Setting Unusual Baud Rates with Visual Basic

Synopsis

The standard **MSComm Control** supplied with **Microsoft Visual Basic** only supports a few baud rates in its **Settings** property. This is a problem if you need to operate a COM port at a baud rate that is not supported by the **Settings** property. One solution is to use a third-party replacement for the **MSComm Control**. Another solution is to bypass the **MSComm Control** when setting the baud rate and go straight to the **Win32 API**. This tip shows how to implement the second solution.

Implementation

In order for the following Visual Basic functions to work, the project must include a module containing definitions extracted from the **Win32API.Txt** file supplied with Visual Basic. A suitable module may be found here: **w32comms.bas**.

The following Visual Basic subroutine uses the **GetCommState** and **SetCommState** functions in the **Win32 API** to set the baud rate:

```
' Set baud rate using Win32 API.
```

```
' The PortOpen property should be set to True before calling.
' May raise the following errors:
'   comPortNotOpen  the PortOpen property has not been set to True
'   comDCBError     failed to read current state of the port
'   comSetCommStateFailed  failed to set new baud rate
Sub SetBaudRate(Com As MSComm, baud As Long)
Dim ComDcb As dcb
Dim ret As Long

    ' Check port is open
    If Not Com.PortOpen Then
        Err.Raise comPortNotOpen, Com.Name, _
            "Operation valid only when the port is open"
        Exit Sub
    End If

    ' Get existing Comm state
    ret = GetCommState(Com.CommId, ComDcb)
    If ret = 0 Then
        Err.Raise comDCBError, Com.Name, _
            "Could not read current state of the port"
        Exit Sub
    End If

    ' Modify state with new baud rate
    ComDcb.BaudRate = baud
    ' Set the new Comm state
    ret = SetCommState(Com.CommId, ComDcb)
    If ret = 0 Then
        Err.Raise comSetCommStateFailed, Com.Name, _
            "Could not set port to specified baud rate"
        Exit Sub
    End If
End Sub
```

Here is a matching Visual Basic function to read the current baud rate using the **GetCommState** function in the **Win32 API**:

```
' Get baud rate using Win32 API
' The PortOpen property should be set to True before calling.
' May raise the following errors:
'   comPortNotOpen  the PortOpen property has not been set to True
'   comDCBError     failed to read current state of the port
Function GetBaudRate(Com As MSComm) As Long
Dim ComDcb As dcb
Dim ret As Long

    GetBaudRate = 0

    ' Check port is open
    If Not Com.PortOpen Then
        Err.Raise comPortNotOpen, Com.Name, _
            "Operation valid only when the port is open"
        Exit Function
    End If

    ' Get Comm state
    ret = GetCommState(Com.CommID, ComDcb)
    If ret = 0 Then
        Err.Raise comDCBError, Com.Name, _
            "Could not read current state of the port"
        Exit Function
```

```
    End If

    ' Extract baud rate
    GetBaudRate = ComDcb.BaudRate
End Function
```

Usage

For a **MSComm** object named **MSComm1**, the above Visual Basic **SetBaudRate** subroutine and **GetBaudRate** function may be called *after* setting **MSComm1.PortOpen** to **True**. Before this, **MSComm1.CommPort** must be set to the desired COM port number and **MSComm1.Settings** may need to be set if the default value of "**9600,N,8,1**" is unacceptable. (The baud rate specified in this string value is unimportant, but needs to be one of those acceptable to the **Settings** property.)

For a baud rate *b*, the COM port's baud rate may be changed as follows:

```
SetBaudRate MSComm1, b
```

If the serial driver for the COM port will not accept the specified baud rate, a run-time error **comSetCommStateFailed** will be raised. Other possible run-time errors include **comPortNotOpen** (if the **PortOpen** property is **False**) or **comDCBError** (unlikely).

The COM port's current baud rate may be read into a variable **b** of type **Long** as follows:

```
b = GetBaudRate(MSComm1)
```

Possible run-time errors include **comPortNotOpen** (if the **PortOpen** property is **False**) or **comDCBError** (unlikely).

Caveats

Changing the **MSComm** object's **Settings** property will reset the COM port's baud rate to that specified by the **Settings** property, as will changing the **Handshaking**, **NullDiscard**, **ParityReplace**, **PortOpen** or **RTSEnable** property.

If program control of the RTS line is required, it will be necessary to write a Visual Basic subroutine to manipulate the COM port's device control block (DCB) directly, in a similar manner to the above **SetBaudRate** subroutine, rather than by setting the **RTSEnable** property. (The alternative of following a change to **RTSEnable** with a call to **SetBaudRate** would result in data glitches.) However, any changes made by this function would be subject to the same caveats as the **SetBaudRate** function. If this level of control is required, it is worth considering the use of a third-party alternative to the **MSComm** control.