.

Return Returns TRUE if bit at nFlagNdx is 1, returns FALSE when the bit is 0.

Example

```
Dim rtnStat as Variant
Dim bit128 as Boolean
rtnStat = ACRBitStatus1.ConnectionControl
ObjStatus.GetACRCustom ("P4100")
bit128 = ACRBitStatus1.IsFlagSet (rtnStat(0), 0)
```

Events

DataChanged

Description This event will fire when the data requested is changed.

Signature **DataChanged ()**

Return Type N/A

Parameters N/A

Return N/A

Example

```
Private Sub ACRBitStatus1_DataChanged
MsgBox  “Data Changed”
End sub
```

Numeric Control Properties, Methods and Events

Numeric Status control will get an input parameter from the user (ParameterSelect property has to be used) and it retrieves an array of 8 parameters, which are belongs to same group, from ACR controller.

Numeric status control contains following Properties, Methods and Events.

Properties

- **Long** **ParameterSelect**
- **Long** **ParameterPlacement**
- **Long** **IndexMask**
- **Long** **PollRate**
- **Boolean** **Autosize**
- **String** **IndexMaskCSV**
- **Object** **ConnectionControl**

Methods

- **Void** **ParameterLabel (bstrParamName As String)**
- **Void** **SetLong (nPparm as Long, nValue as Long, bFast as Boolean)**
- **Void** **SetReal (nPparm as Long, fValue as Double, bFast as Boolean)**
- **Void** **SetGlobal (Card as Long, nGlobal as Long, dValue as Double, bFast as Boolean)**
- **Variant** **GetValue ()**
- **Long** **GetParamType (nParameter as Long)**
- **Boolean** **GetParamInfo (nParameter as Long, nType as Long, nCode as Long, nIndex as Long, bstrCatagory as String, bstrDesc as String)**
- **Long** **GetParamAddr (nParameter as Long)**
- **Long** **GetLocalAddr (nProg as Long, nType as Long, nSize as Long)**
- **Long** **GetLocalArrayAddress (nProg as Long, nType as Long, nArray as Long, nSize as Long)**
- **VariantArray** **GetStatus (nMsgID as Long)**
- **Variant** **GetACRMemory (nType as Long, nAddress as Long, nCount as Long)**
- **VariantArray** **GetACRGroup (bstrRequest as String)**
- **VariantArray** **GetACRCustom (bstrRequest as String)**
- **Variant** **GetACRGroupRaw (nType as Long, nCode as Long, nIndex as Long)**
- **Void** **SetACRMemory (nType as Long, nAddress as Long, Values as Variant)**
- **Void** **SetACRMemoryMask (nAddress as Long, nNAND as Long, nOR as Long)**
- **Void** **SetParamLongMask (nPparm as Long, nNAND as Long, nOR as Long)**
- **Void** **InitPerformance ()**

- **Void** **GetPerformance ()**
- **Long** **AddACRGroup (bstrRequest as String)**
- **Long** **AddACRGroupRaw (nType as Long, nCode as Long, nIndex as Long)**
- **Long** **AddACRCustom (bstrRequest as String)**
- **Long** **AddACRMemory (nType as Long, nAddress as Long, nCount as Long)**
- **Void** **DelStatus (nMsgID as Long)**

Events

- **Void** **DataChanged ()**

Properties

ParameterSelect

Description This property represents the parameter selected.

Property **ParameterSelect**
Return Type Long
Range N/A
Default N/A
Example ACRNumericStatus1.ParameterSelect=6916

ParameterPlacement

Description This property represents the index of the array of 8 parameters that the parameter selected by ParameterSelect resides.

Property **ParameterPlacement**
Return Type Long
Range N/A
Default N/A
Example
```
Dim lngPlacement as Long
ACRNumericStatus1.ParameterSelect=6916
lngPlacement=ACRNumericStatus1.ParameterPlacement
'It gives 5 as result
```

IndexMask

Description This property represents the mask of parameters.

Property **IndexMask**
Return Type Long
Range N/A
Default N/A
Example
```
ACRNumericStatus1.IndexMask=255
'It shows value for 8 Parameters
ACRNumericStatus1.IndexMask=8
'It shows value for 3rd Parameter
```

PollRate

Description This property indicating the period in milliseconds to poll for the status. This value must be set assigned to ACRConnection1.ObjStatus.nStatusWaitRate before calling Connect method. ACRConnection1 is the name of connection control in the form.

Property **PollRate**

Return Type	Long
Range	N/A
Default	10 ms
Example	ACRNumericStatus1.PollRate=100

Autosize

Description	If true the control will be automatically resized based on the number of parameters selected with the IndexMask or IndexMaskCSV.
Property	**AutoSize**
Return Type	Boolean
Range	N/A
Default	TRUE
Example	ACRNumericStatus1.AutoSize=True

IndexMaskCSV

Description	This property represents the mask of parameters with comma separated.
Property	**IndexMaskCSV**
Return Type	String
Range	0-7
Default	N/A
Example	ACRNumericStatus1.IndexMaskCSV=”1,2,3”

ConnectionControl

Description	This property will hold the reference of the instance of connection control.
Property	**ConnectionControl**
Return Type	Object
Range	N/A
Default	N/A
Example	Set ACRNumericStatus1.ConnectionControl=ACRConnection1 ‘ACRConnection1 is the name of Connection Control in the ‘Current form

Methods

ParameterLabel

Description Method used to set the label for a parameter.

Signature **ParameterLabel** (bstrParamName as String)

Return Type N/A

Parameters

bstrParamName Holds the label for a parameter

Return N/A

Example Call ACRNumericStatus1.ParameterLabel (“Six Nine One Six”)

SetLong

Description Changes the value of a specific p-Parameter of type Long.

Signature **SetLong** (nPparm as Long, nValue as Long, bFast as Boolean)

Return Type N/A

Parameters

nPparm p-Parameter number that is to be changed

nValue Value to assign p-Parameter.

bFast How to send the command:

Value	Description
TRUE	Binary
FALSE	ASCII

Return N/A

Example Call ACRNumericStatus1.SetLong (6916, 10, True)

SetReal

Description Changes the value of a specific p-Parameter of type double.

Signature **SetReal** (nPparm as Long, fValue as Double, bFast as Boolean)

Return Type N/A

Parameters

nPparm p-Parameter number that is to be changed

fValue Value to assign p-Parameter.

bFast How to send the command:

Value	Description
TRUE	Binary
FALSE	ASCII

Return N/A

Example Call ACRNumericStatus1.SetReal (13370, 1.02, True)

SetGlobal

Description Changes the value of a specific, pre-dimensioned global parameter.

Signature **SetGlobal** (nCard as Long, nGlobal as Long, dValue as Double, bFast as Boolean)

Return Type N/A

Parameters

ncard Code value for type of card. This information is needed if using a binary command (bFast=TRUE) to find the memory address. Use zero if using ASCII (bFast=FALSE).

Device Types	
Value	**Controller**
0	ACR1200
1	ACR1500
2	ACR2000
3	ACR8010
4	ACR8020
5	ACR8020 (16-axis)
6	ACR1505
7	Reserved

8	ACR9000

nGlobal Global p-Parameter number that is to be changed.

dValue Value to assign p-Parameter.

bFast How to send the command:

Value	Description
TRUE	Binary
FALSE	ASCII

Return N/A

Example Call ACRNumericStatus1.SetGlobal (0,4095,100,false)

GetValue

Description Returns value for ParameterSelect.

Signature **GetValue** ()

Return Type Variant

Parameters N/A

Return Returns the value of the selected parameter through the property ParameterSelect.

Example
ACRNumericStatus1.ParameterSelect=6916
Call ACRNumericStatus1.GetValue

GetParamType

Description A wrapper method used to get the data type of a parameter.

Signature **GetParamType** (nParameter as Long)

Return Type Long

Parameters

nParameter Holds the p-Parameter number whose data type has to be returned.

Return Returns the data type of the p-Parameter.

Value	Description
0	Long
1	Double
2	Single

Example

```
Dim lngParamType as Long
lngParamType = ACRNumericStatus1.GetParamType (6916)
```

GetParamInfo

Description A wrapper method used to get specific information on a parameter.

Signature **GetParamInfo** (nParameter as Long, nType as Long, nCode as Long, nIndex as Long, bstrCategory as String, bstrDesc as String)

Return Type Boolean

Parameters

nParameter A numeric p-Parameter.

nType The data type of the values being read.

Value	Description
0	Long
1	Double
2	Single

nCode The ACR Group Code as documented in the ACR-View online help.

nIndex The ACR Group Index as documented in the ACR-View online help.

bstrCategory A textual description of the category a p-Parameter is in.

bstrDesc A textual description of the p-Parameter.

Return Returns TRUE if p-Parameter found.

Example

```
Dim lngParam as Long
Dim blnParam as Boolean
Dim lngType as Long
Dim lngCode as Long
Dim lngIndex as Long
Dim strCat as String
Dim strDesc as String
blnParam=ACRNumericStatus1.GetParamInfo (lngParam, lngType,
lngCode, lngIndex, strCat, strDesc)
```

GetParamAddr

Description A wrapper method used to retrieve the memory address location of a parameter.

Signature **GetParamAddr** (nParameter as Long)

Return Type Long

Parameters

nParameter A numeric p-Parameter.

Return Returns the address of the p-Parameter.

Example ACRNumericStatus1.GetParamAddr (6916)

GetLocalAddr

Description A wrapper method used to retrieve the memory address of a local variable in a specific program.

Signature **GetLocalAddr** (nProg as Long, nType as Long, nSize as Long)

Return Type Long

Parameters

nProg Provide the program number local variables are dimensioned in a program space.

nType The data type of the values being read:

Value	Description
0	Long
1	Double
2	Single

nSize After the call, this parameter holds the number of dimensioned variable available.

Return The return value is a valid ACR memory address (or zero if no memory is dimensioned for the requested variable type.)

Example

```
Dim lngSize as Long
Dim lngLocalAddr as Long
lngLocalAddr = ACRNumericStatus1.GetLocalAddr (0,0,lngSize)
```

GetLocalArrayAddress

Description A wrapper method used to retrieve the memory address if a local variable array in a specific program.

Signature **GetLocalArrayAddress** (nProg as Long, nType as Long, nArray as Long, nSize as Long)

Return Type Long

Parameters

nProg — Provide the program number local variables are dimensioned in a program space.

nType — The data type of the values being read

Value	Description
0	Long
1	Double
2	Single

nArray — The specific array being looked for.

nSize — After the call, this parameter holds the number of dimensioned variables available.

Return The return value is a valid ACR memory address (or zero if no memory is dimensioned for the requested variable type.)

Example

```
Dim lngSize as Long
Dim lngValue as Long
lngValue = ACRNumericStatus1.GetLocalArrayAddress (0,0,0,lngSize)
```

GetStatus

Description A wrapper method used to retrieve the specific status information.

Signature **GetStatus** (nMsgID as Long)

Return Type Variant

Parameters

nMsgID — The key to a specific status request as returned by one of the Add routines.

Return The returned array can be any size. It holds the values in Variants, either type long or double.

Example

```
Dim msgid as Long
```

```
Dim vntValues as Variant
msgid = ACRNumericStatus1.AddACRGroup (“P6916”)
vntValues=ACRNumericStatus1. GetStatus (msgid)
```

GetACRMemory

Description A wrapper method used to retrieve values requested from the specific memory location.

Signature **GetACRMemory** (nType as Long, nAddress as Long, nCount as Long)

Return Type Variant

Parameters

nType The data type of the values being read

Value	Description
0	Long
1	Double
2	Single

nAddress The starting physical memory address.

nCount The number of values to read (starting at the memory location.) The values of each memory location will be placed in a corresponding position in the returned array.

Return The returned array can be of any size but is limited to a single data type.

Example

```
Dim vntACRMem as Variant
vntACRMem = ACRNumericStatus1. GetACRMemory (0,0,5)
```

GetACRGroup

Description A wrapper method used to get the requested parameter group.

Signature **GetACRGroup** (bstrRequest as String)

Return Type Variant

Parameters

bstrRequest String of up to 4 p-Parameters, comma delimited. These parameters are used to look up the group, which is then used to return the 8 p-Parameter values for each group. Any p-Parameter in a group can be used to identify a group. Up to 4 groups can be requested and any undocumented/reserved items in a group are returned as zero (for example P6144 would return 8 values starting with the encoder position for Axis0).

Return Return an array containing up to 32 Variants, each of which are of type long or double. Each p-Parameter in the request results in a group of 8 values of the same type.

Example
```
Dim vntValues() as Variant
vntValues=ACRNumericStatus1.GetACRGroup("6916")
```

GetACRCustom

Description A wrapper method used to get the requested Parameters.

Signature **GetACRCustom** (bstrRequest as String)

Return Type Variant

Parameters

bstrRequest — String of up to 32 p-Parameters, comma delimited. These parameters are used to look up the individual, or custom, p- Parameter values (for example P6144, P6160 would return the encoder positions for Axis0 and Axis1).

Return The **GetACRCustom** method Return an array of up to 32 Variants (return type: long or double). Each p-Parameter in the request Return the values of the type as defined in the Parameters Reference section of the ACR User's Guide-View online help.

Example
```
Dim vntValues () as Variant
vntValues = ACRNumericStatus1. GetACRCustom ("6916")
```

GetACRGroupRaw

Description A wrapper method used to get the requested parameter group from the index and code.

Signature **GetACRGroupRaw** (nType as Long, nCode as Long, nIndex as Long)

Return Type Variant

Parameters

nType — The data type of the values being read:

Value	Description
0	Long
1	Double

2	Single

nCode — The ACR Group Code as documented in the ACR-View online help.

nIndex — The ACR Group Index as documented in the ACR-View online help.

Return — Return an array containing up to 8 Variants, all of which are of type long or double.

Example

```
Dim vntValues () as Variant
vntValues = ACRNumericStatus1.GetACRGroupRaw (“6916”)
```

SetACRMemory

Description — A wrapper method used to changes the value at a specific memory location

Signature — **SetACRMemory** (nType as Long, nAddress as Long, Values as Variant)

Return Type — N/A

Parameters

nType — The data type of the values being read:

Value	Description
0	Long
1	Double
2	Single

nAddress — The starting physical memory address on the ACR product.

Values — The data to be placed in memory starting at the address.

Return — N/A

Example

```
Dim vntValues () as Variant
vntValues (0) = 10
vntValues (1) = 20
vntValues (2) = 30
Call ACRNumericStatus1. SetACRMemory (0, 0, vntvalues)
```

SetACRMemoryMask

Description A wrapper method used to change the bit value(s) of a specific memory location.

Signature **SetACRMemoryMask** (nAddress as Long, nNAND as Long, nOR as Long)

Return Type N/A

Parameters

nAddress	The starting physical memory address. This address must point to a variable of type long for the mask to properly work.
nNAND	Used to clear bits.
nOR	Used to set bits.

Return N/A

Example ACRNumericStatus1.SetACRMemoryMask (0, 10, 5)

SetParamLongMask

Description A wrapper method used to change the long value through bitwise operations.

Signature **SetParamLongMask** (nPparm as Long, nNAND as Long, nOR as Long)

Return Type N/A

Parameters

nPparm	The parameter on the ACR product. This address must point to a variable of type long for the mask to properly work.
nNAND	Used to clear bits.
nOR	Used to set bits.

Return N/A

Example ACRNumericStatus1.SetParamLongMask (6916, 10, 5)

InitPerformance

Description A wrapper method used to initialize the performance counter of the ISA card to zero.

Signature **InitPerformance** ()

Return Type N/A

Parameters N/A

Return N/A

Example Call ACRNumericStatus1.InitPerformance

GetPerformance

Description A wrapper method used to retrieve the performance data of the ISA Cards.

Signature **GetPerformance** ()

Return Type N/A

Parameters N/A

Return N/A

Example ACRNumericStatus1.GetPerformance

AddACRGroup

Description Method to add a group request into the status queue.

Signature **AddACRGroup** (bstrRequest as String)

Return Type Long

Parameters

bstrRequest	String of up to 4 p-Parameters, comma delimited. These parameters are used to look up the group, which is then used to return the 8 p-Parameter values for each group. Any p-Parameter in a group can be used to identify a group. Up to 4 groups can be requested and any undocumented/reserved items in a group are returned as zero (for example P6144 would return 8 values starting with the encoder position for Axis0).

Return A key identifying the request in the queue. The key can be used to retrieve data using **GetStatus ()** (for example, when the alert is signaled).

Example

```
Dim lngID as Long
lngID = ACRNumericStatus1.AddACRGroup (“P6916”)
```

AddACRGroupRaw

Description Method to Add a group request into the status queue.

Signature **AddACRGroupRaw** (nType as Long, nCode as Long, nIndex as Long)

Return Type Long

Parameters

nType The data type of the values being read:

Value	Description
0	Long
1	Double
2	Single

nCode The ACR Group Code as documented in the ACR-View online help.

nIndex The ACR Group Index as documented in the ACR-View online help.

Return A key identifying the request in the queue. The key can be used to retrieve data using **GetStatus ().**
(for example, when the alert is signaled).

Example

```
Dim lngID as Long
lngID=ACRNumericStatus1.AddACRGroupRaw (0, 0, 0)
```

AddACRCustom

Description Add a custom p-Parameter request into the status queue.

Signature **AddACRCustom** (bstrRequest as String)

Return Type Long

Parameters

bstrRequest — String of up to 32 p-Parameters, comma delimited. These parameters are used to look up individual or custom p-Parameter values (for example P6144, P6160 would return the encoder positions for Axis0 and Axis1).

Return — A key identifying the request in the queue. The key can be used to retrieve data using **GetStatus ()** (for example, when the alert is signaled).

Example

```
Dim lngID as Long
lngID=ACRNumericStatus1.AddACRCustom (“P6916”)
```

AddACRMemory

Description — Add a memory value request into the status queue.

Signature — **AddACRMemory** (nType as Long, nAddress as Long, nCount as Long)

Return Type — Long

Parameters

nType — The data type of the values being read:

Value	Description
0	Long
1	Double
2	Single

nAddress — The starting physical memory address on the ACR product.

nCount — The number of values to read (starting at the memory location.) The values of each memory location will be placed in a corresponding position in the returned array.

Return — A key identifying the request in the queue. The key can be used to retrieve data using **GetStatus ()** (for example, when the alert is signaled).

Example

```
Dim lngID as Long
lngID = ACRNumericStatus1.AddACRMemory (0, 0, 2)
```

DelStatus

Description — Delete a status request from the status queue.

Signature — **DelStatus** (nMsgID as Long)

Return Type N/A

Parameters

nMsgID — The key to a specific status request as returned by one of the Add routines.

Return N/A

Example

```
Dim msgid as Long
msgid = ACRNumericStatus1.AddACRGroup("P6916")
Call ACRNumericStatus1.DelStatus (msgid)
```

Events

DataChanged

Description This event will fire when the data requested has changed.

Signature **DataChanged** ()

Return Type N/A

Parameters N/A

Return N/A

Example

```
Private Sub ACRNumericStatus1_DataChanged
MsgBox  "Data Changed"
End sub
```

Moves Control Properties and Methods

Moves control doesn't have a dialog and primarily will be a wrapper for properties and methods of the ComACRSrvr.dll. With this control we can perform movement in the motor one is linear move and the other is arc move.

Moves Control contains the following Properties and Methods.

Properties

- **Long** **MoveProfile**
- **Double** **MoveVel**
- **Double** **MoveFVel**
- **Double** **MoveAcc**
- **Long** **MoveMode**
- **Boolean** **MoveAbsolute**
- **Long** **MoveCounter**
- **Long** **ArcMode**
- **Boolean** **ArcAbsolute**
- **Boolean** **ArcCCW**
- **Object** **ConnectionControl**

Methods

- **Void** **Moves (nmask As Long, targets As Variant)**
- **Void** **MoveBatch (nmask As Long, movement As Variant)**
- **Void** **Arc (nmask As Long, targets As Variant)**
- **Void** **MoveStop (bdecel As Boolean)**
- **Void** **SendRes (nmask As Long)**
- **Void** **GetMoveCounter (nCounter As Long, nIncrement As Long)**
- **Void** **SetMoveCounter (nCounter As Long, nIncrement As Long)**

Properties

MoveProfile

Description This property specifies the master profile for use to move. Wraps the ComACRsrvr property nMoveProfile.

Property **MoveProfile**
Return Type Long
Range 0-15
Default 0
Example ACRMoves1.MoveProfile = 1

MoveVel

Description This property sets the linear velocity for the next move. Wraps the ComACRsrvr property fMoveVel.

Property **MoveVel**
Return Type Double
Range N/A
Default -1 (Use Preset Velocity)
Example ACRMoves1.MoveVel = 11.25

MoveFVel

Description This property sets the final velocity for the next move. Wraps the ComACRsrvr property fMoveFVel.

Property **MoveFVel**
Return Type Double
Range N/A
Default -1 (Use Preset Velocity)
Example ACRMoves1.MoveFVel = 12.25

MoveAcc

Description This property sets the Acceleration to be used with a linear move. Wraps the ComACRsrvr property fMoveAcc.

Property **MoveAcc**
Return Type Double
Range N/A
Default -1 (Use Preset Velocity)
Example ACRMoves1.MoveAcc = 7.50

MoveMode

Description	This property sets the mode of movement. Wraps the ComACRsrvr property nMoveMode.
Property	**MoveMode**
Return Type	Long
Range	0-3
Default	2
Example	ACRMoves1.MoveMode = 2

MoveAbsolute

Description	This property sets the absolute co-ordinate system or incremental co-ordinate system. Wraps the ComACRsrvr property bMoveAbsolute.
Property	**MoveAbsolute**
Return Type	Boolean
Range	N/A
Default	TRUE (Absolute Moves)
Example	ACRMoves1.MoveAbsolute = True

MoveCounter

Description	This property enables the move counter. Wraps the ComACRsrvr property nMoveCounter.
Property	**MoveCounter**
Return Type	Long
Range	-1, 0 and 1
Default	1 (ON, Count UP)
Example	ACRMoves1.MoveCounter = 1

ArcMode

Description	This property determines primary and secondary axes when performing the arc move. Wraps the ComACRsrvr property nArcMode.
Property	**ArcMode**
Return Type	Long
Range	0-3
Default	0
Example	ACRMoves1.ArcMode = 1

ArcAbsolute

Description	This property determines whether the arc centers are treated in Absolute terms or in relative terms. Wraps the ComACRsrvr property bArcAbsolute.
Property	**ArcAbsolute**
Return Type	Boolean
Range	N/A
Default	TRUE (Arc Absolute Centers)
Example	ACRMoves1.ArcAbsolute = True

ArcCCW

Description	This property determines the direction of the arc move. Wraps the ComACRsrvr property bArcCCW.
Property	**ArcCCW**
Return Type	Boolean
Range	N/A
Default	TRUE (CCW)
Example	ACRMoves1.ArcCCW = True

Connectioncontrol

Description	This property will hold the reference of the instance of connection control.
Property	**Connectioncontrol**
Return Type	Object
Range	N/A
Default	N/A
Example	Set ACRMoves1.ConnectionControl=ACRConnection1

Methods

Moves

Description This wrapper method will generate the move.

Signature **Moves** (nMask As Long, targets As Variant)

Return Type N/A

Parameters

nMask	Specifies which axes to use for move.
targets	The target position information for each specified axis.

Return N/A

Example

```
Dim nMask As Long
Dim targets (0 To 1) As Variant
targets (0) = 150
targets (1) = 150
nMask = 2
ACRMoves1.Moves (nMask, targets)
```

MoveBatch

Description This wrapper method will send a set of fully defined moves for batch processing.

Signature **MoveBatch** (nmask As Long, movement As Variant)

Return Type N/A

Parameters

nMask	The data type of values being read.
Movement	Data required for completing any number of moves.

Return N/A

Example

```
Dim nMask as long
Dim movement (0 To 31) As Variant
nMask = 0
movement (0) = 136
movement (1) = 0
movement (2) = 3
movement (3) = 3
movement (4) = 0
```

```
movement (5) = 0
movement (6) = 0
movement (7) = 0
movement (8) = 0#
movement (9) = 0#
movement (10) = 0#
movement (11) = 15
movement (12) = 15
movement (13) = 0
movement (14) = 0
movement (15) = 0
movement (16) = 0
movement (17) = 0
movement (18) = 0
movement (19) = 0
movement (20) = 0
movement (21) = 0
movement (22) = 0
movement (23) = 0
movement (24) = 0
movement (25) = 0
movement (26) = 0
movement (27) = 0
movement (28) = 0
movement (29) = 0#
movement (30) = 0#
movement (31) = 0#
ACRMoves1.MoveBatch (nMask, fValue)
```

Arc

Description This wrapper method will generate an Arc move.

Signature **Arc** (nMask As Long, targets As Variant)

Return Type N/A

Parameters

nMask	Specifies which axes to use for move.
targets	The arc centers the target position information for each axis.

Return N/A

Example

```
Dim nmask As Long
Dim targets (0 To 3) As Variant
targets (0) = 5
targets (1) = 0
targets (2) = 10
targets (3) = 0
nMask = 2
ACRMoves1.Arc (nMask, targets)
```

MoveStop

Description This wrapper method stops the commanded motion.

Signature **MoveStop** (bDecel as Boolean)

Return Type N/A

Parameters

bDecel Determines how to stop the motion.

Value	Description
TRUE	Stop All Moves
FALSE	Kill All Moves

Return When the bDecel parameter is TRUE, a Stop All Moves flag is set using the binary command.
When the bDecel parameter is FALSE, a Kill All Moves flag is set using binary.

Example

```
Dim bDecel as Boolean
bDecel = TRUE
ACRMoves1.MoveStop (bDecel)
```

SendRes

Description This wrapper method resets the position counters of the ACR Controller.

Signature **SendRes** (nMask as Long)

Return Type N/A

Parameters

nMask Specifies which axes to apply the RES.

Return

Example

```
Dim nMask As Long
nMask = 2
ACRMoves1.SendRes (nMask)
```

GetMoveCounter

Description This wrapper method retrieves the move counter from the ACR Controller.

Signature **GetMoveCounter** (nCounter As Long, nIncrement As Long)

Return Type N/A

Parameters

nCounter	The index value of the move currently active on the controller.
nIncrement	The step used to increment the nCounter.

Return The nCounter and nIncrement are both updated after calling this method.

Example

```
Dim nCounter as long
Dim nIncrement as long
nCounter = 0
nIncrement = 0
ACRMoves1.GetMoveCounter (nCounter, nIncrement)
```

SetMoveCounter

Description This wrapper method sets the move counter of the ACR Controller.

Signature **SetMoveCounter** (nCounter As Long, nIncrement As Long)

Return Type N/A

Parameters

nCounter	The index value of the move currently active on the controller.
nIncrement	The step used to increment the nCounter.

Return N/A

Example

```
Dim nCounter as long
Dim nIncrement as long
nCounter = 2
nIncrement = 2
ACRMoves1.SetMoveCounter (nCounter, nIncrement)
```

Feedrate Control Properties and Methods

The Feedrate control is used to manipulate the move by either increasing or decreasing the speed of the motor. The motor can also be paused while the motor is in motion and can also be unpaused.

Feedrate Control contains the following Properties and Methods.

Properties

- **Long** **MotionProfile**
- **Double** **FOV**
- **Double** **MinFOV**
- **Double** **MaxFOV**
- **Object** **ConnectionControl**

Methods

- **Void** **SetFOV (nMask As Long, fValue As Double)**
- **Void** **SetROV (nMask As Long, fValue As Double)**

Properties

MotionProfile

Description This property holds the Master number that the control will be using.

Property **MotionProfile**
Return Type Long
Range 0 – 15
Default 0
Example ACRFeedRate1.MotionProfile = 0

FOV

Description This property holds the Feedrate override value for the current Master profile. This value will either increase or decrease the speed of the motor. FOV should always be within MinFov and MaxFov.

Property **FOV**
Return Type Double
Range N/A
Default 1
Example ACRFeedRate1.FOV = 1.25

MinFOV

Description	This property holds the Minimum Feedrate override value for the current Master profile. This property keeps check on the lower boundary of FOV.
Property	**MinFOV**
Return Type	Double
Range	N/A
Default	1
Example	ACRFeedRate1.MinFov = 1.0

MaxFOV

Description	This property holds the Maximum Feedrate override value for the current Master profile. This property keeps check on the upper boundary of FOV.
Property	**MaxFOV**
Return Type	Double
Range	N/A
Default	100
Example	ACRFeedRate1.MaxFov = 7.50

ConnectionControl

Description	This property will hold the reference of the instance of connection control.
Property	**Connectioncontrol**
Return Type	Object
Range	N/A
Default	N/A
Example	Set ACRFeedRate1.ConnectionControl=ACRConnection1

Methods

SetFOV

Description This wrapper method will set the Feedrate override value for the current master profile.

Signature **SetFOV** (nMask As Long, fValue As Double)

Return Type N/A

Parameters

nMask Specifies which axes to use for setting the FOV

fValue Set the FOV value for all the specified axes in nMask.

Return N/A

Example

```
Dim nMask as long
Dim fValue as double
nMask = 3
fValue = 1.25
ACRFeedRate1.SetFOV (nMask, fValue)
```

SetROV

Description This wrapper method will set the Rapid Feedrate override value for the current master profile.

Signature **SetROV** (nMask As Long, fValue As Double)

Return Type N/A

Parameters

nMask Specifies which axes to use for setting the ROV

fValue Set the ROV value for all the specified axes in nMask.

Return N/A

Example

```
Dim nMask as long
Dim fValue as double
nMask = 3
fValue = 1.25
ACRFeedRate1.SetROV (nMask, fValue)
```

CANOpen Control Properties and Methods

The CANOpen control is used to configure and start the CANOpen I/O Network based on the value of Master Node ID, Bit Rate, Cyclic period and Number of Slave nodes set by the User.

CANOpen control contains following Properties, and Methods.

Properties

- **Long** **MasterNodeID**
- **Long** **BitRate**
- **Long** **CyclicPeriod**
- **Long** **NumSlaveNodes**
- **Object** **ConnectionControl**

Methods

- **Void** **CalcBitRate ()**
- **Void** **CalcCyclicPeriod ()**
- **Void** **SetSlaveNodeID (slavenodes as Integer)**
- **Integer** **GetSlaveNodeID ()**
- **Void** **StartCANOpen ()**
- **Void** **ResetCANOpen ()**
- **Void** **GetCANOpenStatus ()**

Properties

MasterNodeID

Description Represents the MasterNodeID of the CANOpen network.

Property **MasterNodeId**
Return Type Long
Range 1 - 127
Default 5
Example ACRCANOpen1.MasterNodeId =5

BitRate

Description Represents the Bit Rate for the CANOpen network in kbps.

Property **BitRate**
Return Type Long
Range 10,20,50,125,250,500,800,1000
Default 125
Example ACRCANOpen1.BitRate =125

CyclicPeriod

Description Represents the Cyclic period of the CANOpen network (in ms).

Property **CyclicPeriod**
Return Type Long
Range N/A
Default 50
Example ACRCANOpen1.CyclicPeriod =50

NumSlaveNodes

Description Holds the value of number of slave nodes on the Network (Range 0- 4).

Property **NumSlaveNodes**
Return Type Long
Range 0-4
Default 1
Example ACRCANOpen1.NumSlaveNodes =3

ConnectionControl

Description	This property will hold the reference of the instance of connection control.
Property	**Connectioncontrol**
Return Type	Object
Range	N/A
Default	N/A
Example	Set ACRCANOpen1.ConnectionControl=ACRConnection1 ‘ACRConnection1 is the name of Connection Control in the ‘Current form

Methods

CalcBitRate

Description Displays the BitRate Table selection dialog.

Signature **CalcBitRate ()**

Return Type N/A

Parameters N/A

Return N/A

Example Call ACRCANOpen1.CalcBitRate

CalcCyclicPeriod

Description Displays the CalcCyclicPeriod Table calculator dialog.

Signature **CalcCyclicperiod ()**

Return Type N/A

Parameters N/A

Return N/A

Example Call ACRCANOpen1.CalcCyclicperiod

SetSlaveNodeID

Description Sends an array of integers that represents the NodeID for each of the Slave Nodes.

Signature **SetSlaveNodeID** (slavenodes as Integer)

Return Type N/A

Parameters

Slavenodes	It is an integer array those values are to be set to slave nodes.

Return N/A

Example

```
Dim slavenodes (3) as Integer
Slavenodes (0)=3
Slavenodes (1)=6
Slavenodes (2)=7
Slavenodes (3)=9
Call ACRCANOpen1.SetSlaveNodeID (slavenodes)
```

GetSlaveNodeID

Description Retrieves an array of integers that represents NodeID for each of the Slave Nodes.

Signature **GetSlaveNodeID ()**

Return Type Integer

Parameters N/A

Return This method returns integer array that represents all slave node ids.

Example

```
Dim I () as Integer
I =ACRCANOpen1.GetSlaveNodeID
```

StartCANOpen

Description This method activates the CANOpen Network.

Signature **StartCANOpen ()**

Return Type N/A

Parameters N/A

Return N/A

Example Call ACRCANOpen1.StartCANOpen

ResetCANOpen

Description Method is used to stop and reset the CANOpen Network.

Signature **ResetCANOpen ()**

Return Type N/A

Parameters N/A

Return N/A

Example Call ACRCANOpen1.ResetCANOpen

GetCANOpenStatus

Description	This method is used to request and retrieve the status of the CANOpen Network and displays in the Display textbox.
Signature	**GetCANOpenStatus ()**
Return Type	N/A
Parameters	N/A
Return	N/A
Example	Call ACRCANOpen1.GetCANOpenStatus ()

TeachPanel Control Properties and Methods

The TeachPanel control is used to play movements in the controller and record the same. Jog Neg and Jog Pos are the 2 different movements, which can be handled, in three different modes. Continuous, Incremental and Whileheld are the three modes. While the drive is rotating we can capture the position and store it in an array. The captured array can be cleared or stored in a CSV file. Through the button named Enable we can enable the drive and with the button named EStop we can stop the motor while in motion.

TeachPanel Control contains the following Properties and Methods.

Properties

- **Double ActualPos**
- **Long Axis**
- **Boolean DriveEnable**
- **Boolean DriveFault**
- **Boolean PosLimit**
- **Boolean NegLimit**
- **Boolean Home**
- **Long TeachArrayIndex**
- **Double Velocity**
- **Double Acceleration**
- **Double TargetPosition**
- **Long PPU**
- **Long JogMode**
- **Boolean HomeDirectionPositive**
- **Boolean DisableDriveOnEStop**
- **Long Pollrate**
- **Object ConnectionControl**

Methods

- **Void JogNeg ()**
- **Void JogPos ()**
- **Void JogStop ()**
- **Void EnableDrive (DriveStatus as string)**
- **Void KillAllMotion (StopStatus as string)**
- **Void HomePos (Direction as Long)**
- **Void ClearPos ()**
- **Void CapturePos ()**
- **Void ClearArray ()**
- **Void SaveArray ()**

Properties

ActualPos

Description This is a read only property, which holds the current position of the motor.

Property **ActualPos**
Return Type Double
Range N/A
Default N/A
Example
```
Dim actPosition as Double
actPosition = ACRTeachPanel1.ActualPos
```

Axis

Description This property holds the value of the Axis.

Property **Axis**
Return Type Long
Range 0-15
Default 0
Example
```
ACRTeachPanel1.Axis = 7
```

DriveEnable

Description This is a read only property, which stores the drive enable status in this property. If the value of this property is true then it represents drive is enabled, and if the value is false it represents drive is disabled.

Property **DriveEnable**
Return Type Boolean
Range N/A
Default N/A
Example
```
Dim bdriveEnable as Boolean
bdriveEnable = ACRTeachPanel1.DriveEnable
```

DriveFault

Description Drive fault will be stored in this read only property. If the value of this property is true then it represents drive is disabled, and if the value is false it represents drive is enabled.

Property **DriveFault**
Return Type Boolean
Range N/A
Default N/A
Example
```
Dim bdriveFault as Boolean
bdriveFault = ACRTeachPanel1.DriveFault
```

PosLimit

Description Maximum positive limit status will be stored in this read only property. True represents drive has reached positive limit, false represents drive has not yet reached the positive limit.

Property **PosLimit**
Return Type Boolean
Range N/A
Default N/A
Example
```
Dim bPosLimit as Boolean
bPosLimit = ACRTeachPanel1.PosLimit
```

NegLimit

Description Maximum negative limit status will be stored in this read only property. True represents drive has reached negative limit, false represents drive has not yet reached the negative limit.

Property **NegLimit**
Return Type Boolean
Range N/A
Default N/A
Example
```
Dim bNegLimit as Boolean
bNegLimit = ACRTeachPanel1.NegLimit
```

Home

Description Read only property, which represents the home state of the drive. If true then the position of the drive is in home state.

Property **Home**
Return Type Boolean
Range N/A
Default N/A
Example
```
Dim bHome as Boolean
bHome = ACRTeachPanel1.Home
```

TeachArrayIndex

Description This is a read only property which holds the running index of the teach array index. For every click on the CapturePos the index will be increased by one. This will be zero when ClearPos is clicked.

Property **TeachArrayIndex**
Return Type Long
Range N/A
Default 0
Example
```
Dim lngTeachArrayIndex as Long
lngTeachArrayIndex = ACRTeachPanel1.TeachArrayIndex
```

Velocity

Description	This property is used to make the movement of the motor and its value should be greater than zero.
Property	**Velocity**
Return Type	Double
Range	N/A
Default	N/A
Example	ACRTeachPanel1.Velocity = 33.88

Acceleration

Description	This property increases the speed of the motor and its value should be greater than zero.
Property	**Acceleration**
Return Type	Double
Range	N/A
Default	N/A
Example	ACRTeachPanel1.Acceleration = 88.33

TargetPosition

Description	Property, which makes the jog movement, if JogPos is clicked then the movement will be in positive direction based on the Incremental distance text box. If JogNeg is clicked then the movement will be in negative direction based on the Incremental distance text box.
Property	**TargetPosition**
Return Type	Double
Range	N/A
Default	N/A
Example	ACRTeachPanel1.TargetPosition = 20.235

PPU

Description	Pulses Per Unit will be stored in this property.
Property	**PPU**
Return Type	Long
Range	N/A
Default	N/A
Example	ACRTeachPanel1.PPU = 8000

JogMode

Description Mode for the jog operation will be stored in this property.
0 – Continuous
1 – Whileheld
2 – Incremental

Property **JogMode**
Return Type Long
Range 0-2
Default 0
Example ACRTeachPanel1.Jogmode = 1

HomeDirectionPositive

Description Property that determines whether the motion should happen in the positive direction or in negative direction. If the Boolean value is true, then positive direction else negative direction.

Property **HomeDirectionPositive**
Return Type Boolean
Range N/A
Default TRUE
Example ACRTeachPanel1.HomeDirectionPositive = True

DisableDriveOnEStop

Description Property determines whether the drives should be disabled or not while stopping. If true the drives are disabled while pressing EStop button else drives will not be disabled.

Property **DisableDriveOnEStop**
Return Type Boolean
Range N/A
Default TRUE
Example ACRTeachPanel1.DisableDriveonEstop = True

Pollrate

Description Delay time between fetching values from the ACR Controller will be stored in this property.

Property **Pollrate**
Return Type Long
Range N/A
Default 10
Example ACRTeachPanel1.Pollrate = 10

Connectioncontrol

Description	This property will hold the reference of the instance of connection control.
Property	**Connectioncontrol**
Return Type	Object
Range	N/A
Default	N/A
Example	Set ACRTeachPanel1.ConnectionControl=ACRConnection1

Methods

JogNeg

Description	This method will make the jog movement in negative direction.
Signature	**JogNeg** ()
Return Type	N/A
Parameters	N/A
Return	N/A
Example	ACRTeachPanel1.JogNeg ()

JogPos

Description	This method will make the jog movement in positive direction.
Signature	**JogPos** ()
Return Type	N/A
Parameters	N/A
Return	N/A
Example	ACRTeachPanel1.JogPos ()

JogStop

Description	This method will stop the jog movement.
Signature	**JogStop** ()
Return Type	N/A
Parameters	N/A
Return	N/A
Example	ACRTeachPanel1.JogStop ()

EnableDrive

Description This method will either enable or disable the drive based on the value of the drivestatus string. If the value of Drivestatus string is “On” then the drive is enabled or if the drivestatus string is “Off” then it is disabled.

Signature **EnableDrive** (Drivestatus as string)

Return Type N/A

Parameters

Drivestatus	Sends the string either “ON” or “OFF”.

Return N/A

Example

```
Dim Drivestatus as string
Drivestatus = “ON”
ACRTeachPanel1.Enabledrive (Drivestatus)
```

KillAllMotion

Description This method will stop the motion of the motor by sending the values Ctrl + Z or Ctrl + X if the caption of the EStop button is “EStop”. If the button caption is “Clear EStop” then Ctrl + Y will be sent to the controller to clear the flags.

Signature **KillAllMotion** (Stopstatus as string)

Return Type N/A

Parameters

StopStatus	Sends the character value of Ctrl + Z, Ctrl + Y or Ctrl + X.

Return N/A

Example

```
Dim StopStatus as string
StopStatus = chr (24)
ACRTeachPanel1.KillAllMotion (StopStatus)
```

HomePos

Description	This method will make the jog movement either in Positive direction or in Negative direction based on the value of the property HomeDirectionPositive.
Signature	**HomePos** (Direction as Long)
Return Type	N/A
Parameters	This long variable holds either 1 or –1. Value 1 makes rotates the motor in positive direction and –1 rotates the motor in negative direction.
Return	N/A
Example	ACRTeachPanel1.HomePos ()

ClearPos

Description	This method will make the AcutalPos property to zero.
Signature	**ClearPos** ()
Return Type	N/A
Parameters	N/A
Return	N/A
Example	ACRTeachPanel1.ClearPos ()

CapturePos

Description	This method will capture the present ActualPos value in an array.
Signature	**CapturePos** ()
Return Type	N/A
Parameters	N/A
Return	N/A
Example	ACRTeachPanel1.CapturePos ()

ClearArray

Description	This method will clear the so far captured positions from the array.
Signature	**ClearArray** ()
Return Type	N/A
Parameters	N/A
Return	N/A
Example	ACRTeachPanel1.ClearArray ()

SaveArray

Description	This method will save the so far captured position into a CSV file.
Signature	**SaveArray** ()
Return Type	N/A
Parameters	N/A
Return	N/A
Example	ACRTeachPanel1.SaveArray ()

PlaybackPanel Control Properties and Methods

The Playback Panel control is used to play various teach point movements and also playback array movements. The movement can be paused and can be resumed. Through this control we can also perform row-by-row movement. This control saves the played points in a CSV file. Events can also be saved against the points in a separate grid. Velocity, Acceleration, Deceleration and StopRamp has to be provided for the movement of the motor.

Playback Panel Control contains the following Properties and Methods.

Properties

- **Long** **StartingIndex**
- **Long** **CurrentIndex**
- **Double** **Velocity**
- **Double** **Acceleration**
- **Double** **Deceleration**
- **Double** **StopRamp**
- **Long** **MasterProfile**
- **Object** **ConnectionControl**

Methods

- **Void** **ImportArray (strCSVFileName as String)**
- **Void** **ImportFromTeach (strCSVFileName as String)**
- **Void** **AddEvent ()**
- **Void** **Playback ()**
- **Void** **StepPlayback ()**
- **Void** **SaveArray (strCSVFileName as string)**
- **Void** **PausePlayback ()**
- **Void** **ResumePlayback ()**
- **Void** **StopPlayback ()**
- **String** **GetValue (lngRow As Long, lngCol As Long) As String**
- **Void** **SetValue (lngRow As Long, lngCol As Long, strValue As String)**

Properties

StartingIndex

Description This property holds the starting index value for the StepNext point movement. This property should be greater than zero. This property will be used only for the first time when StepPlayback is clicked.

Property **StartingIndex**
Return Type Long
Range N/A
Default N/A
Example ACRPlaybackPanel1.StartingIndex = 3

CurrentIndex

Description This read only property holds the index of the current row while playing. This property gets increased one by one while playing.

Property **CurrentIndex**
Return Type Long
Range N/A
Default 0
Example Dim lngCurrIndex as Long
ACRPlaybackPanel1.CurrentIndex = lngCurrIndex

Velocity

Description Velocity value will be stored in this property. Velocity should be greater than zero for the movement of the motor.

Property **Velocity**
Return Type Double
Range N/A
Default 0
Example ACRPlaybackPanel1.Velocity = 20

Acceleration

Description Acceleration value will be stored in this property. This property increases the speed of the motor.

Property **Acceleration**
Return Type Double
Range N/A
Default 0
Example ACRPlaybackPanel1.Acceleration = 20

Deceleration

Description Deceleration value will be stored in this property. This property decreases the speed of the motor.

Property **Deceleration**
Return Type Double
Range N/A
Default 0
Example ACRPlaybackPanel1.Deceleration = 20

StopRamp

Description	StopRamp value will be stored in this property. This property will be used to stop the motor while in motion.
Property	**StopRamp**
Return Type	Double
Range	N/A
Default	0
Example	ACRPlaybackPanel1.StopRamp = 20

MasterProfile

Description	MasterProfile value will be stored in this property. This represents which master is being used.
Property	**MasterProfile**
Return Type	Long
Range	0 – 15
Default	0
Example	ACRPlaybackPanel1.MasterProfile = 2

Connectioncontrol

Description	This property will hold the reference of the instance of connection control.
Property	**Connectioncontrol**
Return Type	Object
Range	N/A
Default	N/A
Example	Set ACRPlaybackPanel1.ConnectionControl = ACRConnection1

Methods

ImportArray

Description This method will load the CSV file into the Grid.

Signature **ImportArray** (strCSVFilename as string)

Return Type N/A

Parameters

strCSVFileName	Filename of the CSV File.

Return N/A

Example ACRPlaybackPanel1.ImportArray (strCSVFileName)

ImportFromTeach

Description This method will load the CSV file created through TeachPanel Control.

Signature **ImportFromTeach** (strCSVFileName As String)

Return Type N/A

Parameters

strCSVFileName	Filename of the CSV File.

Return N/A

Example
```
Dim strReadLine as string
strReadLine = “c:\Array.CSV”
ACRPlaybackPanel1.ImportFromTeach (strCsvFileName)
```

AddEvent

Description This method will add the command string and the added event will be played back.

Signature **AddEvent** ()

Return Type N/A

Parameters N/A

Return N/A

Example ACRPlaybackPanel1.AddEvent ()

Playback

Description This method will download the file to the controller and execute the playback points.

Signature **Playback** ()

Return Type N/A

Parameters N/A

Return N/A

Example ACRPlaybackPanel1.Playback ()

StepPlayback

Description This method will perform the playback for one step from the current Index.

Signature **StepPlayback** ()

Return Type N/A

Parameters N/A

Return N/A

Example ACRPlaybackPanel1.StepPlayback ()

SaveArray

Description This method will save the teach point array and the Event if available.

Signature **SaveArray** (strCSVFilename as string)

Return Type N/A

Parameters

strCSVFileName	Filename of the CSV file

Return N/A

Example Dim strCSVFileName as string

```
strCSVFileName = “c:\Array.csv”
ACRPlaybackPanel1.SaveArray (strCSVFileName)
```

PausePlayback

Description This method will pause the movement of the motor at the current index.

Signature **PausePlayback** ()

Return Type N/A

Parameters N/A

Return N/A

Example ACRPlaybackPanel1.PausePlayback ()

ResumePlayback

Description This method will resume the movement of the motor from the paused stage.

Signature **ResumePlayback** ()

Return Type N/A

Parameters N/A

Return N/A

Example ACRPlaybackPanel1.ResumePlayback ()

StopPlayback

Description This method will stop the movement of the motor at the current index.

Signature **StopPlayback** ()

Return Type N/A

Parameters N/A

Return N/A

Example ACRPlaybackPanel1.StopPlayback ()

GetValue

Description This function will retrieve the value from the Grid for the specified row and column.

Signature **GetValue** (lngRow As Long, lngCol As Long)

Return Type String

Parameters

lngRow	Row number has to be provided.
lngCol	Col number has to be provided.

Return This function returns the value of the cell specified in lngRow and lngCol.

Example

```
Dim lngRow as long, lngCol as long
lngRow = 2
lngCol = 2
ACRPlaybackPanel1.GetValue (lngRow, lngCol)
```

SetValue

Description This method will update the grid cell at the specified row and column with the specified value.

Signature **SetValue** (lngRow As Long, lngCol As Long, strValue As String)

Return Type N/A

Parameters

lngRow	Row number has to be provided.
lngCol	Col number has to be provided.
strValue	Value to be placed in the Grid cell.

Return N/A

Example

```
Dim lngRow as long, lngCol as long
Dim strValue as string
lngRow = 2
lngCol = 2
strValue = 23.55
ACRPlaybackPanel1.SetValue (lngRow, lngCol, strValue)
```

StatusPanel Control Properties Methods and Events

The Status Panel Control displays the status of the most commonly used bits and the numeric value of the P-parameters.

Status Panel Control contains following Properties, Methods, and Events.

Properties

- **Long** **PollRate**
- **Object** **ConnectionControl**

Methods

- **Boolean** **GetMasterBitStatus (Row as Integer, Col as Integer)**
- **Boolean** **GetAxisBitStatus (Row as Integer, Col as Integer)**
- **Variant** **GetMasterNumericStatus (Row as Integer, Col as Integer)**
- **Variant** **GetAxisNumericStatus (Row as Integer, Col as Integer)**

Events

- **Void** **DataChanged ()**

Properties

Pollrate

Description	This property indicates the period in milliseconds to poll for the status. This value must be set assigned to ACRConnection1.ObjStatus.nStatusWaitRate before calling Connect method.
Property	**Pollrate**
Return Type	Long
Range	N/A
Default	10
Example	ACRStatusPanel1.PollRate=100

ConnectionControl

Description	This property will hold the reference of the instance of connection control.
Property	**ConnectionControl**
Return Type	Object
Range	N/A
Default	N/A
Example	Set ACRStatusPanel1.ConnectionControl=ACRConnection1 ‘ACRConnection1 is the name of Connection Control in the ‘Current form

Methods

GetMasterBitStatus

Description Returns the value from the user interface for a particular label in master bit status display.

Signature **GetMasterBitStatus** (Row as Integer, Col as Integer)

Return Type Boolean

Parameters

Row — Holds the number of label in master bit status tab.
Range 0-7
0 Moving
1 Accelerating
2 Decelerating
3 Stopping
4 Kill All Moves Request
5 Program Running
6 Program Inhibited
7 Program Dwelling

Col — Holds the Master Number (Range 0- 7).

Return Return value TRUE represents the corresponding bit is glowing and value FALSE represents the corresponding bit is not glowing.

Example

```
Dim blnStatus as Boolean
‘ To get Moving status of Master 0
blnStatus=ACRStatusPanel1.GetMasterBitStatus (0, 0)
‘ To get Moving status of Master 1
blnStatus=ACRStatusPanel1.GetMasterBitStatus (0, 1)
```

GetAxisBitStatus

Description Returns the value from the user interface for a particular label in axis bit status display.

Signature **GetAxisBitStatus** (Row as Integer, Col as Integer)

Return Type Boolean

Parameters

Row — Holds the number of label in axis bit status tab.

Range 0-9.
0 Drive Enable
1 Drive Fault
2 Kill All Motion Request
3 Jog Active
4 Positive Hard Limit
5 Negative Hard Limit
6 Positive Soft Limit
7 Negative Soft Limit
8 Home Found
9 Maximum Position Error

Col — Holds the Axis Number (Range 0- 7).

Return — Return value TRUE represents the corresponding bit is glowing and value FALSE represents the corresponding bit is not glowing.

Example

```
Dim blnStatus as Boolean
‘ To get “Drive Enable” status for Axis 0
blnStatus=ACRStatusPanel1.GetAxisBitStatus (0, 0)
 ‘To get “Drive Enable” status for Axis 1
blnStatus=ACRStatusPanel1.GetAxisBitStatus (0, 1)
 ‘To get “Drive Fault” status for Axis 0
blnStatus=ACRStatusPanel1.GetAxisBitStatus (1, 0)
‘To get “Drive Fault” status for Axis 1
blnStatus=ACRStatusPanel1.GetAxisBitStatus (1, 1)
```

GetMasterNumericStatus

Description — Returns the value from the user interface for a particular label in master numeric status display.

Signature — **GetMasterNumericStatus** (Row as Integer, Col as Integer)

Return Type — Variant

Parameters

Row — Holds the number of label in master numeric status tab.
Range 0-4.
0 Vel
1 Acc
2 Distance Into Move
3 Distance To Go
4 Program Line Number

Col — Holds the Master Number (Range 0- 7).

Return — This returns the variant value of the corresponding cell from the grid.

Example

```
Dim vntValue as Variant
```

```
‘ To get “vel” value for axis0
vntValue =ACRStatusPanel1.GetMasterNumericStatus (0,0)
‘ To get “vel” value for axis1
vntValue =ACRStatusPanel1.GetMasterNumericStatus (0,1)
‘ To get “Acc” value for axis0
vntValue =ACRStatusPanel1.GetMasterNumericStatus (1,0)
‘To get “Acc” value for axis1
vntValue =ACRStatusPanel1.GetMasterNumericStatus (1,1)
```

GetAxisNumericStatus

Description Returns the value from the user interface for a particular label in axis numeric status display.

Signature **GetAxisNumericStatus** (Row as Integer, Col as Integer)

Return Type Variant

Parameters

Row — Holds the number of label in axis numeric status tab.
Range 0 - 7.

0	Current Position
1	Target Position
2	Actual Position
3	Jog Offset
4	Jog Vel Setting
5	Jog Vel Current
6	Jog Acc Setting
7	Jog Acc Current

Col — Holds the axis number (Range 0- 7).

Return This returns the variant value of the corresponding cell from the grid.

Example

```
Dim vntValue as variant

‘ To get “Current Position” value for Axis 0
vntValue=ACRStatusPanel1.GetAxisNumericStatus (0,0)

‘ To get “Current Position” value for Axis 1
vntValue=ACRStatusPanel1.GetAxisNumericStatus (0,1)

‘To get “Target Position” value for Axis 0
vntValue=ACRStatusPanel1.GetAxisNumericStatus (1,0)

‘To “Target Position” value for Axis 1
vntValue=ACRStatusPanel1.GetAxisNumericStatus (1,1)
```

Events

DataChanged

Description	This event will fire when the data requested has changed.
Signature	**DataChanged** ()
Return Type	N/A
Parameters	N/A
Return	N/A
Example	Private Sub ACRStatusPanel1_DataChanged () MsgBox “DataChanged” End sub

DriveTalk Control Properties and Methods

The DriveTalk Control is the primary mechanism for configuration of DriveTalk communication with DriveTalk enabled drives. This is also the main mechanism for retrieving Drive Status (bits and numeric) from these drives. Also this control allows for sending commands as well as configuration information to these drives.

This control is used to provide the following functionalities:

i) To configure Drive Talk communication with DriveTalk enabled drives.
ii) For retrieving status of the drive from DriveTalk enabled drives.
iii) To send commands and configuration information to DriveTalk enabled drives.

Properties

- **Long** **AxesMask**
- **Boolean** **EnableDriveTalk**
- **Double** **DriveDataMask**
- **Long** **DriveTalkMode**
- **Object** **ConnectionControl**

Methods

- **Void** **GetDriveDataRequest ()**
- **Void** **GetConfig ()**
- **Void** **SendConfig ()**
- **Void** **GetErrorLog ()**
- **Void** **SendASTFile (strASTFileName As String)**

Properties

AxesMask

Description Represents the mask value for number of axes (for Axis 0 –15), which are to be connected to the drive talk.

Property **AxesMask**
Return Type Long
Range 1 - 65535
Default N/A
Example
```
ACRDriveTalk1.AxesMask=1 ‘ Axis 0 to be connected
ACRDriveTalk1.AxesMask=3  ‘ Axis 0, Axis 1 are to be connected
ACRDriveTalk1.AxesMask=8 ‘Axis 3 to be connected
ACRDriveTalk1.AxesMask=65535 ‘ Axis 0 to 15 are to be connected
```

EnableDriveTalk

Description Represents the enabled/disabled status of the drive talk controller.

Property **EnableDriveTalk**
Return Type Boolean
Range N/A
Default FALSE
Example ACRDriveTalk1.EnableDriveTalk=True

DriveDataMask

Description DriveDataMask is a 32-bit mask that indicates what data parameters the controller should query the Aries Drive for. The lists of the data parameter types are indicated in Bits10496-10750. So it indicates what information the user is interested in.

Property **DriveDataMask**
Return Type Double
Range N/A
Default N/A
Example ACRDriveTalk1.DriveDataMask =128 'for DriveReset
ACRDriveTalk1.DriveDataMask=1048576 'for Actual Torque

DriveTalkMode

Description Represents the mode of communication with the DriveTalk enabled drives. This property has 3 modes.

1 – Drive Talk
2 – DTalk
3 – TalkTo

Property **DriveTalkMode**
Return Type Long
Range 1 - 3
Default N/A
Example ACRDriveTalk1.DriveTalkMode=3 'Mode is set to TalkTo

ConnectionControl

Description This property will hold the reference of the instance of connection control.

Property **ConnectionControl**
Return Type Object
Range N/A
Default N/A
Example Set ACRDriveTalk1.ConnectionControl=ACRConnection1
'ACRConnection1 is the name of Connection Control in the
'Current form

Methods

GetDriveDataRequest

Description This method used to initiate the controller to query the drive. After the query the data should be available in Parameters P28672-30543. This method will call the IControl.SetParmLong and IControl.SetFlag methods of the ComACRsrvr.dll to change the drive data, based on AxesMask and DriveDataMask properties. This method can be called only if EnableDriveTalk is True and DriveTalkMode is 1.

Signature **GetDriveDataRequest ()**

Return Type N/A

Parameters N/A

Return N/A

Example

```
ACRDriveTalk1.AxesMask=1     ' For Axis 0
ACRDriveTalk1.DriveDataMask=128
'The following will assign 128 to parameter P4424
Call ACRDriveTalk1.GetDriveDataRequest ()
```

GetConfig

Description This method will call the IControl.SetFlag () method of the comACRSrvr.dll to get the drive configuration from the drive. This method will call the SetFlag () method and pass the value for the nBit parameter based on the axis number. (Axis number can be received from AxesMask property). This method can be called only if EnableDriveTalk is True and DriveTalkMode is 1.

nBit for Axis0 is 10498
nBit for Axis1 is 10530
nBit for Axis2 is 10562
nBit for Axis3 is 10594
nBit for Axis4 is 10626
nBit for Axis5 is 10658
nBit for Axis6 is 10690
nBit for Axis7 is 10722
nBit for Axis8 is 10754
nBit for Axis9 is 10786
nBit for Axis10 is 10818
nBit for Axis11 is 10850
nBit for Axis12 is 10882
nBit for Axis13 is 10914
nBit for Axis14 is 10946
nBit for Axis15 is 10978

Signature **GetConfig ()**

Return Type N/A

Parameters N/A

Return N/A

Example

```
ACRDriveTalk1.AxesMask=3  ‘ for axis 0 and axis 1
Call ACRDriveTalk1.GetConfig ()  ‘ it will set bit 10498 and 10530
```

SendConfig

Description This method will call the IControl.SetFlag () method of the comACRSrvr.dll to send the configuration to the drive. This method will call the SetFlag () method and pass the value for the nBit parameter based on the axis number. This method can be called only if EnableDriveTalk is True and DriveTalkMode is 1.

nBit for Axis0 is 10497
nBit for Axis1 is 10529
nBit for Axis2 is 10561
nBit for Axis3 is 10593
nBit for Axis4 is 10625
nBit for Axis5 is 10657
nBit for Axis6 is 10689
nBit for Axis7 is 10721
nBit for Axis8 is 10753
nBit for Axis9 is 10785
nBit for Axis10 is 10817
nBit for Axis11 is 10849
nBit for Axis12 is 10881
nBit for Axis13 is 10913
nBit for Axis14 is 10945
nBit for Axis15 is 10977

Signature **SendConfig ()**

Return Type N/A

Parameters N/A

Return N/A

Example

```
ACRDriveTalk1.AxesMask=3  ‘ For axis 0 and axis 1
Call ACRDriveTalk1.SendConfig ()  ‘ It will set bit 10497 and 10529
```

GetErrorLog

Description This method will call the IControl.SetFlag () method of the comACRSrvr.dll to get the error log from the drive. This GetErrorLog () method should call the SetFlag () method and pass the value for the nBit

parameter based on the axis number. This method can be called only if EnableDriveTalk is True and DriveTalkMode is 1.

nBit for Axis0 is 10499
nBit for Axis1 is 10531
nBit for Axis2 is 10563
nBit for Axis3 is 10595
nBit for Axis4 is 10627
nBit for Axis5 is 10659
nBit for Axis6 is 10691
nBit for Axis7 is 10723
nBit for Axis8 is 10755
nBit for Axis9 is 10787
nBit for Axis10 is 10819
nBit for Axis11 is 10851
nBit for Axis12 is 10883
nBit for Axis13 is 10915
nBit for Axis14 is 10947
nBit for Axis15 is 10979

Signature **GetErrorLog ()**

Return Type N/A

Parameters N/A

Return N/A

Example

```
ACRDriveTalk1.AxesMask=3  ‘ for axis 0 and axis 1
Call ACRDriveTalk1.GetErrorLog () ‘ It will set bit 10499 and 10531
```

SendASTFile

Description This method calls the IUtility.DownloadFile () to send the config file generated from the Aries Support Tool (.AST file) to the drive. This method can be called only if the property EnableDriveTalk is set to true and DriveTalkMode it is set to 2 or 3.

Signature **SendASTFile** (strASTFileName as String)

Return Type N/A

Parameters

strASTFileName	Fully qualified AST file name.

Return N/A

Example

```
ACRDriveTalk1.SendASTFile (“Test1.AST”)
```

EStop Control Properties and Methods

This control can be called as Emergency Stop. The control will have the label as “EStop” and “Clear EStop”. EStop will stop the motion of the motor and also it can disable the drive based on the Boolean property “DisableDriveonEStop”. Clear EStop will be in blinking stage and this clears the KAMR bit set and makes the motor ready for motion if the motor is not disabled.

EStop Control contains the following Properties and Methods.

Properties

- **OLE_COLOR ClearColor**
- **OLE_COLOR StopColor**
- **String ClearLabel**
- **String StopLabel**
- **Boolean DisableDriveOnEStop**
- **Object ConnectionControl**

Methods

- **Void SendEStop ()**
- **Void ClearEStop ()**

Properties

ClearColor

Description	This property holds the back color of the control and will be displayed only when the control is in “Clear EStop” stage. This will be in blinking stage.
Property	**ClearColor**
Return Type	OLE_COLOR
Range	N/A
Default	Green
Example	ACREStop1.ClearColor = RGB (75, 75, 75)

StopColor

Description	This property holds the back color of the control and will be displayed only when the control is in “EStop” stage.
Property	**StopColor**
Return Type	OLE_COLOR
Range	N/A
Default	Red
Example	ACREStop1.StopColor = RGB (75, 75, 75)

ClearLabel

Description	This property holds the text and will be displayed only when the control is in clear Estop stage.
Property	**ClearLabel**
Return Type	String
Range	N/A
Default	Clear EStop
Example	ACREStop1.ClearLabel = “Clear EStop”

StopLabel

Description	This property holds the text and will be displayed only when the control is in Estop stage. Default text is “EStop”.
Property	**StopLabel**
Return Type	String
Range	N/A
Default	EStop
Example	ACREStop1.StopLabel = “EStop”

DisableDriveOnEStop

Description	Property determines whether the drives should be disabled or not while stopping. If true the drives are disabled while pressing EStop button else drives will not be disabled.
Property	**DisableDriveOnEStop**
Return Type	Boolean
Range	N/A
Default	FALSE
Example	ACREStop1.DisableDriveonEstop = True

Connectioncontrol

Description	This property will hold the reference of the instance of connection control.
Property	**Connectioncontrol**
Return Type	Object
Range	N/A
Default	N/A
Example	Set ACREStop1.ConnectionControl=ACRConnection1

Methods

SendEStop

Description	This method will send the “Chr (26)” to the controller, which will stop all the motion of the motors, and the KAMR bit will be set.
Signature	**SendEStop ()**
Return Type	N/A
Parameters	N/A
Return	N/A
Example	ACREStop1.SendEStop ()

ClearEStop

Description	This method will send the “Chr (25)” to the controller, which will clear only the KAMR bit set.
Signature	**ClearEStop ()**
Return Type	N/A
Parameters	N/A
Return	N/A
Example	ACREStop1.ClearEStop ()