Stroke Scribe

StrokeReader properties and methods

StrokeReader properties

Member	Description and Examples

Port	Changes the serial port number to which the ActiveX is connected.
	
	This parameter can be changed on the fly while receiving or transmitting, but all data buffered but not yet delivered to the user application will be lost.
	
	To get a list of port numbers available on the particular PC, use PortsAvailable property.
	
	Example:
	
	`StrokeReader1.Port=1 'Connects to COM1:`
	
	`StrokeReader1.Port=9 'Connects to COM9:`
	`If StrokeReader1.Error Then`
	`  MsgBox StrokeReader1.ErrorDescription`
	`End If`

BaudRate	Sets the data rate of the serial port. Most popular standard values are: 9600, 57600, 115200 **Example:** `StrokeReader1.BaudRate=115200` **NOTE:** After specifying the port speed, you can read back the **BaudRate** value to check the serial port driver supports your data rate setting. The property may be automatically changed to the nearest value supported by the driver.
DataBits	The number of bits in each byte transmitted or received by serial port. The default setting is 8 bits. **Example:** `StrokeReader1.DataBits=8`
Parity	The parity scheme to be used by the serial port. Can be one of the following values: • NOPARITY = 0 • ODDPARITY = 1 • EVENPARITY = 2 • MARKPARITY = 3 • SPACEPARITY = 4 **NOPARITY** is the default value. **Example:** `StrokeReader1.Parity = EVENPARITY`
StopBits	The number of stop bits to be used. Can be one of the following values: • ONESTOPBIT = 0 • ONE5STOPBITS = 1 • TWOSTOPBITS = 2 **ONESTOPBIT** is the default value. **Example:** `StrokeReader1.Parity = ONE5STOPBITS`

StrokeReader ActiveX
About StrokeSeader
Download/Buy
Contacts

Serial port programming
Developer zone
Properties and methods
Programming examples

Serial port in Office, .NET, JS
Excel 2007
Access 2010
VB.NET
JavaScript
JavaScript + Binary data

Non-ActiveX
Serial port programming with WinAPI

Error	Exposes the result code for the last action (property assignment or method call). **Example:** See ErrorDescription property.

ErrorDescription	Contains the error message string for the **Error** property code.
	Example:
	`StrokeReader1.Port=88 'Trying to connect to non-existent port`
	`If StrokeReader1.Error <> 0 Then`
	`  MsgBox StrokeReader1.ErrorDescription`
	`End If`

Connected	When set to *true*, connects the ActiveX to the serial port specified by the **.Port** property. When set to *false*, disconnects the ActiveX from the serial port. The ActiveX will try to automatically reconnect to the USB adapter (which port name is specified by the **.Port** property) upon it's reconnection to PC. To disable the automatic reconnection, set **.Connected** to false. The user application can be notified about connection/disconnection by firing of the CommEvent.

PortsAvailable	Returns the list of serial ports available on PC.
	The list is the text string with port numbers separated by commas: "1,3,9" for COM1, COM3 and COM9
	Example:
	`ports = StrokeReader1.PortsAvailable`
	`MsgBox ports`

PortsAvailableArr	Returns an array of serial port numbers available on PC.
	VBA Example:
	`ports = StrokeReader1.PortsAvailableArr`
	`For i = 0 To UBound(ports)`
	`  p = ports(i)`
	`  MsgBox p & " " & StrokeReader1.GetPortFriendlyName(p)`
	`Next i`
	Javascript Example:
	`vbPortArr = port.PortsAvailableArr;`
	`ports = new VBArray(vbPortArr).toArray();`
	`for (i = 0; i<ports.length; i++)`
	`{`
	`  alert(ports[i] + " " + port.GetPortFriendlyName(ports[i]));`
	`}`

DataMode	Specifies how the received data is reported to the user application. Can be one of: • BINARY = 0 • TEXT = 1 • BINARYJS = 2 **BINARY** is the default setting. When **DataMode** is set to **BINARY**, user application will receive an array of bytes (COM/OLE: SAFEARRAY of BYTE). In the **TEXT** mode, data is reported as a text string with nul-bytes substituted with user defined values (COM/OLE: BSTR). See CommEvent description and NullSubst property. In the **BINARYJS** mode, data is reported as Javascript-compatible arrays with bytes (COM/OLE: SAFEARRAY of VARIANT of BYTE) and can be converted to normal arrays using VBArray class.

	Example: `StrokeReader1.DataMode = TEXT` See the sample VBA code for events for more information how to handle incoming data depending on the value of **DataMode**.
NullSubst	Specifies the value which is used to substitute nul bytes in the **TEXT** data mode. **Example:** `StrokeReader1.NullSubst = 32` 'zeros will be substituted with spaces (ASCII 32) `StrokeReader1.NullSubst = Asc("!")` 'zeros will be substituted with exclamation characters

DTR
RTS
	These properties allow to set the state of DTR and RTS lines of the serial port.
	Example:
	`StrokeReader1.DTR=true`
	`StrokeReader1.RTS=false`

DSR
CTS
RI
	These properties allow to read the current state of DTR, RTS and RING lines.
	The user application can be asynchronously notified about the serial port state change by catching of the CommEvent.
	Example:
	`If StrokeReader1.CTS Then`
	`  MsgBox "CTS line set"`
	`Else`
	`  MsgBox "CTS line clear"`
	`End If`

CtsFlow
DsrFlow
	If CtsFlow is **True**, the remote device can suspend the data output by turning the **CTS** line off. The driver will resume data transmission when **CTS** goes on.
	If DsrFlow is **True**, the remote device can suspend the data output by turning the **DSR** line off. The driver will resume data transmission when **DSR** goes on.
	False is the default value of both properties (driver ignores the CTS/DSR lines and always transmits queued data to the remote device).
	These properties are the same with fOutxCtsFlow/fOutxDsrFlow members of the DCB structure, see MSDN.
	Example:
	`StrokeReader1.CtsFlow = True`
	`StrokeReader1.DsrFlow = False`

RecvIntervalTimeout	If the time interval between arrival of any two bytes exceeds the amount set by RecvIntervalTimeout, the data already received is returned to user application using CommEvent. **Example:** `StrokeReader1.ReadIntervalTimeout=50` 'A 50 millisecond timeout for incoming data reception
RecvSizeThreshold	Specifies the maximum amount of bytes to receive without timeout. This guarantees periodic CommEvent firing to the user application when receiving large amount of data without timeouts between data bytes. **Example:** `StrokeReader1.RecvSizeThreshold =1000` 'Fire an event for each 1000 received bytes

Methods

Method	Description and Examples

Send	This method allows to send a text string or byte array or a single byte value to the serial port.
	Syntax:
	long Send(VARIANT data)
	Return value:
	Zero on success. (**TBD**)
	Example:
	`dim x(3) as byte 'Sending an array of bytes to serial port`
	`x(1)=1`
	`x(2)=2`
	`x(3)=3`
	`StrokeReader1.Send(x)`
	
	`Dim s As String 'Sending a text string to serial port`
	`s = "ABC"`
	`StrokeReader1.Send s`
	
	`Dim b As Byte 'Sending one byte to serial port`
	`b = &H21 ' ASCII "!"`
	`StrokeReader1.Send b`

GetPortFriendlyName	Returns the user-friendly hardware description string for specified port number.
	Syntax:
	String GetPortFriendlyName(Long port)
	Return value:
	A manufacturer-specified hardware description string.
	Example:
	How to read a description string for known serial port number:
	`s = StrokeReader1.GetPortFriendlyName(12) 'Returns description of COM12:`
	`MsgBox s 'For example: "HDAUDIO Soft Data Fax Modem with SmartCP"`
	How to list all available serial ports with hardware descriptions in Excel:
	`Dim x() As String`
	`avail = StrokeReader1.PortsAvailable 'Returns a list of serial ports: "1,4,9,10"`
	`x = Split(avail, ",") 'Splits to array of strings: x[0]="1", x[1]="4", ...`
	
	`For i = 0 To UBound(x)`
	`  s = StrokeReader1.GetPortFriendlyName(x(i)) 'A friendly name of port`
	`  Row = i + 1 'Excel row number must start from 1`
	`  Cells(Row, 1) = "COM" + x(i) 'Ex: "COM9"`
	`  Cells(Row, 2) = s 'Ex: "Standard Modem over Bluetooth link"`
	`Next i`

SetBreak	Allows to put the transmission line into a break state.
	Syntax:
	SetBreak(bool break)
	If *break=true*, the data transmission will be paused and TX line will be put into a break state. If *break=false*, the data transmission will be resumed.
	Example:
	`StrokeReader1.SetBreak(true)`
	`StrokeReader1.SetBreak(false)`

Events

Event	Description and Examples

CommEvent	This event is fired by the ActiveX when the data is arrived or state of the serial port lines is changed.
	
	Syntax:
	
	CommEvent(ByVal Evt As StrokeReader.Event, ByVal data As Variant)
	
	Parameters:
	
	Evt - Can be one of following values:
	
	• EVT_DISCONNECT (=0)
	• EVT_DATA (=1)
	• EVT_SERIALEVENT (=2)
	• EVT_CONNECT (=3)
	• EVT_ARRIVAL (=4)
	• EVT_REMOVE (=5)
	
	Data - Can contain an array of received bytes or a text string or an integer mask value indicating the type of serial port events that occurred (depending on the value of DataMode).
	
	EVT_DISCONNECT means the serial port (to which the ActiveX is bound) is just removed from the system. The **data** parameter is not used with this event code. This event may be generated by a USB-To-Serial adapter or by a USB Bluetooth dongle. In most cases, the system does not remove the serial port when a remote Bluetooth device goes offline.
	
	EVT_CONNECT means the serial port specified in the Port property becomes available and the ActiveX just successfully connected to the port. The **data** parameter is not used with this event code. This event may be generated by a USB-To-Serial adapter or by a Bluetooth dongle. A USB dongle may show some serial ports available even if the remote device stays offline.
	
	EVT_DATA event is fired if there is some data is received from the serial port.
	
	• If DataMode = TEXT, the **data** parameter contains the pointer to the text string (BSTR).
	• If DataMode = BINARY, the **data** parameter contains the pointer to the byte array (SAFEARRAY).
	• If DataMode = BINARYJS, the **data** parameter contains a reference to Javascript-compatible byte array (SAFEARRAY of VARIANT of BYTE).
	
	EVT_SERIALEVENT is sent when some serial port lines has changed their state. The **data** parameter contains combination of the following flags:
	
	• EV_BREAK = 0x0040 - A break was detected on input
	• EV_CTS = 0x0008 - The CTS signal changed state
	• EV_DSR = 0x0010 - The DSR signal changed state
	• EV_ERR = 0x0080 - A line-status error occurred (framing, overrun or parity)
	• EV_RING = 0x0100 - A ring indicator was detected
	• EV_RLSD = 0x0020 - The RLSD signal changed state
	• EV_TXEMPTY = 0x0004 - The last character in the output buffer was sent
	
	EVT_ARRIVAL is sent if a new serial port becomes available. The application can use this event to detect when a USB VCP device is connected to the system. The **data** parameter contains the number of the serial port just became available.
	
	The **EVT_ARRIVAL** message is only intended for maintaining the list of available ports displayed to the user. Do not try to connect to the port number specified in the **data** parameter upon receiving of EVT_ARRIVAL.
	
	EVT_REMOVE is sent when a serial port becomes unavailable. The **data** parameter contains the number of the serial port just removed from the system. This event can be generated by any port in the system - not just by the port specified in the Port property.
	
	Example:
	
	`Private Sub StrokeReader1_CommEvent( _`
	`ByVal Evt As StrokeReader.Event, _`
	`ByVal data As Variant)`

```
  Select Case Evt
    Case EVT_DISCONNECT
        MsgBox "Disconnected"

    Case EVT_CONNECT
        MsgBox "Connected"

    Case EVT_DATA
        If StrokeReader1.DataMode = Text Then
            s = data
        Else
            s = StrConv(data, vbUnicode)
        End If

        MsgBox s

    Case EVT_SERIALEVENT
        If CLng(data) And EV_CTS Then
            MsgBox "CTS=" + Str(StrokeReader1.CTS)
        End If
        If CLng(data) And EV_DSR Then
            MsgBox "DSR=" + Str(StrokeReader1.DSR)
        End If

    Case EVT_ARRIVAL
        Dim port As Integer
        port = data
        Debug.Print "EVT_ARRIVAL:" & Str(port)
        Debug.Print StrokeReader1.GetPortFriendlyName(port)

    Case EVT_REMOVE
        Dim port As Integer
        port = data
        Debug.Print "EVT_REMOVE, COM" & Format(port)
    End Select
End Sub
```

Last modified: 05-Mar-2015 14:15 UTC