



---

Supervision and control XML-based  
from Windows Vista to Windows CE

# **I/O Drivers VBA Interface HandBuch**

---

Version 11.3 - Ed. Feb.2012

Cod. DOCS 11 VBD-E Build 1101

# Inhaltsverzeichnis

<b>1. API BASIC-SCRIPT-SCHNITTSETELLEN .....</b>	<b>5</b>
1.1. DIE BENUTZUNG DER BASIC-SCRIPT-SCHNITTSTELLE.....	5
<b>2. DRIVERINTERFACE .....</b>	<b>7</b>
2.1. ADDDRIVERTASK, DRIVERINTERFACE FUNCTION .....	7
2.2. DELAYEVENTS, DRIVERINTERFACE PROPERTY .....	9
2.3. FIREONDATASENT, DRIVERINTERFACE EVENT .....	10
2.4. FIREONNEWDATA, DRIVERINTERFACE EVENT.....	10
2.5. FIREONQUALITYCHANGED, DRIVERINTERFACE EVENT .....	10
2.6. GETDRIVERSTATION, DRIVERINTERFACE FUNCTION .....	11
2.7. GETDRIVERTASK, DRIVERINTERFACE FUNCTION.....	11
2.8. INITDRIVER, DRIVERINTERFACE FUNCTION.....	12
2.9. REMOVEDRIVERTASK, DRIVERINTERFACE FUNCTION.....	13
2.10. SETDRIVER, DRIVERINTERFACE FUNCTION .....	13
2.11. TERMINATEDRIVER, DRIVERINTERFACE FUNCTION.....	14
<b>3. TASKINTERFACE .....</b>	<b>17</b>
3.1. AUTODELETE, TASKINTERFACE PROPERTY .....	17
3.2. CONDITIONALVARIABLE, TASKINTERFACE PROPERTY .....	17
3.3. EXECUTE, TASKINTERFACE FUNCTION.....	18
3.4. GETFLATVARIABLELIST, TASKINTERFACE FUNCTION .....	18
3.5. GETMAXBYTESIZE, TASKINTERFACE FUNCTION .....	19
3.6. GETREADBYTEBUFFER, TASKINTERFACE FUNCTION .....	20
3.7. GETSTATIONOBJECT, TASKINTERFACE FUNCTION .....	21
3.8. GETTOTALSIZEBYTES, TASKINTERFACE FUNCTION.....	22
3.9. GETVARIABLELISTSEPARATOR, TASKINTERFACE FUNCTION .....	22
3.10. GETXMLSETTINGS, TASKINTERFACE FUNCTION .....	23
3.11. HASBEENEXECUTED, TASKINTERFACE FUNCTION .....	24
3.12. INUSE, TASKINTERFACE PROPERTY .....	24
3.13. ISVALID, TASKINTERFACE FUNCTION.....	25
3.14. LASTEXECUTIONTIME, TASKINTERFACE PROPERTY.....	26
3.15. NAME, TASKINTERFACE PROPERTY .....	26
3.16. POLLINGTIME, TASKINTERFACE PROPERTY.....	27
3.17. POLLINGTIMENOINUSE, TASKINTERFACE PROPERTY .....	28
3.18. SETWRITEBYTEBUFFER, TASKINTERFACE FUNCTION .....	28
3.19. STATION, TASKINTERFACE PROPERTY .....	29
3.20. SWAPBYTES, TASKINTERFACE PROPERTY .....	29
3.21. SWAPWORDS, TASKINTERFACE PROPERTY.....	30
3.22. TOSTRING, TASKINTERFACE FUNCTION .....	31
3.23. TYPE, TASKINTERFACE PROPERTY .....	31
3.24. WRITEOUTPUTSATSTARTUP, TASKINTERFACE PROPERTY .....	32
<b>4. STATIONINTERFACE .....</b>	<b>33</b>
4.1. GETBASESTATIONINTERFACE, STATIONINTERFACE FUNCTION .....	33
4.2. GETSUBSTATIONINTERFACE, STATIONINTERFACE FUNCTION .....	33
4.3. GETTASK, STATIONINTERFACE FUNCTION .....	34
4.4. GETXMLSETTINGS, STATIONINTERFACE FUNCTION.....	35
4.5. ISINERROR, STATIONINTERFACE FUNCTION.....	36
4.6. LASTERRORCODE, STATIONINTERFACE PROPERTY.....	36
4.7. LASTERRORSTRING, STATIONINTERFACE PROPERTY .....	37
4.8. NAME, STATIONINTERFACE PROPERTY .....	38

4.9. QUALITY, STATIONINTERFACE PROPERTY .....	38
4.10. STATIONTYPE, STATIONINTERFACE PROPERTY .....	39
<b>5. SERIALSTATIONINTERFACE .....</b>	<b>41</b>
5.1. BAUDRATE, SERIALSTATIONINTERFACE PROPERTY .....	41
5.2. BYTESIZE, SERIALSTATIONINTERFACE PROPERTY .....	42
5.3. CDTIMEOUT, SERIALSTATIONINTERFACE PROPERTY .....	42
5.4. CtsTIMEOUT, SERIALSTATIONINTERFACE PROPERTY .....	43
5.5. DsrTIMEOUT, SERIALSTATIONINTERFACE PROPERTY .....	43
5.6. FLOWCONTROL, SERIALSTATIONINTERFACE PROPERTY .....	43
5.7. GETBASESTATIONINTERFACE, SERIALSTATIONINTERFACE FUNCTION .....	44
5.8. GETSUBSTATIONINTERFACE, SERIALSTATIONINTERFACE FUNCTION .....	45
5.9. GETXMLSETTINGS, SERIALSTATIONINTERFACE FUNCTION .....	45
5.10. KEEPPORTOPENED, SERIALSTATIONINTERFACE PROPERTY .....	46
5.11. PARITY, SERIALSTATIONINTERFACE PROPERTY .....	47
5.12. PORTID, SERIALSTATIONINTERFACE PROPERTY .....	48
5.13. RXTIMEOUT, SERIALSTATIONINTERFACE PROPERTY .....	48
5.14. SIZERECEIVINGQUEUE, SERIALSTATIONINTERFACE PROPERTY .....	49
5.15. SIZETRANSMISSIONQUEUE, SERIALSTATIONINTERFACE PROPERTY .....	50
5.16. STOPBITS, SERIALSTATIONINTERFACE PROPERTY .....	51
5.17. TXTIMEOUT, SERIALSTATIONINTERFACE PROPERTY .....	51
<b>6. TAPISTATIONINTERFACE .....</b>	<b>53</b>
6.1. DISCONNECTAFTERSECS, TAPISTATIONINTERFACE PROPERTY .....	53
6.2. ENABLETAPICALLONTHISSTATION, TAPISTATIONINTERFACE PROPERTY ..	54
6.3. ENDCONNECTIONTIME, TAPISTATIONINTERFACE PROPERTY .....	54
6.4. GETBASESTATIONINTERFACE, TAPISTATIONINTERFACE FUNCTION .....	55
6.5. GETSUBSTATIONINTERFACE, TAPISTATIONINTERFACE FUNCTION .....	56
6.6. GETXMLSETTINGS, TAPISTATIONINTERFACE FUNCTION .....	56
6.7. ISCONNECTED, TAPISTATIONINTERFACE FUNCTION .....	57
6.8. LASTCONNECTIONTIME, TAPISTATIONINTERFACE PROPERTY .....	58
6.9. LASTTAPIERROR, TAPISTATIONINTERFACE PROPERTY .....	58
6.10. LASTTAPIERRORSTRING, TAPISTATIONINTERFACE PROPERTY .....	58
6.11. PHONENUMBER, TAPISTATIONINTERFACE PROPERTY .....	59
6.12. PROMPTFORCONNECTION, TAPISTATIONINTERFACE PROPERTY .....	59
6.13. RETRIES, TAPISTATIONINTERFACE PROPERTY .....	60
6.14. RETRYAFTERSECS, TAPISTATIONINTERFACE PROPERTY .....	60
6.15. SHOWCONNECTIONDLG, TAPISTATIONINTERFACE PROPERTY .....	61
6.16. STARTCONNECTIONTIME, TAPISTATIONINTERFACE PROPERTY .....	61
6.17. TOTALCONNECTIONTIME, TAPISTATIONINTERFACE PROPERTY .....	62
<b>7. TCPIPSTATIONINTERFACE .....</b>	<b>63</b>
7.1. BACKUPSERVERADDRESS, TCPIPSTATIONINTERFACE PROPERTY .....	63
7.2. GETBASESTATIONINTERFACE, TCPIPSTATIONINTERFACE FUNCTION .....	64
7.3. GETSUBSTATIONINTERFACE, TCPIPSTATIONINTERFACE FUNCTION .....	65
7.4. GETXMLSETTINGS, TCPIPSTATIONINTERFACE FUNCTION .....	65
7.5. LOCALBOUNDADDRESS, TCPIPSTATIONINTERFACE PROPERTY .....	66
7.6. LOCALBOUNDPORT, TCPIPSTATIONINTERFACE PROPERTY .....	66
7.7. MAXRECEIVE, TCPIPSTATIONINTERFACE PROPERTY .....	67
7.8. MAXSEND, TCPIPSTATIONINTERFACE PROPERTY .....	68
7.9. RXTIMEOUT, TCPIPSTATIONINTERFACE PROPERTY .....	68
7.10. SERVERADDRESS, TCPIPSTATIONINTERFACE PROPERTY .....	69
7.11. SERVERPORT, TCPIPSTATIONINTERFACE PROPERTY .....	70
7.12. SWITCHSERVERTIMEOUT, TCPIPSTATIONINTERFACE PROPERTY .....	71
7.13. TXTIMEOUT, TCPIPSTATIONINTERFACE PROPERTY .....	71
<b>8. RASSTATIONINTERFACE .....</b>	<b>73</b>
8.1. DISCONNECTAFTERSECS, RASSTATIONINTERFACE PROPERTY .....	73

8.2. ENABLERASCALLONTHISSTATION, RASSTATIONINTERFACE PROPERTY .... 73  
8.3. ENDCONNECTIONTIME, RASSTATIONINTERFACE PROPERTY ..... 74  
8.4. GETBASESTATIONINTERFACE, RASSTATIONINTERFACE FUNCTION ..... 74  
8.5. GETSUBSTATIONINTERFACE, RASSTATIONINTERFACE FUNCTION ..... 75  
8.6. GETXMLSETTINGS, RASSTATIONINTERFACE FUNCTION ..... 75  
8.7. ISCONNECTED, RASSTATIONINTERFACE FUNCTION ..... 76  
8.8. LASTCONNECTIONTIME, RASSTATIONINTERFACE PROPERTY ..... 77  
8.9. LASTRASERRORNUMBER, RASSTATIONINTERFACE PROPERTY ..... 77  
8.10. LASTRASERRORSTRING, RASSTATIONINTERFACE PROPERTY ..... 78  
8.11. NUMRETRIES, RASSTATIONINTERFACE PROPERTY ..... 78  
8.12. PASSWORD, RASSTATIONINTERFACE PROPERTY ..... 78  
8.13. PHONEBOOKENTRY, RASSTATIONINTERFACE PROPERTY ..... 79  
8.14. PHONENUMBER, RASSTATIONINTERFACE PROPERTY ..... 80  
8.15. PROMPTFORCONNECTION, RASSTATIONINTERFACE PROPERTY ..... 81  
8.16. RETRYAFTERSECS, RASSTATIONINTERFACE PROPERTY ..... 81  
8.17. SHOWCONNECTIONDLG, RASSTATIONINTERFACE PROPERTY ..... 81  
8.18. STARTCONNECTIONTIME, RASSTATIONINTERFACE PROPERTY ..... 82  
8.19. TOTALCONNECTIONTIME, RASSTATIONINTERFACE PROPERTY ..... 83  
8.20. USERNAME, RASSTATIONINTERFACE PROPERTY ..... 83

# 1. API Basic-Script-Schnittstellen

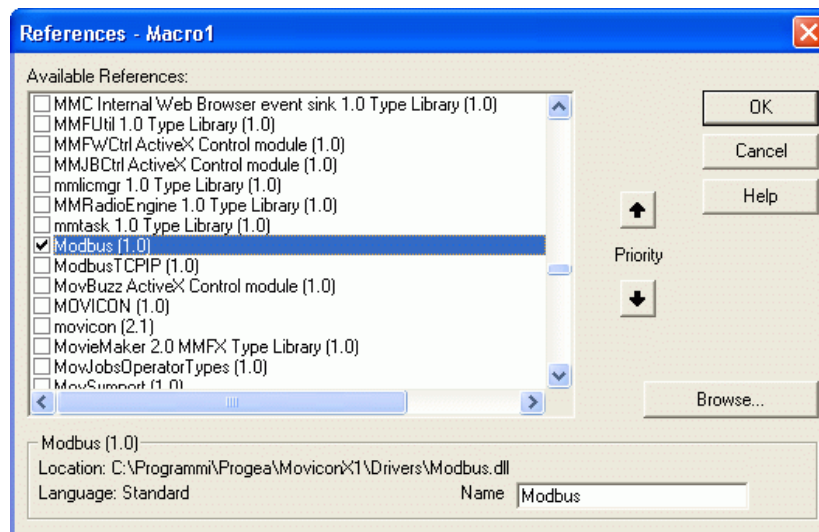
## 1.1. Die Benutzung der Basic-Script-Schnittstelle

Der Supervisor erlaubt Ihnen, eine Reihe von Funktionen zum Management der Kommunikationstreiber mit Basic-Scripts zu benutzen. Damit können Sie z.B. Kommunikationsaufgaben während der Laufzeit erstellen oder zerstören, Informationen über den Status von Stationsaufgaben abzurufen, usw.



**Um die COM-Schnittstelle der Kommunikationstreibern (Component Object Model) benutzen zu können, müssen Sie in den Lizenz-Optionen die Option "VBA-Treiber" aktiviert haben. Ansonsten werden die Basic-Script-Funktionen der Treiber nicht in Laufzeit ausgeführt werden.**

Um das Programmieren zu erleichtern, können Sie die gewünschte Treiber-Referenz innerhalb der Basic-Script-Routine belegen, um die dafür benötigten Objekt-Typen zu deklarieren (z.B. DriverInterface, TaskInterface, usw.) und keine "generic" Typen (Objekt). Dies wird Ihnen eine Liste der Methoden und Eigenschaften von jedem spezifischen Objekt, das Sie deklarieren.



Es ist daran zu erinnern, dass die Referenz-Einschaltung nicht von WinCE unterstützt wird. Wenn Sie ein Projekt entwickeln müssen, das in WinCE ausgeführt werden soll, werden die Referenzen nicht eingeschaltet und Sie müssen alle Objekt-Arten deklarieren.

Die auf der Dokumentation der Funktionen angezeigten Beispiele wurden alle durch Einschalten der Treiber-Referenz erstellt.

Das folgende Beispiel kann nur in Windows 32/64 bit-Plattformen funktionieren.

Sub Main

```
Dim drv As DriverInterface
Dim station As StationInterface
Dim bError As Boolean
```

```
Set drv = GetDriverInterface("Modbus Serial")
Set station = drv.GetDriverStation("Station1")
```

```

bError = station.IsInError
MsgBox "IsInError = " & CStr(bError),vbInformation,"ERROR"

Set drv = Nothing
Set station = Nothing
End Sub

```

Um es mit der WinCE-Plattform kompatibel zu machen, müssen Treiber-Referenzen eingeschaltet und Variablen wie folgt deklariert werden:

```

Sub Main
    Dim drv As Object
    Dim station As Object
    Dim bError As Boolean

    Set drv = GetDriverInterface("Modbus Serial")
    Set station = drv.GetDriverStation("Station1")

    bError = station.IsInError
    MsgBox "IsInError = " & CStr(bError),vbInformation,"ERROR"

    Set drv = Nothing
    Set station = Nothing
End Sub

```

## Synchrone und Asynchrone Tasks

---

Wenn Sie eine Task durch die Funktion "Execute()" in Ausführung bringen, können Sie definieren, ob die Task synchron oder asynchron ausgeführt werden soll:

### Synchrone Tasks

Die Task wird synchron ausgeführt, wenn Sie der Funktion "Execute()" eine Timeout zuweisen, deren Wert größer als Null ist. Auf diese Weise, wenn die Funktion von dem Script aufgerufen wird, wird die Ausführung des Scripts abwarten, bis die Funktion beendet ist, und erst dann wird die Ausführung fortgesetzt. Gibt nun die Funktion den Wert "True" zurück, bedeutet, dass sie ausgeführt worden ist. Gibt sie den Wert "False" zurück, bedeutet, dass sie nicht ausgeführt worden ist. Dabei könnte bspw. die Timeout abgelaufen sein.



**Obwohl die Funktion "Execute()" gibt den Wert "True" zurück, wenn sie ausgeführt wurde, was wiederum bedeutet, dass die Task ausgeführt wurde, bedeutet dies nicht, dass der Task fehlerfrei ausgeführt werden könnte. Deshalb müssen Sie den Zustand der Station überprüfen, um festzustellen, dass sie fehlerfrei funktioniert, oder nicht.**

Die synchronen Tasks werden normalerweise in bestimmten Fällen eingesetzt, bei denen eine Kontrolle des tatsächlich ausgeführten Lesen/Schreiben von Daten erforderlich ist. Dies geschieht zum Beispiel, wenn man sich vergewissern muss, dass ein Rezept übertragen wurde, bevor der Befehl zum Start der Produktion erteilt wird.



Für den richtigen Betrieb der synchronen Task ist es nötig, einen Timeout-Parameter in der Funktion "Execute()" zu konfigurieren, der höher ist als der Timeout-Parameter, der man in der Station konfiguriert hat. Wurde bspw. die Timeout der Station auf 2000 msek konfiguriert, kann man für die Funktion "Execute()" eine Timeout von 5000 msek wählen. Dadurch können Sie verhindern, dass die Timeout der Funktion vor jener der Station ausgelöst wird.

### Asynchrone Tasks

Die Task kann asynchron ausgeführt werden, wenn Sie der Funktion "Execute()" eine Timeout gleich Null zuweisen. Auf diese Weise, wenn die Funktion von dem Script aufgerufen wird, wird die Ausführung des Scripts fortgesetzt, ohne abzuwarten, dass die Funktion beendet ist. So wird die Task normal ausgeführt, aber auf diese Weise kann nicht wissen, wann genau die Task ausgeführt wurde. In diesem Fall wird die Funktion immer den Wert "True" zurückgeben, sofern beim Erstellen der Task keine Fehler auftreten, wie etwa ein fehlerhafter Parameter. Der Wert "True" bei eine asynchrone Task bedeutet deshalb nicht, dass die Task ausgeführt wurde, sondern lediglich dass sie erstellt wurde, und dass sie darauf wartet, ausgeführt zu werden.

## 2. DriverInterface

### 2.1. AddDriverTask, DriverInterface Function

<b>Syntax</b>	AddDriverTask(IpszName, IpszSettings)
<b>Description</b>	Adds a Dynamic Task to a driver. A new Dynamic Task is created and its resources are allocated, but it will not be executed until a call to the Execute function is made. The task will remain allocated in the driver until removed by a call to RemoveDriverTask.
<b>Remarks</b>	

Parameter	Description
IpszName As String	Task Name. Must be unique.
IpszSettings As String	<p>String containing the Dynamic Settings for the task. Please refer to each driver's specific documentation for Dynamic Settings formats. This string is the same one inserted in the "Dynamic" variable's property.</p> <p>As a general rule, Dynamic Settings strings are a list of keywords separated by the " " (pipe) character. For instance, the Modbus driver may have:</p> <pre>[DRV]Modbus TCPIP.Sta=ModbusTCP Typ=10 Size=10 TaskType=0 Unit=1 FC=2 SA=20</pre> <p>where:</p> <p><b>[DRV]</b>: this is the parameter that indicates driver use and must be followed by driver's name, in the example this will be "Modbus TCPIP"  <b>Sta</b>: this is the parameter that indicated the name of the reference station in which task is created. In the example the station will be called "ModbusTCP"  <b>Typ</b>: indicates data type to be exchanged. The following values may be used in this parameter:</p> <ul style="list-style-type: none"> <li>0 = Bit</li> <li>1 = Sign Byte</li> <li>2 = Byte</li> <li>3 = Sign Word</li> <li>4 = Word</li> <li>5 = Sign DWord</li> <li>6 = DWord</li> <li>7 = Float</li> <li>8 = Double</li> <li>9 = String</li> <li>10 = Array</li> <li>11 = Structure</li> </ul> <p><b>Size</b>:this is the parameter that indicates the data size in bytes to be exchanged by the task  <b>TaskType</b>:this is the parameter that indicates type of task to be created. The following values may be used in this parameter:</p>

0 = Task Input  
 1 = Task Input/Output  
 2 = Task Exception Output  
 3 = Task Unconditional Output

**Unit:** this is the parameter that indicates the unit number. This parameter is only available for the Modbus driver only

**FC:** this is the parameter that indicated the function code to use. This parameter is only available for the Modbus driver only

**SA:** this is the start address parameter that indicates the start address from where data is to be read/written on device.

**Result** TaskInterface

**Example:**

Sub Main

```

Dim drv As DriverInterface
Dim Station As StationInterface
Dim TaskReset As TaskInterface
Dim sDynString As String
Dim vBuffer As Variant

'Create driver object
Set drv = GetDriverInterface("Modbus TCP/IP")
If drv Is Nothing Then
    MsgBox "Modbus TCP/IP driver not installed in the project!",
        vbExclamation, GetProjectTitle
    Exit Sub
End If

'Create station object
Set Station = drv.GetDriverStation("Station1")
If Station Is Nothing Then
    MsgBox "Station1 didn't found in the driver stations list!", vbExclamation,
        GetProjectTitle
    Exit Sub
End If

If Station.IsInError Then
    MsgBox "Station in Error!", vbExclamation, GetProjectTitle
    Exit Sub
End If

sDynString = "[DRV]Modbus
TCP/IP.Sta=Station1|Typ=4|Size=2|TaskType=2|Unit=1|FC=2|SA=100"

'Create task object
Set TaskReset = drv.AddDriverTask("Dyn1", sDynString)
If TaskReset Is Nothing Then
    MsgBox "the dynamic string '" & sDynString & "' is wrong!",
        vbExclamation, GetProjectTitle
    Exit Sub
End If

If Not TaskReset.IsValid Then
    MsgBox "AddDriverTask Failed!", vbExclamation, GetProjectTitle
    Exit Sub
End If

TaskReset.PollingTime = 0

Dim ret As Boolean

'Reset buffer value
    
```



```

vBuffer = 0

ret = TaskReset.SetWriteByteBuffer(vBuffer)

If ret = False Then
    MsgBox "SetWriteByteBuffer Failed!", vbExclamation, GetProjectTitle
    Exit Sub
End If

'Execute a sync write command
ret = TaskReset.Execute(5000)

If ret = False Then
    MsgBox "TaskReset.Execute Failed!", vbExclamation, GetProjectTitle
    Exit Sub
End If

'Remove task
ret = drv.RemoveDriverTask("Dyn1")

If ret = False Then
    MsgBox "RemoveDriverTask Failed!", vbExclamation, GetProjectTitle
    Exit Sub
End If

Set Station = Nothing
Set drv = Nothing
Set TaskReset = Nothing
End Sub

```

## 2.2. DelayEvents, DriverInterface Property

<b>Syntax</b>	DelayEvents = Long
<b>Description</b>	Gets or sets the delay for DriverInterface events.
<b>Remarks</b>	Only available from external COM clients. Not available from Supervisor BasicScripts.

Parameter	Description
None	None

**Result** Long  
Events delay interval in milliseconds.

**Example:**

```

Private Sub Form_Load()
    Dim INewVal As Long
    Dim bRet As Boolean
    Dim drv As Object

    Set drv = GetDriverInterface("ModbusTCP/IP")
    INewVal = drv.DelayEvents
    INewVal = INewVal + 100
    drv.DelayEvents = INewVal
End Sub

```

## 2.3. FireOnDataSent, DriverInterface Event

---

**Description** This event is fired when some data have been sent by a task. It can be used to retrieve new information without polling.

**Remarks** Only available from external COM clients. Not available from Supervisor BasicScripts.

Parameter	Description
(ByVal IpszTaskName As String)	The name of the task in which data have been sent.

**Example:**

## 2.4. FireOnNewData, DriverInterface Event

---

**Description** This event is fired when some data have changed on a task. It can be used to retrieve new information without polling.

**Remarks** Only available from external COM clients. Not available from Supervisor BasicScripts.

Parameter	Description
(ByVal IpszTaskName As String)	The name of the task in which data have changed.

**Example:**

```
Option Explicit
Dim WithEvents Drv As ModbusTCP/IP.DriverInterface
Dim task As ModbusTCP/IP.TaskInterface
Dim task2 As ModbusTCP/IP.TaskInterface
Dim var As Variant

Private Sub Drv_FireOnNewData(ByVal IpszTaskName As String)
    If IpszTaskName = "Dyn1" Then
        var = task.GetReadByteBuffer
        Label1.Caption = CStr(var)
    ElseIf IpszTaskName = "Dyn2" Then
        var = task2.GetReadByteBuffer
        Text2.Text = CStr(var)
    End If
End Sub
```

## 2.5. FireOnQualityChanged, DriverInterface Event

---

**Description** This event is fired when data quality has changed on a task. It can be used to retrieve new information without polling.

**Remarks** Only available from external COM clients. Not available from Supervisor BasicScripts.

Parameter	Description
(ByVal IpszTaskName As String)	The name of the task in which data have changed quality.

**Example:**

```

Dim WithEvents Drv As ModbusTCP/IP.DriverInterface
Dim task As ModbusTCP/IP.TaskInterface
Dim var As Variant

Private Sub Drv_FireOnQualityChanged(ByVal IpszTaskName As String)
    Dim Station As ModbusTCP/IP.StationInterface
    Set Station = task.GetStationObject
    lblQuality = CStr(Station.Quality)
    If Station.IsInError Then
        lblError = Station.LastErrorString
    Else
        lblError = "No Error"
    End If
End Sub

```

## 2.6. GetDriverStation, DriverInterface Function

---

**Syntax**      GetDriverStation(IpszStation)

**Description**      Returns a Station object.

**Remarks**

Parameter	Description
IpszStation As String	Station Name.

**Result**      Object  
If Function has been executed successfully it will retrieve an object of type StationInterface if otherwise Nothing is returned.

**Example:**

See the "**AddDriverTask**" function example.

## 2.7. GetDriverTask, DriverInterface Function

---

**Syntax**      GetDriverTask(IpszName)

**Description**      Returns a task object. This function can be used to retrieve the interface to any static or dynamic task.

**Remarks**

Parameter	Description
IpszName As String	Task Name.

**Result** Object  
 If Function has been executed successfully it will retrieve an object of type TaskInterface if otherwise Nothing is returned.

**Example:**

```

Sub Main
    Dim drv As DriverInterface
    Dim Station As StationInterface
    Dim task As TaskInterface
    Dim var As Variant
    Dim nVal As Integer
    Dim nVal2 As Integer
    Dim AppByte As Byte

    Set drv = GetDriverInterface("Modbus TCP/IP")
    Set task = drv.GetDriverTask("AsyncW")

    If task Is Nothing Then
        MsgBox "Async task not found"
        Exit Sub
    End If

    While Not This.IsStopping
        var = task.GetReadByteBuffer
        If (var(1) And &H80) > 0 Then
            AppByte = Not(var(0))
            nVal = AppByte + 1
            AppByte = Not(var(1))
            If nVal > 255 Then
                nVal2 = (((AppByte + 1) * 256) + (nVal And &HFF)) * -1
            Else
                nVal2 = ((AppByte * 256) + (nVal And &HFF)) * -1
            End If
            SetVariableValue("AsyncRead", nVal2)
        Else
            SetVariableValue("AsyncRead", var(0) + (var(1)*256))
        End If
        DoEvents
    Wend

    Set Station = Nothing
    Set drv = Nothing
    Set task = Nothing
End Sub
    
```

## 2.8. InitDriver, DriverInterface Function

**Syntax** InitDriver(IpszSerialNumber, IpszFileSettings)

**Description** Initialises a Communication Driver. Once this function has been called successfully, the driver is ready to operate.

**Remarks** Only available from external COM clients. Not available from Supervisor BasicScripts

Parameter	Description
-----------	-------------

IpszSerialNumber As String	Driver Serial Number. Contact Sails Office for a valid Serial Number for the Driver you want to use.
IpszFileSettings As String	<p>Path to a Directory containing the Driver Settings File. The file name must have the following format:</p> <p>&lt;DriverDLLName&gt;.drvsettings</p> <p>Where "DriverDLLName" is the name of the Dynamic Link Library for the Driver. The file is in XML format and can be written in 3 ways:</p> <ol style="list-style-type: none"> <li>1. Manually with a text editor</li> <li>2. By a previous call to the SetDriver function</li> <li>3. By using the Supervisor user interface (design mode)</li> </ol>

**Result** Boolean  
 True = the driver has been initialised successfully  
 False = driver initialization failed

```

Example:
Private Sub Form_Load()
    Dim bInitOk As Boolean
    Set Drv = CreateObject("SupervisorDriver.ModbusTCP.1")
    'Drv.SetDriver "C:\Projects\TestPrj\RESOURCES\TestPrj\"
    bInitOk = Drv.InitDriver("XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX",
    "C:\Projects\TestPrj\RESOURCES\TestPrj\")
    If Not bInitOk Then
        MsgBox "InitDriver failed!", vbExclamation, "ERROR"
    End If
End Sub
    
```

## 2.9. RemoveDriverTask, DriverInterface Function

**Syntax** RemoveDriverTask(IpszName)

**Description** Removes a Dynamic Task from a driver. A Dynamic Task is removed and its resources are freed.

**Remarks**

Parameter	Description
IpszName As String	Task Name.

**Result** Boolean  
 True = the function has been executed successfully  
 False = function execution failed

**Example:**  
 See the "AddDriverTask" function example.

## 2.10. SetDriver, DriverInterface Function

**Syntax** SetDriver(IpszFileSettings)

- Description** Displays a Property Sheet dialog box, by which the user can change a driver's General and Station settings.
- Remarks** Only available from external COM clients. Not available from Supervisor BasicScripts.  
This function must be called when the driver is not initialised, that is before InitDriver is called, otherwise it will fail.

Parameter	Description
lpszFileSettings As String	Path to a Directory containing the Driver Settings File. The file name must have the following format:  <DriverDLLName>.drvsettings  Where "DriverDLLName" is the name of the Dynamic Link Library for the Driver.

- Result** Boolean  
True = the function has been executed successfully  
False = function execution failed

**Example:**

```
Private Sub Form_Load()
    Dim bInitOk As Boolean
    Set Drv = CreateObject("SupervisorDriver.ModbusTCPIP.1")

    Dim bRet As Boolean
    bRet = Drv.SetDriver("C:\Projects\TestPrj\RESOURCES\TestPrj\")
    If Not bRet Then
        MsgBox " SetDriver failed!",vbExclamation,"ERROR"
    End If
End Sub
```

## 2.11. TerminateDriver, DriverInterface Function

---

- Syntax** TerminateDriver()
- Description** Terminates a Communication Driver previously initialised by the InitDriver function.
- Remarks** Only available from external COM clients. Not available from Supervisor BasicScripts

Parameter	Description
None	None

- Result** Boolean  
True = the driver has been terminated successfully  
False = driver termination failed

**Example:**

```
Private Sub Form_Load()  
    Dim bInitOk As Boolean  
    Set Drv = CreateObject("SupervisorDriver.ModbusTCP/IP.1")  
    'Drv.SetDriver "C:\Projects\TestPrj\RESOURCES\TestPrj\  
    bInitOk = Drv.InitDriver("XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX",  
    "C:\Projects\TestPrj\RESOURCES\TestPrj\  
    If Not bInitOk Then  
        MsgBox "InitDriver failed"  
    End If  
  
    If not Drv.TerminateDriver()  
        MsgBox "TerminateDriver failed!",vbExclamation,"ERROR"  
    End If  
End Sub
```





# 3. TaskInterface

---

## 3.1. AutoDelete, TaskInterface Property

---

**Syntax** Boolean = task.AutoDelete

**Description** Gets a task's AutoDelete flag.

**Remarks** This property is read-only.

Parameter	Description
None	None

**Result** Boolean  
True = AutoDelete enabled  
False = AutoDelete disabled

**Example:**

```
Sub Main
    Dim drv As DriverInterface
    Dim task As TaskInterface
    Dim bAutoDelete As Boolean

    Set drv = GetDriverInterface("Modbus TCP/IP")
    Set task = drv.GetDriverTask("Task")
    If task Is Nothing Then
        MsgBox "Task not found!", vbExclamation, "ERROR "
        Exit Sub
    End If

    bAutoDelete = task.AutoDelete
    MsgBox "AutoDelete = " & CStr(bAutoDelete), vbInformation, "ERROR "

    Set drv = Nothing
    Set task = Nothing
End Sub
```

## 3.2. ConditionalVariable, TaskInterface Property

---

**Syntax** Date = task.LastExecutionTime

**Description** Gets a task's conditional variable.

**Remarks** This property is read-only.

Parameter	Description
-----------	-------------

None	None
------	------

**Result**      Date

**Example:**

```

Sub Main
    Dim drv As DriverInterface
    Dim task As TaskInterface
    Dim dLastExec As Date

    Set drv = GetDriverInterface("Modbus TCP/IP")
    Set task = drv.GetDriverTask("Task")
    If task Is Nothing Then
        MsgBox "Task not found!", vbExclamation, "ERROR "
    Exit Sub
    End If

    dLastExec = task.LastExecutionTime
    MsgBox "LastExecutionTime = " & CStr(dLastExec), vbInformation, "ERROR "

    Set drv = Nothing
    Set task = Nothing
End Sub
    
```

### 3.3. Execute, TaskInterface Function

---

**Syntax**      Execute(dwTimeout)

**Description**      Executes a task. The Task must be previously be added by a call to "**AddDriverTask**" function.

**Remarks**

Parameter	Description
DwTimeout As Long	Time (milliseconds) to wait for task completion. If dwTimeout is zero, the function will return immediately and the task will be executed asynchronously. In both cases the task will be executed only once.

**Result**      Boolean  
 True = the function has been executed successfully  
 False = function execution failed

**Example:**  
 See the "**AddDriverTask**" function example.

### 3.4. GetFlatVariableList, TaskInterface Function

---

**Syntax**      GetFlatVariableList

**Description**      Gets the Variable List of a static task. As only static tasks have a Variable List, this function is useless with dynamic tasks.

**Remarks**

Parameter	Description
None	None

**Result** String  
A String containing the Variable List. The separator character can be retrieved by calling the "**GetVariableListSeparator**" function.

**Example:**

```
Sub Main
    Dim drv As DriverInterface
    Dim task As TaskInterface
    Dim strVarlist As String

    Set drv = GetDriverInterface("Modbus TCP/IP")
    Set task = drv.GetDriverTask("Task")
    If task Is Nothing Then
        MsgBox "Task not found!", vbExclamation, "ERROR "
        Exit Sub
    End If

    strVarlist = task.GetFlatVariableList
    MsgBox "GetFlatVariableList = " & strVarlist, vbInformation, "ERROR "

    Set drv = Nothing
    Set task = Nothing
End Sub
```

## 3.5. GetMaxByteSize, TaskInterface Function

**Syntax** GetMaxByteSize

**Description** Gets the maximum number of bytes that can be allocated for a task's read/write buffers.

**Remarks**

Parameter	Description
None	None

**Result** Long

**Example:**

```
Sub Main
    Dim drv As DriverInterface
    Dim task As TaskInterface
    Dim nSizeBytes As Long

    Set drv = GetDriverInterface("Modbus TCP/IP")
    Set task = drv.GetDriverTask("Task")
    If task Is Nothing Then
```

```

        MsgBox "Task not found!",vbExclamation,"ERROR "
    Exit Sub
End If

nSizeBytes = task.GetMaxByteSize
MsgBox "GetMaxByteSize = " & CStr(nSizeBytes),vbInformation,"ERROR "

Set drv = Nothing
Set task = Nothing
End Sub

```

### 3.6. GetReadByteBuffer, TaskInterface Function

**Syntax**            GetReadByteBuffer

**Description**      Returns the latest read values from the task. The Task must be previously be added by a call to "**AddDriverTask**" function and at least one call to the "**Execute**" function must have been made. If task is already present in the static task list of the project, is not necessary create an execute the task but it must only be referenced by a call to "**GetDriverTask**" function.

**Remarks**

Parameter	Description
None	None

**Result**            Variant  
 A Variant containing the last read values from the task. Usually the variable data type is dimensioned by the driver in base the data area to read. If The task is a static task then the function retrieve an Array of Byte containing the data.

**Example1:**  
 See the "**GetDriverTask**" function example.

```

Example2:
Sub Main
    Dim drv As DriverInterface
    Dim Station As StationInterface
    Dim task As TaskInterface
    Dim ret As Boolean
    Dim var As Variant

    Set drv = GetDriverInterface("Modbus TCP/IP")
    Set task = drv.AddDriverTask("Async", "Sta=Station1|Unit=1|FC=2|SA=2")
    If Not task.IsValid Then
        MsgBox "Task Async is not valid!",vbExclamation,"ERROR "
    Exit Sub
    End If

    task.PollingTime = 500

    While Not This.IsStopping
        ret = task.Execute(0)
        If ret = False Then
            MsgBox "task.Execute failed!",vbExclamation,"ERROR "
            Exit While
        End If
    End If

```

```

        var = task.GetReadByteBuffer
        SetVariableValue("AsyncRead", var)
        DoEvents
    Wend

    Set Station = Nothing
    Set drv = Nothing
    Set task = Nothing
End Sub

```

## 3.7. GetStationObject, TaskInterface Function

**Syntax**      GetStationObject

**Description**    Gets the interface to a tasks's station object

**Remarks**

Parameter	Description
None	None

**Result**      Object  
 If Function has been executed successfully it will retrieve an object of type StationInterface if otherwise Nothing is returned.

**Example:**

```

Sub Main
    Dim drv As DriverInterface
    Dim Station As StationInterface
    Dim task As TaskInterface

    'Create driver object
    Set drv = GetDriverInterface("Modbus TCP/IP")
    Set task = drv.GetDriverTask("Async")
    If task Is Nothing Then
        MsgBox "Async task not found!", vbExclamation, "ERROR "
        Exit Sub
    End If

    'Create station object
    Set Station = task.GetStationObject
    If Station.IsInError Then
        MsgBox "Station in Error!", vbExclamation, "ERROR "
        Exit Sub
    End If

    Set Station = Nothing
    Set drv = Nothing
    Set task = Nothing
End Sub

```

## 3.8. GetTotalSizeBytes, TaskInterface Function

---

**Syntax**            GetTotalSizeBytes

**Description**     Gets the number of bytes allocated for a task's read/write buffers.

**Remarks**         If bit variables (being the Project RealTimeDB variables) have been used in the task, the returned value is a bit number (the driver reserves a byte for each bit internally).

Parameter	Description
None	None

**Result**            Long

**Example:**

```

Sub Main
    Dim drv As DriverInterface
    Dim task As TaskInterface
    Dim nSizeBytes As Long

    Set drv = GetDriverInterface("Modbus TCPIP")
    Set task = drv.GetDriverTask("Task")
    If task Is Nothing Then
        MsgBox "Task not found!", vbExclamation, "ERROR "
        Exit Sub
    End If

    nSizeBytes = task.GetTotalSizeBytes
    MsgBox "GetTotalSizeBytes = " & CStr(nSizeBytes), vbInformation, "ERROR "

    Set drv = Nothing
    Set task = Nothing
End Sub
    
```

## 3.9. GetVariableListSeparator, TaskInterface Function

---

**Syntax**            GetVariableListSeparator

**Description**     Gets the Variable List Separator of a static task. As only static tasks have a Variable List, this function is useless with dynamic tasks.

**Remarks**

Parameter	Description
None	None

**Result**            String  
 A String containing the Variable List separator character (default is `;').

**Example:**

```

Sub Main
  Dim drv As DriverInterface
  Dim task As TaskInterface
  Dim strListSep As String

  Set drv = GetDriverInterface("Modbus TCP/IP")
  Set task = drv.GetDriverTask("Task")
  If task Is Nothing Then
    MsgBox "Task not found!", vbExclamation, "ERROR "
    Exit Sub
  End If

  strListSep = task.GetVariableListSeparator
  MsgBox "GetVariableListSeparator = " & strListSep, vbInformation, "ERROR "

  Set drv = Nothing
  Set task = Nothing
End Sub

```

## 3.10. GetXMLSettings, TaskInterface Function

**Syntax**      GetXMLSettings

**Description**      Gets all settings of a task in XML format.

**Remarks**

Parameter	Description
None	None

**Result**      String

**Example:**

```

Sub Main
  Dim drv As DriverInterface
  Dim task As TaskInterface
  Dim strGetXMLSettings As String

  Set drv = GetDriverInterface("Modbus TCP/IP")
  Set task = drv.GetDriverTask("Task")
  If task Is Nothing Then
    MsgBox "Task not found!", vbExclamation, "ERROR "
    Exit Sub
  End If

  strGetXMLSettings = task.GetXMLSettings
  MsgBox "GetXMLSettings = " & strGetXMLSettings, vbInformation, "ERROR "

  Set drv = Nothing
  Set task = Nothing
End Sub

```

## 3.11. HasBeenExecuted, TaskInterface Function

---

**Syntax** HasBeenExecuted

**Description** Returns information about whether a task has been successfully executed at least once.

**Remarks**

Parameter	Description
None	None

**Result** Boolean  
 True = the task has been successfully executed at least once  
 False = the task has never been executed

**Example:**

```

Sub Main
    Dim drv As DriverInterface
    Dim task As TaskInterface
    Dim bHasBeenExecuted As Boolean

    Set drv = GetDriverInterface("Modbus TCP/IP")
    Set task = drv.GetDriverTask("Task")
    If task Is Nothing Then
        MsgBox "Task not found!", vbExclamation, "ERROR "
        Exit Sub
    End If

    bHasBeenExecuted = task.HasBeenExecuted
    MsgBox "HasBeenExecuted = " & CStr(bHasBeenExecuted), vbInformation, "ERROR "

    Set drv = Nothing
    Set task = Nothing
End Sub
    
```

## 3.12. InUse, TaskInterface Property

---

**Syntax** Boolean = task.InUse

**Description** Gets a task's InUse flag.

**Remarks** This property is read-only.

Parameter	Description
None	None

**Result** Boolean  
 True = task in use  
 False = task not in use



**Example:**

```

Sub Main
    Dim drv As DriverInterface
    Dim task As TaskInterface
    Dim bInUse As Boolean

    Set drv = GetDriverInterface("Modbus TCP/IP")
    Set task = drv.GetDriverTask("Task")
    If task Is Nothing Then
        MsgBox "Task not found!", vbExclamation, "ERROR "
    Exit Sub
    End If

    bInUse = task.InUse
    MsgBox "InUse = " & CStr(bInUse), vbInformation, "ERROR "

    Set drv = Nothing
    Set task = Nothing
End Sub

```

## 3.13. IsValid, TaskInterface Function

**Syntax**      IsValid

**Description**      Returns information about whether a task is valid.

**Remarks**

Parameter	Description
None	None

**Result**      Boolean  
True = task is valid  
False = task is not valid

**Example:**

```

Sub Main
    Dim drv As DriverInterface
    Dim task As TaskInterface
    Dim bIsValid As Boolean

    Set drv = GetDriverInterface("Modbus TCP/IP")
    Set task = drv.GetDriverTask("Task")
    If task Is Nothing Then
        MsgBox "Task not found!", vbExclamation, "ERROR "
    Exit Sub
    End If

    bIsValid = task.IsValid
    MsgBox "IsValid = " & CStr(bIsValid), vbInformation, "ERROR "

    Set drv = Nothing
    Set task = Nothing
End Sub

```

## 3.14. LastExecutionTime, TaskInterface Property

---

**Syntax**           String = task.ConditionalVariable

**Description**     Gets a task's last execution time.

**Remarks**         This property is read-only.

Parameter	Description
None	None

**Result**           String

**Example:**

```

Sub Main
    Dim drv As DriverInterface
    Dim task As TaskInterface
    Dim strCondVar As String

    Set drv = GetDriverInterface("Modbus TCP/IP")
    Set task = drv.GetDriverTask("Task")
    If task Is Nothing Then
        MsgBox "Task not found!", vbExclamation, "ERROR "
        Exit Sub
    End If

    strCondVar = task.ConditionalVariable
    MsgBox "ConditionalVariable = " & strCondVar, vbInformation, "ERROR "

    Set drv = Nothing
    Set task = Nothing
End Sub
    
```

## 3.15. Name, TaskInterface Property

---

**Syntax**           String = task.Name

**Description**     Gets the name of a task.

**Remarks**         This property is read-only.

Parameter	Description
None	None

**Result**           String

**Example:**

```

Sub Main
    
```

```

Dim drv As DriverInterface
Dim task As TaskInterface
Dim strName As String

Set drv = GetDriverInterface("Modbus TCP/IP")
Set task = drv.GetDriverTask("Task")
If task Is Nothing Then
    MsgBox "Task not found!", vbExclamation, "ERROR "
    Exit Sub
End If

strName = task.Name
MsgBox "Task Name = " & strName, vbInformation, "ERROR "

Set drv = Nothing
Set task = Nothing
End Sub

```

## 3.16. PollingTime, TaskInterface Property

**Syntax**      Long = task.PollingTime

**Description**      Gets a task's in use Polling Time. The time is in Milliseconds.

**Remarks**

Parameter	Description
None	None

**Result**      Long

**Example:**

```

Sub Main
    Dim drv As DriverInterface
    Dim task As TaskInterface
    Dim IPollTime As Long

    Set drv = GetDriverInterface("Modbus TCP/IP")
    Set task = drv.GetDriverTask("Task")
    If task Is Nothing Then
        MsgBox "Task not found!", vbExclamation, "ERROR "
        Exit Sub
    End If

    IPollTime = task.PollingTime
    MsgBox "PollingTime = " & CStr(IPollTime), vbInformation, "ERROR "

    Set drv = Nothing
    Set task = Nothing
End Sub

```

### 3.17. PollingTimeNoInUse, TaskInterface Property

**Syntax** Long = task.PollingTimeNoInUse

**Description** Gets a task's "not-in-use" Polling Time. The time is in Milliseconds.

**Remarks**

Parameter	Description
None	None

**Result** Long

**Example:**

```

Sub Main
    Dim drv As DriverInterface
    Dim task As TaskInterface
    Dim IPollTime As Long

    Set drv = GetDriverInterface("Modbus TCP/IP")
    Set task = drv.GetDriverTask("Task")
    If task Is Nothing Then
        MsgBox "Task not found!", vbExclamation, "ERROR "
        Exit Sub
    End If

    IPollTime = task.PollingTimeNoInUse
    MsgBox "PollingTimeNoInUse = " & CStr(IPollTime), vbInformation, "ERROR "

    Set drv = Nothing
    Set task = Nothing
End Sub
    
```

### 3.18. SetWriteByteBuffer, TaskInterface Function

**Syntax** SetWriteByteBuffer(vBuffer)

**Description** Sets the task with values to write. The Task must be previously be added by a call to **"AddDriverTask"** function and at least one call to the **"Execute"** function must have been made. If task is already present in the static task list of the project, is not necessary create an execute the task but it must only be referenced by a call to **"GetDriverTask"** function.

**Remarks**

Parameter	Description
VBuffer As Variant	The buffer to be written

**Result** Boolean  
 True = the function has been executed successfully  
 False = function execution failed

**Example:**  
 See the "AddDriverTask" function example.

## 3.19. Station, TaskInterface Property

**Syntax** String = task.Station

**Description** Gets the name of the task's station.

**Remarks** This property is read-only.

Parameter	Description
None	None

**Result** String

**Example:**

```

Sub Main
    Dim drv As DriverInterface
    Dim task As TaskInterface
    Dim strStation As String

    Set drv = GetDriverInterface("Modbus TCP/IP")
    Set task = drv.GetDriverTask("Task")
    If task Is Nothing Then
        MsgBox "Task not found!", vbExclamation, "ERROR "
        Exit Sub
    End If

    strStation = task.Station
    MsgBox "Task Station = " & strStation, vbInformation, "ERROR "

    Set drv = Nothing
    Set task = Nothing
End Sub

```

## 3.20. SwapBytes, TaskInterface Property

**Syntax** Boolean = task.SwapBytes

**Description** Gets a task's SwapBytes flag.

**Remarks** This property is read-only.

Parameter	Description
-----------	-------------

None	None
------	------

**Result** Boolean  
 True = Swap Byte enabled  
 False = Swap Byte disabled

**Example:**

```

Sub Main
    Dim drv As DriverInterface
    Dim task As TaskInterface
    Dim bSwapBytes As Boolean

    Set drv = GetDriverInterface("Modbus TCP/IP")
    Set task = drv.GetDriverTask("Task")
    If task Is Nothing Then
        MsgBox "Task not found!", vbExclamation, "ERROR "
        Exit Sub
    End If

    bSwapBytes = task.SwapBytes
    MsgBox "SwapBytes = " & CStr(bSwapBytes), vbInformation, "ERROR "

    Set drv = Nothing
    Set task = Nothing
End Sub
    
```

## 3.21. SwapWords, TaskInterface Property

---

**Syntax** Boolean = task.SwapWords

**Description** Gets a task's SwapWords flag.

**Remarks** This property is read-only.

Parameter	Description
None	None

**Result** Boolean  
 True = Swap Word enabled  
 False = Swap Word disabled

**Example:**

```

Sub Main
    Dim drv As DriverInterface
    Dim task As TaskInterface
    Dim bSwapWords As Boolean

    Set drv = GetDriverInterface("Modbus TCP/IP")
    Set task = drv.GetDriverTask("Task")
    If task Is Nothing Then
        MsgBox "Task not found!", vbExclamation, "ERROR "
        Exit Sub
    End If

    bSwapBytes = task.SwapWords
    
```

```

MsgBox "SwapWords = " & CStr(bSwapWords),vbInformation,"ERROR "

Set drv = Nothing
Set task = Nothing
End Sub

```

## 3.22. ToString, TaskInterface Function

**Syntax** ToString

**Description** Gets the Dynamic Settings string of a task. Please refer to each driver's specific documentation for Dynamic Settings formats.

**Remarks**

Parameter	Description
None	None

**Result** String  
String containing the Dynamic Settings for the task. Please refer to each driver's specific documentation for Dynamic Settings formats.

**Example:**

```

Sub Main
    Dim drv As DriverInterface
    Dim task As TaskInterface
    Dim strToString As String

    Set drv = GetDriverInterface("Modbus TCP/IP")
    Set task = drv.GetDriverTask("Task")
    If task Is Nothing Then
        MsgBox "Task not found!",vbExclamation,"ERROR "
        Exit Sub
    End If

    strToString = task.ToString
    MsgBox "ToString = " & strToString,vbInformation,"ERROR "

    Set drv = Nothing
    Set task = Nothing
End Sub

```

## 3.23. Type, TaskInterface Property

**Syntax** Byte = task.Type

**Description** Gets a task's type.

**Remarks** This property is read-only.

Parameter	Description
None	None

**Result**      Byte  
 0 = Input Type  
 1 = Input/Output Type  
 2 = Exception Output Type  
 3 = Unconditional Output Type  
 -2147220992 = not defined

**Example:**

```
Sub Main
    Dim drv As DriverInterface
    Dim task As TaskInterface
    Dim nTaskType As Byte

    Set drv = GetDriverInterface("Modbus TCPIP")
    Set task = drv.GetDriverTask("Task")
    If task Is Nothing Then
        MsgBox "Task not found!",vbExclamation,"ERROR "
        Exit Sub
    End If

    nTaskType = task.Type
    MsgBox "Task Type = " & CStr(nTaskType),vbInformation,"ERROR "

    Set drv = Nothing
    Set task = Nothing
End Sub
```

## 3.24. WriteOutputsAtStartup, TaskInterface Property

---

**Syntax**      Boolean = task.WriteOutputsAtStartup

**Description**      This property returns the write settings of the task at project startup. Shows the value of the "Write Outputs at Startup" property. This only has meaning for "Input/Output" or "Exception Output" tasks.

**Remarks**      Read only property.

Parameter	Description
None	None

**Result**      Boolean

**Example:**

```
Sub Main
    Dim drv As DriverInterface
    Dim task As TaskInterface
    Set drv = GetDriverInterface("Modbus TCPIP")
    Set task = drv.GetDriverTask("Task")
    If Not task.IsValid Then
        MsgBox "Task is not valid!",vbExclamation,"ERROR "
        Exit Sub
    End If
    MsgBox "WriteOutputsAtStartup = " & task.WriteOutputsAtStartup,vbExclamation,"ERROR "
    Set drv = Nothing
    Set task = Nothing
End Sub
```



## 4. StationInterface

### 4.1. GetBaseStationInterface, StationInterface Function

**Syntax**      GetBaseStationInterface

**Description**    This function used in this interface return an object Nothing.

**Remarks**

Parameter	Description
None	None

**Result**      Nothing

**Example:**

### 4.2. GetSubStationInterface, StationInterface Function

**Syntax**      GetSubStationInterface

**Description**    Returns a SubStation object. If the used driver is serial the returned object will be a "**TAPIStationInterface**" type, if the used driver is TCPIP the returned object will be a "**RASStationInterface**" type.

**Remarks**

Parameter	Description
None	None

**Result**      Object  
                   TAPIStationInterface for serial drivers  
                   RASStationInterface for TCPIP drivers

**Example:**

```
Sub Main
    Dim drv As DriverInterface
    Dim station As StationInterface
    Dim drv1 As DriverInterface
    Dim station1 As StationInterface
    Dim SerialStation As SerialStationInterface
```

```

Dim TAPIStation As TAPIStationInterface
Dim TCPIPStation As TCPIPStationInterface
Dim RASStation As RASStationInterface

Set drv = GetDriverInterface("Modbus Serial")
Set drv1 = GetDriverInterface("Modbus TCPIP")

Set station = drv.GetDriverStation("Station1")
If station Is Nothing Then
    MsgBox "Station not found!",vbExclamation,"ERROR "
    Exit Sub
End If

Set TAPIStation = station.GetSubStationInterface
If TAPIStation Is Nothing Then
    MsgBox "TAPIStation not found!",vbExclamation,"ERROR "
    Exit Sub
End If

Set SerialStation = TAPIStation.GetBaseStationInterface
If SerialStation Is Nothing Then
    MsgBox "SerialStation not found!",vbExclamation,"ERROR "
    Exit Sub
End If

Set station1 = drv1.GetDriverStation("Station2")
If station1 Is Nothing Then
    MsgBox "Station1 not found!",vbExclamation,"ERROR "
    Exit Sub
End If

Set RASStation = station1.GetSubStationInterface
If RASStation Is Nothing Then
    MsgBox "RASStation not found!",vbExclamation,"ERROR "
    Exit Sub
End If

Set TCPIPStation = RASStation.GetBaseStationInterface
If TCPIPStation Is Nothing Then
    MsgBox "TCPIPStation not found!",vbExclamation,"ERROR "
    Exit Sub
End If

Set drv = Nothing
Set station = Nothing
Set drv1 = Nothing
Set station1 = Nothing
Set TAPIStation = Nothing
Set SerialStation = Nothing
Set RASStation = Nothing
Set TCPIPStation = Nothing
End Sub

```

### 4.3. GetTask, StationInterface Function

---

**Syntax**            GetTask(IpszTaskName)

**Description**     Returns a task object. This function can be used to retrieve the interface to any static or dynamic task.

**Remarks**

Parameter	Description
-----------	-------------

lpzTaskName As String	Nome del Task.
-----------------------	----------------

**Result** Object  
If Function has been executed successfully it will retrieve an object of type TaskInterface if otherwise Nothing is returned.

**Example:**

```
Sub Main
    Dim drv As DriverInterface
    Dim station As StationInterface
    Dim task As TaskInterface
    Dim bInUse As Boolean

    Set drv = GetDriverInterface("Modbus Serial")
    Set station = drv.GetDriverStation("Station1")
    If station Is Nothing Then
        MsgBox "Station not found!", vbExclamation, "ERROR "
        Exit Sub
    End If

    Set task = station.GetTask("Task")
    If task Is Nothing Then
        MsgBox "Task not found!", vbExclamation, "ERROR "
        Exit Sub
    End If

    bInUse = task.InUse
    MsgBox "InUse = " & CStr(bInUse), vbInformation, "ERROR "

    Set drv = Nothing
    Set station = Nothing
End Sub
```

## 4.4. GetXMLSettings, StationInterface Function

**Syntax** GetXMLSettings

**Description** Gets all settings of a station in XML format.

**Remarks**

Parameter	Description
None	None

**Result** String

**Example:**

```
Sub Main
    Dim drv As DriverInterface
    Dim station As StationInterface
    Dim strGetXMLSettings As String

    Set drv = GetDriverInterface("Modbus Serial")
    Set station = drv.GetDriverStation("Station1")
```

```

    If station Is Nothing Then
        MsgBox "Station not found!",vbExclamation,"ERROR "
        Exit Sub
    End If

    strGetXMLSettings = station.GetXMLSettings
    MsgBox "GetXMLSettings = " & strGetXMLSettings,vbInformation,"ERROR "

    Set drv = Nothing
    Set station = Nothing
End Sub

```

## 4.5. IsInError, StationInterface Function

---

**Syntax**      IsInError

**Description**      Returns information about whether a station is in error.

**Remarks**

Parameter	Description
None	None

**Result**      Boolean  
 True = station in error  
 False = Station ok

**Example:**

```

Sub Main
    Dim drv As DriverInterface
    Dim station As StationInterface
    Dim bError As Boolean

    Set drv = GetDriverInterface("Modbus Serial")
    Set station = drv.GetDriverStation("Station1")
    If station Is Nothing Then
        MsgBox "Station not found!",vbExclamation,"ERROR "
        Exit Sub
    End If

    bError = station.IsInError
    MsgBox "IsInError = " & CStr(bError),vbInformation,"ERROR "

    Set drv = Nothing
    Set station = Nothing
End Sub

```

## 4.6. LastErrorCode, StationInterface Property

---

**Syntax**      Long = task.LastErrorCode

**Description**      Gets a station's last error code.

**Remarks** This property is read-only.

Parameter	Description
None	None

**Result** Long

**Example:**

```
Sub Main
    Dim drv As DriverInterface
    Dim task As TaskInterface
    Dim ILastError As Long

    Set drv = GetDriverInterface("Modbus Serial")
    Set station = drv.GetDriverStation("Station1")
    If station Is Nothing Then
        MsgBox "Station not found!", vbExclamation, "ERROR "
        Exit Sub
    End If

    ILastError = station.LastErrorCode
    MsgBox "LastErrorCode = " & CStr(ILastError), vbInformation, "ERROR "

    Set drv = Nothing
    Set task = Nothing
End Sub
```

## 4.7. LastErrorString, StationInterface Property

**Syntax** String = task.LastErrorString

**Description** Gets a station's last error description string.

**Remarks** This property is read-only.

Parameter	Description
None	None

**Result** String

**Example:**

```
Sub Main
    Dim drv As DriverInterface
    Dim task As TaskInterface
    Dim ILastError As String

    Set drv = GetDriverInterface("Modbus Serial")
    Set station = drv.GetDriverStation("Station1")
    If station Is Nothing Then
        MsgBox "Station not found!", vbExclamation, "ERROR "
```

```

        Exit Sub
    End If

    ILastError = station.LastErrorString
    MsgBox "LastErrorString = " & CStr(ILastError),vbInformation,"ERROR "

    Set drv = Nothing
    Set task = Nothing
End Sub

```

## 4.8. Name, StationInterface Property

---

- Syntax**        String = task.Name
- Description**    Gets a station's name.
- Remarks**        This property is read-only.

Parameter	Description
None	None

**Result**        String

**Example:**

```

Sub Main
    Dim drv As DriverInterface
    Dim task As TaskInterface
    Dim strName As String

    Set drv = GetDriverInterface("Modbus Serial")
    Set station = drv.GetDriverStation("Station1")
    If station Is Nothing Then
        MsgBox "Station not found!",vbExclamation,"ERROR "
        Exit Sub
    End If

    strName = station.Name
    MsgBox "Name = " & strName,vbInformation,"ERROR "

    Set drv = Nothing
    Set task = Nothing
End Sub

```

## 4.9. Quality, StationInterface Property

---

- Syntax**        Integer = task.Quality
- Description**    Gets a station's quality state.
- Remarks**        This property is read-only.

Parameter	Description
-----------	-------------

None	None
------	------

**Result** Integer  
The returned values tally with the OPC specification quality values:

252 = OPC\_STATUS\_MASK  
3 = OPC\_LIMIT\_MASK  
0 = OPC\_QUALITY\_BAD  
64 = OPC\_QUALITY\_UNCERTAIN  
192 = OPC\_QUALITY\_GOOD  
4 = OPC\_QUALITY\_CONFIG\_ERROR  
8 = OPC\_QUALITY\_NOT\_CONNECTED  
12 = OPC\_QUALITY\_DEVICE\_FAILURE  
16 = OPC\_QUALITY\_SENSOR\_FAILURE  
20 = OPC\_QUALITY\_LAST\_KNOWN  
24 = OPC\_QUALITY\_COMM\_FAILURE  
28 = OPC\_QUALITY\_OUT\_OF\_SERVICE  
68 = OPC\_QUALITY\_LAST\_USABLE  
80 = OPC\_QUALITY\_SENSOR\_CAL  
84 = OPC\_QUALITY\_EGU\_EXCEEDED  
88 = OPC\_QUALITY\_SUB\_NORMAL  
216 = OPC\_QUALITY\_LOCAL\_OVERRIDE

**Example:**

```
Sub Main
    Dim drv As DriverInterface
    Dim task As TaskInterface
    Dim nQuality As Integer

    Set drv = GetDriverInterface("Modbus Serial")
    Set station = drv.GetDriverStation("Station1")
    If station Is Nothing Then
        MsgBox "Station not found!", vbExclamation, "ERROR "
        Exit Sub
    End If

    nQuality = station.Quality
    MsgBox "Quality = " & CStr(nQuality), vbInformation, "ERROR "

    Set drv = Nothing
    Set task = Nothing
End Sub
```

## 4.10. StationType, StationInterface Property

**Syntax** Byte = task.StationType

**Description** Gets a station's type.

**Remarks** This property is read-only.

Parameter	Description
None	None

**Result**      Byte  
0 = normal. This value is returned when the driver uses external library,  
such as Modbus TCP/IP, INTERBUS, etc.  
1 = SerialType  
2 = TAPIType  
3 = TCPIPType  
4 = RASTCPIPType  
-2147220992 = station not defined

**Example:**

```
Sub Main
    Dim drv As DriverInterface
    Dim task As TaskInterface
    Dim nStationType As Byte

    Set drv = GetDriverInterface("Modbus Serial")
    Set station = drv.GetDriverStation("Station1")
    If station Is Nothing Then
        MsgBox "Station not found!", vbExclamation, "ERROR "
        Exit Sub
    End If

    nStationType = station.StationType
    MsgBox "StationType = " & CStr(nStationType), vbInformation, "ERROR "

    Set drv = Nothing
    Set task = Nothing
End Sub
```



# 5. SerialStationInterface

---

## 5.1. BaudRate, SerialStationInterface Property

---

<b>Syntax</b>	Long = station.BaudRate
<b>Description</b>	This property sets or returns the COM Port BaudeRate value of the Station.
<b>Remarks</b>	A modification of this value will come acquired only to the following serial port opening.

Parameter	Description
None	None

**Result** Long

**Example:**

```
Sub Main
    Dim drv As DriverInterface
    Dim station As StationInterface
    Dim SerialStation As SerialStationInterface
    Dim TAPIStation As TAPIStationInterface
    Dim lBaudRate As Long
    Dim nParity As Byte
    Dim nPortID As Byte
    Dim nStopBits As Byte
    Dim nByteSize As Byte
    Dim nFlowControl As Byte

    Set drv = GetDriverInterface("Modbus Serial")
    Set station = drv.GetDriverStation("Station1")
    If station Is Nothing Then
        MsgBox "Station not found!", vbExclamation, "ERROR "
        Exit Sub
    End If

    Set TAPIStation = station.GetSubStationInterface
    If TAPIStation Is Nothing Then
        MsgBox "TAPIStation not found!", vbExclamation, "ERROR "
        Exit Sub
    End If

    Set SerialStation = TAPIStation.GetBaseStationInterface
    If SerialStation Is Nothing Then
        MsgBox "SerialStation not found!", vbExclamation, "ERROR "
        Exit Sub
    End If

    nPortID = SerialStation.PortID
    lBaudRate = SerialStation.BaudRate
    nByteSize = SerialStation.ByteSize
    nParity = SerialStation.Parity
    nStopBits = SerialStation.StopBits
```

```
nFlowControl = SerialStation.FlowControl

MsgBox "PortID = " & CStr(nPortID) & vbCrLf & _
      "BaudRate = " & CStr(lBaudRate) & vbCrLf & _
      "ByteSize = " & CStr(nByteSize) & vbCrLf & _
      "Parity = " & CStr(nParity) & vbCrLf & _
      "StopBits = " & CStr(nStopBits) & vbCrLf & _
      "FlowControl = " & CStr(nFlowControl),vbInformation,"ERROR "

Set drv = Nothing
Set station = Nothing
Set TAPIStation = Nothing
Set SerialStation = Nothing
End Sub
```

## 5.2. ByteSize, SerialStationInterface Property

---

**Syntax**      Byte = station.ByteSize

**Description**    This property sets or returns the COM Port ByteSize value of the Station.

**Remarks**      A modification of this value will come acquired only to the following serial port opening.

Parameter	Description
None	None

**Result**      Byte

**Example:**  
See the "**BaudRate**" property example.

## 5.3. CdTimeout, SerialStationInterface Property

---

**Syntax**      Long = station.CdTimeout

**Description**    This property sets or returns the Timeout for the COM Port Cd signal of the Station. This time is in milliseconds.

**Remarks**      A modification of this value will come acquired only to the following serial port opening.

Parameter	Description
None	None

**Result**      Long

**Example:**

See the "RxTimeout" property example.

## 5.4. CtsTimeout, SerialStationInterface Property

---

**Syntax**      Long = station.CtsTimeout

**Description**    This property sets or returns the Timeout for the COM Port Cts signal of the Station. This time is in milliseconds.

**Remarks**      A modification of this value will come acquired only to the following serial port opening.

Parameter	Description
None	None

**Result**      Long

**Example:**

See the "RxTimeout" property example.

## 5.5. DsrTimeout, SerialStationInterface Property

---

**Syntax**      Long = station.DsrTimeout

**Description**    This property sets or returns the Timeout for the COM Port Dsr signal of the Station. This time is in milliseconds.

**Remarks**      A modification of this value will come acquired only to the following serial port opening.

Parameter	Description
None	None

**Result**      Long

**Example:**

See the "RxTimeout" property example.

## 5.6. FlowControl, SerialStationInterface Property

---

**Syntax**      Byte = station.FlowControl

**Description** This property sets or returns the COM Port Flow Control value of the Station.

**Remarks** A modification of this value will come acquired only to the following serial port opening.

Parameter	Description
None	None

**Result** Byte  
 0 = None  
 1 = Hardware  
 2 = Xon / Xoff

**Example:**  
 See the "BaudRate" property example.

## 5.7. GetBaseStationInterface, SerialStationInterface Function

---

**Syntax** GetBaseStationInterface

**Description** This function used in this interface return an object "SerialStationInterface".

**Remarks**

Parameter	Description
None	None

**Result** Object  
 If Function has been executed successfully it will retrieve an object of type SerialStationInterface if , otherwise Nothing is returned.

**Example:**  
 Sub Main  
     Dim drv As DriverInterface  
     Dim station As StationInterface  
     Dim SerialStation As SerialStationInterface  
     Dim TAPISStation As TAPISStationInterface  
     Dim SubSerialStation As SerialStationInterface  
     Dim SubTAPISStation As TAPISStationInterface  
  
     Set drv = GetDriverInterface("Modbus Serial")  
     Set station = drv.GetDriverStation("Station1")  
     If station Is Nothing Then  
         MsgBox "Station not found!",vbExclamation,"ERROR "  
         Exit Sub  
     End If  
  
     Set TAPISStation = station.GetSubStationInterface  
     If TAPISStation Is Nothing Then  
         MsgBox "TAPISStation not found!",vbExclamation,"ERROR "  
         Exit Sub  
     End If

```

Set SerialStation = TAPIStation.GetBaseStationInterface
If SerialStation Is Nothing Then
    MsgBox "SerialStation not found!",vbExclamation,"ERROR "
Exit Sub
End If

Set SubSerialStation = SerialStation.GetBaseStationInterface
If SubSerialStation Is Nothing Then
    MsgBox "SubSerialStation not found!",vbExclamation,"ERROR "
Exit Sub
End If

Set SubTAPIStation = SerialStation.GetSubStationInterface
If SubTAPIStation Is Nothing Then
    MsgBox "SubTAPIStation not found!",vbExclamation,"ERROR "
Exit Sub
End If

Set drv = Nothing
Set station = Nothing
Set TAPIStation = Nothing
Set SerialStation = Nothing
Set SubSerialStation = Nothing
Set SubTAPIStation = Nothing
End Sub
    
```

## 5.8. GetSubStationInterface, SerialStationInterface Function

---

**Syntax**      GetSubStationInterface

**Description**    This function used in this interface return an object "**TAPIStationInterface**".

**Remarks**

Parameter	Description
None	None

**Result**      Object  
 If Function has been executed successfully it will retrieve an object of type TAPIStationInterface if , otherwise Nothing is returned.

**Example:**  
 See the "**GetBaseStationInterface**" property example.

## 5.9. GetXMLSettings, SerialStationInterface Function

---

**Syntax**      GetXMLSettings

**Description**    Gets all settings of a station in XML format.

**Remarks**

Parameter	Description
None	None

**Result**      String

**Example:**

```

Sub Main
    Dim drv As DriverInterface
    Dim station As StationInterface
    Dim SerialStation As SerialStationInterface
    Dim TAPIStation As TAPIStationInterface
    Dim strGetXMLSettings As String

    Set drv = GetDriverInterface("Modbus Serial")
    Set station = drv.GetDriverStation("Station1")
    If station Is Nothing Then
        MsgBox "Station not found!", vbExclamation, "ERROR "
        Exit Sub
    End If

    Set TAPIStation = station.GetSubStationInterface
    If TAPIStation Is Nothing Then
        MsgBox "TAPIStation not found!", vbExclamation, "ERROR "
        Exit Sub
    End If

    Set SerialStation = TAPIStation.GetBaseStationInterface
    If SerialStation Is Nothing Then
        MsgBox "SerialStation not found!", vbExclamation, "ERROR "
        Exit Sub
    End If

    strGetXMLSettings = SerialStation.GetXMLSettings
    MsgBox "GetXMLSettings = " & CStr(strGetXMLSettings), vbInformation, "ERROR "

    Set drv = Nothing
    Set station = Nothing
    Set TAPIStation = Nothing
    Set SerialStation = Nothing
End Sub
    
```

## 5.10. KeepPortOpened, SerialStationInterface Property

---

**Syntax**      Boolean = station.KeepPortOpened

**Description**      This property allow to keep the COM Port always opened during the project runtime. If the property is set to false the COM port will be opened and then will be closed each time that some data have to be exchanged.

**Remarks**      A modification of this value will come acquired only to the following serial port opening.

Parameter	Description
None	None

**Result** Boolean  
 True = COM port is keep opened  
 False = COM port is opened and closed each time some data have to be exchanged

**Example:**

```

Sub Main
    Dim drv As DriverInterface
    Dim station As StationInterface
    Dim SerialStation As SerialStationInterface
    Dim TAPIStation As TAPIStationInterface
    Dim bKeepPortOpened As Boolean

    Set drv = GetDriverInterface("Modbus Serial")
    Set station = drv.GetDriverStation("Station1")
    If station Is Nothing Then
        MsgBox "Station not found!",vbExclamation,"ERROR "
        Exit Sub
    End If

    Set TAPIStation = station.GetSubStationInterface
    If TAPIStation Is Nothing Then
        MsgBox "TAPIStation not found!",vbExclamation,"ERROR "
        Exit Sub
    End If

    Set SerialStation = TAPIStation.GetBaseStationInterface
    If SerialStation Is Nothing Then
        MsgBox "SerialStation not found!",vbExclamation,"ERROR "
        Exit Sub
    End If

    bKeepPortOpened = SerialStation.KeepPortOpened
    MsgBox "KeepPortOpened = " & CStr(bKeepPortOpened),vbInformation,"ERROR "

    Set drv = Nothing
    Set station = Nothing
    Set TAPIStation = Nothing
    Set SerialStation = Nothing
End Sub
    
```

## 5.11. Parity, SerialStationInterface Property

---

**Syntax** Byte = station.Parity

**Description** This property sets or returns the COM Port Parity value of the Station.

**Remarks** A modification of this value will come acquired only to the following serial port opening.

Parameter	Description
None	None

**Result** Byte  
 0 = None  
 1 = Odd  
 2 = Even

3 = Mark  
4 = Blank

**Example:**

See the "**BaudRate**" property example.

## 5.12. PortID, SerialStationInterface Property

---

**Syntax**      Byte = station.PortID

**Description**    This property sets or returns the COM Port number of the Station.

**Remarks**      A modification of this value will come acquired only to the following serial port opening.

Parameter	Description
None	None

**Result**      Byte  
0 = COM1  
1 = COM2  
2 = COM3  
3 = COM4  
.....

**Example:**

See the "**BaudRate**" property example.

## 5.13. RxTimeout, SerialStationInterface Property

---

**Syntax**      Long = station.RxTimeout

**Description**    This property sets or returns the Timeout for the COM Port Rx signal of the Station. This time is in milliseconds.

**Remarks**      A modification of this value will come acquired only to the following serial port opening.

Parameter	Description
None	None

**Result**      Long

**Example:**

```
Sub Main
    Dim drv As DriverInterface
    Dim station As StationInterface
    Dim SerialStation As SerialStationInterface
```



```

Dim TAPIStation As TAPIStationInterface
Dim IRxTimeout As Long
Dim ITxTimeout As Long
Dim ICdTimeout As Long
Dim ICTsTimeout As Long
Dim IDsrTimeout As Long

Set drv = GetDriverInterface("Modbus Serial")
Set station = drv.GetDriverStation("Station1")
If station Is Nothing Then
    MsgBox "Station not found!",vbExclamation,"ERROR "
    Exit Sub
End If

Set TAPIStation = station.GetSubStationInterface
If TAPIStation Is Nothing Then
    MsgBox "TAPIStation not found!",vbExclamation,"ERROR "
    Exit Sub
End If

Set SerialStation = TAPIStation.GetBaseStationInterface
If SerialStation Is Nothing Then
    MsgBox "SerialStation not found!",vbExclamation,"ERROR "
    Exit Sub
End If

IRxTimeout = SerialStation.RxTimeout
ITxTimeout = SerialStation.TxTimeout
ICdTimeout = SerialStation.CdTimeout
ICTsTimeout = SerialStation.CtsTimeout
IDsrTimeout = SerialStation.DsrTimeout

MsgBox "RxTimeout = " & CStr(IRxTimeout) & vbLf & _
    "TxTimeout = " & CStr(ITxTimeout) & vbLf & _
    "CdTimeout = " & CStr(ICdTimeout) & vbLf & _
    "CtsTimeout = " & CStr(ICTsTimeout) & vbLf & _
    "DsrTimeout = " & CStr(IDsrTimeout),vbInformation,"ERROR "

Set drv = Nothing
Set station = Nothing
Set TAPIStation = Nothing
Set SerialStation = Nothing
End Sub

```

## 5.14. SizeReceivingQueue, SerialStationInterface Property

---

- Syntax**            Long = station.SizeReceivingQueue
- Description**    This property sets or returns the Receiving Buffer Queue (Bytes) for the COM Port of the Station.
- Remarks**        A modification of this value will come acquired only to the following serial port opening.

Parameter	Description
None	None

**Result**            Long

**Example:**

```

Sub Main
    Dim drv As DriverInterface
    Dim station As StationInterface
    Dim SerialStation As SerialStationInterface
    Dim TAPISStation As TAPISStationInterface
    Dim ISizeReceivQueue As Long

    Set drv = GetDriverInterface("Modbus Serial")
    Set station = drv.GetDriverStation("Station1")
    If station Is Nothing Then
        MsgBox "Station not found!",vbExclamation,"ERROR "
        Exit Sub
    End If

    Set TAPISStation = station.GetSubStationInterface
    If TAPISStation Is Nothing Then
        MsgBox "TAPISStation not found!",vbExclamation,"ERROR "
        Exit Sub
    End If

    Set SerialStation = TAPISStation.GetBaseStationInterface
    If SerialStation Is Nothing Then
        MsgBox "SerialStation not found!",vbExclamation,"ERROR "
        Exit Sub
    End If

    ISizeReceivQueue = SerialStation.SizeReceivingQueue
    MsgBox "SizeReceivingQueue = " & CStr(ISizeReceivQueue),vbInformation,"ERROR "

    Set drv = Nothing
    Set station = Nothing
    Set TAPISStation = Nothing
    Set SerialStation = Nothing
End Sub
    
```

## 5.15. SizeTransmissionQueue, SerialStationInterface Property

---

- Syntax**            Long = station.SizeTransmissionQueue
  
- Description**    This property sets or returns the Transmission Buffer Queue (Bytes) for the COM Port of the Station.
  
- Remarks**        A modification of this value will come acquired only to the following serial port opening.

Parameter	Description
None	None

**Result**            Long

**Example:**

```

Sub Main
    Dim drv As DriverInterface
    Dim station As StationInterface
    Dim SerialStation As SerialStationInterface
    Dim TAPISStation As TAPISStationInterface
    
```

```

Dim ISizeTransQueue As Long

Set drv = GetDriverInterface("Modbus Serial")
Set station = drv.GetDriverStation("Station1")
If station Is Nothing Then
    MsgBox "Station not found!",vbExclamation,"ERROR "
Exit Sub
End If

Set TAPIStation = station.GetSubStationInterface
If TAPIStation Is Nothing Then
    MsgBox "TAPIStation not found!",vbExclamation,"ERROR "
Exit Sub
End If

Set SerialStation = TAPIStation.GetBaseStationInterface
If SerialStation Is Nothing Then
    MsgBox "SerialStation not found!",vbExclamation,"ERROR "
Exit Sub
End If

ISizeTransQueue = SerialStation.SizeTransmissionQueue
MsgBox "SizeTransmissionQueue = " & CStr(ISizeTransQueue),vbInformation,"ERROR "

Set drv = Nothing
Set station = Nothing
Set TAPIStation = Nothing
Set SerialStation = Nothing
End Sub
    
```

## 5.16. StopBits, SerialStationInterface Property

---

- Syntax**      Byte = station.StopBits
- Description**    This property sets or returns the COM Port Stop Bits value of the Station.
- Remarks**      A modification of this value will come acquired only to the following serial port opening.

Parameter	Description
None	None

- Result**      Byte  
                  0 = 1 stop bit  
                  1 = 1.5 stop bits  
                  2 = 2 stop bits

**Example:**  
 See the "**BaudRate**" property example.

## 5.17. TxTimeout, SerialStationInterface Property

---

- Syntax**      Long = station.TxTimeout

**Description** This property sets or returns the Timeout for the COM Port Tx signal of the Station. This time is in milliseconds.

**Remarks** A modification of this value will come acquired only to the following serial port opening.

<b>Parameter</b>	<b>Description</b>
None	None

**Result** Long

**Example:**  
See the "**RxTimeout**" property example.

# 6. TAPIStationInterface

---

## 6.1. DisconnectAfterSecs, TAPIStationInterface Property

---

**Syntax** Long = station.DisconnectAfterSecs

**Description** This property sets or returns the time in milliseconds after which the modem connection of the TAPI Station will hang up. The counter of the time will start from the moment in which all the communication tasks are not in use, and it will be reset if at least one task returns in use before the expiration of the time.

**Remarks** A modification of this value will come acquired only to the following call.

Parameter	Description
None	None

**Result** Long

**Example:**

Sub Main

```
Dim drv As DriverInterface
Dim station As StationInterface
Dim TAPIStation As TAPIStationInterface
Dim strPhoneNumber As String
Dim nRetries As Byte
Dim lDisconAfter As Long
Dim lRetryAfter As Long
Dim bEnableTAPI As Boolean
Dim bPrompt As Boolean
Dim bShowCon As Boolean

Set drv = GetDriverInterface("Modbus Serial")
Set station = drv.GetDriverStation("Station1")
If station Is Nothing Then
    MsgBox "Station not found!",vbExclamation,"ERROR "
    Exit Sub
End If

Set TAPIStation = station.GetSubStationInterface
If TAPIStation Is Nothing Then
    MsgBox "TAPIStation not found!",vbExclamation,"ERROR "
    Exit Sub
End If

strPhoneNumber = TAPIStation.PhoneNumber
nRetries = TAPIStation.Retries
lDisconAfter = TAPIStation.DisconnectAfterSecs
lRetryAfter = TAPIStation.RetryAfterSecs
bEnableTAPI = TAPIStation.EnableTAPICallOnThisStation
bPrompt = TAPIStation.PromptForConnection
bShowCon = TAPIStation.ShowConnectionDlg
```

```

MsgBox "PhoneNumber = " & CStr(strPhoneNumber) & vbCrLf & _
      "Retries = " & CStr(nRetries) & vbCrLf & _
      "DisconnectAfterSecs = " & CStr(IDisconAfter) & vbCrLf & _
      "RetryAfterSecs = " & CStr(IRetryAfter) & vbCrLf & _
      "EnableTAPICallOnThisStation = " & CStr(bEnableTAPI) & vbCrLf & _
      "PromptForConnection = " & CStr(bPrompt) & vbCrLf & _
      "DsrTimShowConnectionDlgeout = " & CStr(bShowCon),vbInformation,"ERROR "

Set drv = Nothing
Set station = Nothing
Set TAPIStation = Nothing
End Sub

```

## 6.2. EnableTAPICallOnThisStation, TAPIStationInterface Property

---

- Syntax** Boolean = station.DisconnectAfterSecs
- Description** This property sets or returns the enable to execute the modem connection for the Station.
- Remarks** A modification of this value will come acquired only to the following call.

Parameter	Description
None	None

- Result** Boolean  
 True = TAPI call enable  
 False = TAPI call disable

**Example:**  
 See the "DisconnectAfterSecs" property example.

## 6.3. EndConnectionTime, TAPIStationInterface Property

---

- Syntax** Date = station.EndConnectionTime
- Description** This property returns the date and time of end of the last modem connection of the Station.
- Remarks** This property is read-only.

Parameter	Description
None	None

- Result** Date

**Example:**  
 See the "StartConnectionTime" property example.

## 6.4. GetBaseStationInterface, TAPIStationInterface Function

**Syntax**      GetBaseStationInterface

**Description**    This function used in this interface return an object "**SerialStationInterface**".

**Remarks**

Parameter	Description
None	None

**Result**          Object  
If Function has been executed successfully it will retrieve an object of type SerialStationInterface if , otherwise Nothing is returned.

**Example:**

```
Sub Main
    Dim drv As DriverInterface
    Dim station As StationInterface
    Dim SerialStation As SerialStationInterface
    Dim TAPIStation As TAPIStationInterface
    Dim SubTAPIStation As TAPIStationInterface

    Set drv = GetDriverInterface("Modbus Serial")
    Set station = drv.GetDriverStation("Station1")
    If station Is Nothing Then
        MsgBox "Station not found!",vbExclamation,"ERROR "
        Exit Sub
    End If

    Set TAPIStation = station.GetSubStationInterface
    If TAPIStation Is Nothing Then
        MsgBox "TAPIStation not found!",vbExclamation,"ERROR "
        Exit Sub
    End If

    Set SerialStation = TAPIStation.GetBaseStationInterface
    If SerialStation Is Nothing Then
        MsgBox "SerialStation not found!",vbExclamation,"ERROR "
        Exit Sub
    End If

    Set SubTAPIStation = TAPIStation.GetSubStationInterface
    If SubTAPIStation Is Nothing Then
        MsgBox "SubTAPIStation not found!",vbExclamation,"ERROR "
        Exit Sub
    End If

    Set drv = Nothing
    Set station = Nothing
    Set TAPIStation = Nothing
    Set SerialStation = Nothing
    Set SubTAPIStation = Nothing
End Sub
```

## 6.5. GetSubStationInterface, TAPIStationInterface Function

---

**Syntax**            GetSubStationInterface

**Description**    This function used in this interface return an object "TAPIStationInterface".

**Remarks**

Parameter	Description
None	None

**Result**            Object  
 If Function has been executed successfully it will retrieve an object of type TAPIStationInterface if , otherwise Nothing is returned.

**Example:**  
 See the "GetBaseStationInterface" property example.

## 6.6. GetXMLSettings, TAPIStationInterface Function

---

**Syntax**            GetXMLSettings

**Description**    Gets all settings of a station in XML format.

**Remarks**

Parameter	Description
None	None

**Result**            String

**Example:**  
 Sub Main  
     Dim drv As DriverInterface  
     Dim station As StationInterface  
     Dim TAPIStation As TAPIStationInterface  
     Dim strGetXMLSettings As String  
  
     Set drv = GetDriverInterface("Modbus Serial")  
     Set station = drv.GetDriverStation("Station1")  
     If station Is Nothing Then  
         MsgBox "Station not found!", vbExclamation, "ERROR "  
         Exit Sub  
     End If  
  
     Set TAPIStation = station.GetSubStationInterface  
     If TAPIStation Is Nothing Then  
         MsgBox "TAPIStation not found!", vbExclamation, "ERROR "



```

        Exit Sub
    End If

    strGetXMLSettings = TAPIStation.GetXMLSettings
    MsgBox "GetXMLSettings = " & CStr(strGetXMLSettings),vbInformation,"ERROR "

    Set drv = Nothing
    Set station = Nothing
    Set TAPIStation = Nothing
End Sub

```

## 6.7. IsConnected, TAPIStationInterface Function

**Syntax** IsConnected

**Description** Gets the connection state of the RAS Station.

**Remarks**

Parameter	Description
None	None

**Result** Boolean  
 True = Station Connected  
 False = Station not Connected

**Example:**

```

Sub Main
    Dim drv As DriverInterface
    Dim station As StationInterface
    Dim TAPIStation As TAPIStationInterface
    Dim bIsConnected As Boolean

    Set drv = GetDriverInterface("Modbus Serial")
    Set station = drv.GetDriverStation("Station1")
    If station Is Nothing Then
        MsgBox "Station not found!",vbExclamation,"ERROR "
        Exit Sub
    End If

    Set TAPIStation = station.GetSubStationInterface
    If TAPIStation Is Nothing Then
        MsgBox "TAPIStation not found!",vbExclamation,"ERROR "
        Exit Sub
    End If

    bIsConnected = TAPIStation.IsConnected
    MsgBox "IsConnected = " & CStr(bIsConnected),vbInformation,"ERROR "

    Set drv = Nothing
    Set station = Nothing
    Set TAPIStation = Nothing
End Sub

```

## 6.8. LastConnectionTime, TAPIStationInterface Property

---

- Syntax**            Date = station.LastConnectionTime
- Description**      This property returns the connection time of the last modem call of the Station.
- Remarks**          This property is read-only.

Parameter	Description
None	None

**Result**            Date

**Example:**  
See the "**StartConnectionTime**" property example.

## 6.9. LastTAPIError, TAPIStationInterface Property

---

- Syntax**            Long = station.LastTAPIError
- Description**      This property returns the last error code occurs during the modem call of the Station.
- Remarks**          This property is read-only.

Parameter	Description
None	None

**Result**            Long

**Example:**  
See the "**StartConnectionTime**" property example.

## 6.10. LastTAPIErrorString, TAPIStationInterface Property

---

- Syntax**            String = station.LastTAPIErrorString
- Description**      This property returns a description text about the state of the modem call of the Station. This string can be an error message or a state connection message.
- Remarks**          This property is read-only.

Parameter	Description
None	None

**Result** String

**Example:**

See the "**StartConnectionTime**" property example.

## 6.11. PhoneNumber, TAPIStationInterface Property

**Syntax** String = station.PhoneNumber

**Description** This property sets or returns the telephone number to call.

**Remarks** A modification of this value will come acquired only to the following call.

Parameter	Description
None	None

**Result** String

**Example:**

See the "**DisconnectAfterSecs**" property example.

## 6.12. PromptForConnection, TAPIStationInterface Property

**Syntax** Boolean = station.PromptForConnection

**Description** This property allows you to show or hide the connection confirmation dialog window. If this property is true, before start connection, Supervisor will show a confirmation dialog window. Only if the user will click the OK button Supervisor will start the connection.

**Remarks** A modification of this value will come acquired only to the following call.

Parameter	Description
None	None

**Result** Boolean  
 True = show the dialog window before connecting  
 False = don't show the dialog window and start connection

**Example:**

See the "DisconnectAfterSecs" property example.

## 6.13. Retries, TAPIStationInterface Property

---

- Syntax**      Byte = station.Retries
- Description**    This property sets or returns the number of retries to execute for the modem call of the Station when the connection fails.
- Remarks**      A modification of this value will come acquired only to the following call.

Parameter	Description
None	None

**Result**      Byte

**Example:**

See the "DisconnectAfterSecs" property example.

## 6.14. RetryAfterSecs, TAPIStationInterface Property

---

- Syntax**      Long = station.RetryAfterSecs
- Description**    This property sets or returns the time in milliseconds after which the modem connection will be retried if the previous call has failed.
- Remarks**      A modification of this value will come acquired only to the following call.

Parameter	Description
None	None

**Result**      Long

**Example:**

See the "DisconnectAfterSecs" property example.

## 6.15. ShowConnectionDlg, TAPIStationInterface Property

- Syntax** Boolean = station.ShowConnectionDlg
- Description** This property allows you to show or hide the connection status dialog window.
- Remarks** A modification of this value will come acquired only to the following call.

Parameter	Description
None	None

- Result** Boolean  
True = show dialog window  
False = hide dialog window

**Example:**  
See the "DisconnectAfterSecs" property example.

## 6.16. StartConnectionTime, TAPIStationInterface Property

- Syntax** Date = station.StartConnectionTime
- Description** This property returns the date and time of begin of the last modem connection of the Station.
- Remarks** This property is read-only.

Parameter	Description
None	None

- Result** Date

**Example:**

```
Sub Main
    Dim drv As DriverInterface
    Dim station As StationInterface
    Dim TAPIStation As TAPIStationInterface
    Dim dStartConn As Date
    Dim dEndConn As Date
    Dim dTotalConn As Date
    Dim dLastConn As Date
    Dim lLastError As Long
    Dim strLastErrorString As String

    Set drv = GetDriverInterface("Modbus Serial")
    Set station = drv.GetDriverStation("Station1")
    If station Is Nothing Then
        MsgBox "Station not found!", vbExclamation, "ERROR "
    End If
    Exit Sub
End Sub
```

```

End If

Set TAPIStation = station.GetSubStationInterface
If TAPIStation Is Nothing Then
    MsgBox "TAPIStation not found!",vbExclamation,"ERROR "
    Exit Sub
End If

dStartConn = TAPIStation.StartConnectionTime
dEndConn = TAPIStation.EndConnectionTime
dTotalConn = TAPIStation.TotalConnectionTime
dLastConn = TAPIStation.LastConnectionTime
lLastError = TAPIStation.LastTAPIError
strLastErrorString = TAPIStation.LastTAPIErrorString

MsgBox "StartConnectionTime = " & CStr(dStartConn) & vbLf & _
"EndConnectionTime = " & CStr(dEndConn) & vbLf & _
"TotalConnectionTime = " & CStr(dTotalConn) & vbLf & _
"LastConnectionTime = " & CStr(dLastConn) & vbLf & _
"LastTAPIError = " & CStr(lLastError) & vbLf & _
"LastTAPIErrorString = " & CStr(strLastErrorString),vbInformation,"ERROR "

Set drv = Nothing
Set station = Nothing
Set TAPIStation = Nothing
End Sub

```

## 6.17. TotalConnectionTime, TAPIStationInterface Property

---

- Syntax**      Date = station.TotalConnectionTime
- Description**      This property returns the total connection time of the modem call of the Station. The time is the sum of all calls executed from the start of the Project.
- Remarks**      This property is read-only.

Parameter	Description
None	None

**Result**      Date

**Example:**  
 See the "**StartConnectionTime**" property example.

# 7. TCPIPStationInterface

---

## 7.1. BackupServerAddress, TCPIPStationInterface Property

---

**Syntax** String = station.BackupServerAddress

**Description** This property sets or returns the IP address of the Backup Server for the station. Shows the value of the "Backup Server Address" property.

**Remarks**

Parameter	Description
None	None

**Result** String

**Example:**

```
Sub Main
    Dim drv As DriverInterface
    Dim station As StationInterface
    Dim RASStation As RASStationInterface
    Dim TCPIPStation As TCPIPStationInterface

    Set drv = GetDriverInterface("Modbus TCPIP")
    Set station = drv.GetDriverStation("Station1")
    If station Is Nothing Then
        MsgBox "Station not found!",vbExclamation,"ERROR "
        Exit Sub
    End If

    Set RASStation = station.GetSubStationInterface
    If RASStation Is Nothing Then
        MsgBox "RASStation not found!",vbExclamation,"ERROR "
        Exit Sub
    End If

    Set TCPIPStation = RASStation.GetBaseStationInterface
    If TCPIPStation Is Nothing Then
        MsgBox "TCPIPStation not found!",vbExclamation,"ERROR "
        Exit Sub
    End If

    TCPIPStation.BackupServerAddress = "192.168.0.10"
    TCPIPStation.SwitchServerTimeout = 20000

    MsgBox "BackupServerAddress = " & TCPIPStation.BackupServerAddress & vbLf & _
        "SwitchServerTimeout = " &
        TCPIPStation.SwitchServerTimeout,vbExclamation,"ERROR "
    Set drv = Nothing
    Set station = Nothing
    Set RASStation = Nothing
    Set TCPIPStation = Nothing
End Sub
```

## 7.2. GetBaseStationInterface, TCPIPStationInterface Function

**Syntax** GetBaseStationInterface

**Description** This function used in this interface return an object "TCPIPStationInterface".

**Remarks**

Parameter	Description
None	None

**Result** Object  
 If Function has been executed successfully it will retrieve an object of type TCPIPStationInterface if , otherwise Nothing is returned.

**Example:**

```

Sub Main
    Dim drv As DriverInterface
    Dim station As StationInterface
    Dim RASStation As RASStationInterface
    Dim SubRASStation As RASStationInterface
    Dim TCPIPStation As TCPIPStationInterface
    Dim SubTCPIPStation As TCPIPStationInterface

    Set drv = GetDriverInterface("Modbus TCPIP")
    Set station = drv.GetDriverStation("Station1")
    If station Is Nothing Then
        MsgBox "Station not found!",vbExclamation,"ERROR "
        Exit Sub
    End If

    Set RASStation = station.GetSubStationInterface
    If RASStation Is Nothing Then
        MsgBox "RASStation not found!",vbExclamation,"ERROR "
        Exit Sub
    End If

    Set TCPIPStation = RASStation.GetBaseStationInterface
    If TCPIPStation Is Nothing Then
        MsgBox "TCPIPStation not found!",vbExclamation,"ERROR "
        Exit Sub
    End If

    Set SubTCPIPStation = TCPIPStation.GetBaseStationInterface
    If SubTCPIPStation Is Nothing Then
        MsgBox "SubTCPIPStation not found!",vbExclamation,"ERROR "
        Exit Sub
    End If

    Set SubRASStation = TCPIPStation.GetSubStationInterface
    If SubRASStation Is Nothing Then
        MsgBox "SubRASStation not found!",vbExclamation,"ERROR "
        Exit Sub
    End If

    Set drv = Nothing
    Set station = Nothing
    
```



```

Set RASStation = Nothing
Set TCPIPStation = Nothing
Set SubTCPIPStation = Nothing
Set SubRASStation = Nothing
End Sub
    
```

### 7.3. GetSubStationInterface, TCPIPStationInterface Function

**Syntax**      GetSubStationInterface

**Description**    This function used in this interface return an object "RASStationInterface".

**Remarks**

Parameter	Description
None	None

**Result**      Object  
 If Function has been executed successfully it will retrieve an object of type RASStationInterface if , otherwise Nothing is returned.

**Example:**  
 See the "GetBaseStationInterface" property example.

### 7.4. GetXMLSettings, TCPIPStationInterface Function

**Syntax**      GetXMLSettings

**Description**    Gets all settings of a station in XML format.

**Remarks**

Parameter	Description
None	None

**Result**      String

**Example:**  
 Sub Main  
     Dim drv As DriverInterface  
     Dim station As StationInterface  
     Dim RASStation As RASStationInterface  
     Dim TCPIPStation As TCPIPStationInterface  
     Dim strGetXMLSettings As String  
  
     Set drv = GetDriverInterface("Modbus TCPIP")

```

Set station = drv.GetDriverStation("Station1")
If station Is Nothing Then
    MsgBox "Station not found!",vbExclamation,"ERROR "
Exit Sub
End If

Set RASStation = station.GetSubStationInterface
If RASStation Is Nothing Then
    MsgBox "RASStation not found!",vbExclamation,"ERROR "
Exit Sub
End If

Set TCPIPStation = RASStation.GetBaseStationInterface
If TCPIPStation Is Nothing Then
    MsgBox "TCPIPStation not found!",vbExclamation,"ERROR "
Exit Sub
End If

strGetXMLSettings = TCPIPStation.GetXMLSettings
MsgBox "GetXMLSettings = " & CStr(strGetXMLSettings),vbInformation,"ERROR "

Set drv = Nothing
Set station = Nothing
Set TAPIStation = Nothing
Set SerialStation = Nothing
End Sub

```

## 7.5. LocalBoundAddress, TCPIPStationInterface Property

---

- Syntax**      String = station.LocalBoundAddress
- Description**      This property sets or returns the local IP Address that the station will use for the connection. This property is useful when on the PC are present two or more network cards. If the field comes lets voids the IP address of the default card will be used.
- Remarks**      A modification of this value will come acquired only when the socket will open. For example on the project startup before that the communication tasks are executed at least once.

Parameter	Description
None	None

**Result**      String

**Example:**  
See the "**ServerAddress**" property example.

## 7.6. LocalBoundPort, TCPIPStationInterface Property

---

- Syntax**      long = station.LocalBoundPort
- Description**      This property sets or returns the local Port that the station will use for the connection. If the field comes lets voids the default Port will be used.

**Remarks**      A modification of this value will come acquired only when the socket will open. For example on the project startup before that the communication tasks are executed at least once.

Parameter	Description
None	None

**Result**      Long

**Example:**  
See the "**ServerAddress**" property example.

## 7.7. MaxReceive, TCPIPStationInterface Property

---

**Syntax**      Long = station.MaxReceive

**Description**      This property sets or returns the Receiving Buffer Queue (Bytes) for the Station.

**Remarks**      A modification of this value will come acquired only when the socket will open. For example on the project startup before that the communication tasks are executed at least once.

Parameter	Description
None	None

**Result**      Long

**Example:**  
Sub Main  
    Dim drv As DriverInterface  
    Dim station As StationInterface  
    Dim RASStation As RASStationInterface  
    Dim TCPIPStation As TCPIPStationInterface  
    Dim IMaxReceive As Long  
    Dim IMaxSend As Long  
  
    Set drv = GetDriverInterface("Modbus TCPIP")  
    Set station = drv.GetDriverStation("Station1")  
    If station Is Nothing Then  
        MsgBox "Station not found!", vbExclamation, "ERROR "  
        Exit Sub  
    End If  
  
    Set RASStation = station.GetSubStationInterface  
    If RASStation Is Nothing Then  
        MsgBox "RASStation not found!", vbExclamation, "ERROR "  
        Exit Sub  
    End If  
  
    Set TCPIPStation = RASStation.GetBaseStationInterface  
    If TCPIPStation Is Nothing Then  
        MsgBox "TCPIPStation not found!", vbExclamation, "ERROR "

```

        Exit Sub
    End If

    IMaxReceive = TCPIPStation.MaxReceive
    IMaxSend = TCPIPStation.MaxSend

    MsgBox "MaxReceive = " & CStr(IMaxReceive) & vbLf & _
        "MaxSend = " & CStr(IMaxSend),vbInformation,"ERROR "

    Set drv = Nothing
    Set station = Nothing
    Set RASStation = Nothing
    Set TCPIPStation = Nothing
End Sub

```

## 7.8. MaxSend, TCPIPStationInterface Property

---

- Syntax**      Long = station.MaxSend
- Description**    This property sets or returns the Sending Buffer Queue (Bytes) for the Station.
- Remarks**      A modification of this value will come acquired only when the socket will open. For example on the project startup before that the communication tasks are executed at least once.

Parameter	Description
None	None

**Result**      Long

**Example:**  
See the "**MaxReceive**" property example.

## 7.9. RxTimeout, TCPIPStationInterface Property

---

- Syntax**      Long = station.RxTimeout
- Description**    This property sets or returns the Timeout for the data receiving. This time is in milliseconds.
- Remarks**      A modification of this value will come acquired only when the socket will open. For example on the project startup before that the communication tasks are executed at least once.

Parameter	Description
None	None

**Result**      Long

**Example:**

```

Sub Main
    Dim drv As DriverInterface
    Dim station As StationInterface
    Dim RASStation As RASStationInterface
    Dim TCPIPStation As TCPIPStationInterface
    Dim IRxTimeout As Long
    Dim ITxTimeout As Long

    Set drv = GetDriverInterface("Modbus TCPIP")
    Set station = drv.GetDriverStation("Station1")
    If station Is Nothing Then
        MsgBox "Station not found!", vbExclamation, "ERROR "
        Exit Sub
    End If

    Set RASStation = station.GetSubStationInterface
    If RASStation Is Nothing Then
        MsgBox "RASStation not found!", vbExclamation, "ERROR "
        Exit Sub
    End If

    Set TCPIPStation = RASStation.GetBaseStationInterface
    If TCPIPStation Is Nothing Then
        MsgBox "TCPIPStation not found!", vbExclamation, "ERROR "
        Exit Sub
    End If

    IRxTimeout = TCPIPStation.RxTimeout
    ITxTimeout = TCPIPStation.TxTimeout

    MsgBox "RxTimeout = " & CStr(IRxTimeout) & vbLf & _
        "TxTimeout = " & CStr(ITxTimeout), vbInformation, "ERROR "

    Set drv = Nothing
    Set station = Nothing
    Set RASStation = Nothing
    Set TCPIPStation = Nothing
End Sub
    
```

## 7.10. ServerAddress, TCPIPStationInterface Property

---

**Syntax**            String = station.ServerAddress

**Description**      This property sets or returns the Name or the IP Address of the Server to which the station will be connected.

**Remarks**            A modification of this value will come acquired only when the socket will open. For example on the project startup before that the communication tasks are executed at least once.

Parameter	Description
None	None

**Result**            String

**Example:**  
Sub Main

```

Dim drv As DriverInterface
Dim station As StationInterface
Dim RASStation As RASStationInterface
Dim TCPIPStation As TCPIPStationInterface
Dim strServerAddress As String
Dim lServerPort As Long
Dim strLocalBoundAddress As String
Dim lLocalBoundPort As Long

Set drv = GetDriverInterface("Modbus TCPIP")
Set station = drv.GetDriverStation("Station1")
If station Is Nothing Then
    MsgBox "Station not found!",vbExclamation,"ERROR "
    Exit Sub
End If

Set RASStation = station.GetSubStationInterface
If RASStation Is Nothing Then
    MsgBox "RASStation not found!",vbExclamation,"ERROR "
    Exit Sub
End If

Set TCPIPStation = RASStation.GetBaseStationInterface
If TCPIPStation Is Nothing Then
    MsgBox "TCPIPStation not found!",vbExclamation,"ERROR "
    Exit Sub
End If

strServerAddress = TCPIPStation.ServerAddress
lServerPort = TCPIPStation.ServerPort
strLocalBoundAddress = TCPIPStation.LocalBoundAddress
lLocalBoundPort = TCPIPStation.LocalBoundPort

MsgBox "ServerAddress = " & CStr(strServerAddress) & vbLf & _
"ServerPort = " & CStr(lServerPort) & vbLf & _
"LocalBoundAddress = " & CStr(strLocalBoundAddress) & vbLf & _
"LocalBoundPort = " & CStr(lLocalBoundPort),vbInformation,"ERROR "

Set drv = Nothing
Set station = Nothing
Set RASStation = Nothing
Set TCPIPStation = Nothing
End Sub

```

## 7.11. ServerPort, TCPIPStationInterface Property

- Syntax** Long = station.ServerPort
- Description** This property sets or returns the Port number of the Server to which the station will be connected.
- Remarks** A modification of this value will come acquired only when the socket will open. For example on the project startup before that the communication tasks are executed at least once.

Parameter	Description
None	None

**Result** Long

**Example:**  
See the "ServerAddress" property example.

## 7.12. SwitchServerTimeout, TCPIPStationInterface Property

**Syntax** Long = station.SwitchServerTimeout

**Description** This property sets or returns the time, in milliseconds, which passes between the communication with a server error verification and the attempt to connect to another server. Shows the value of the "Switch Server Timeout" property.

**Remarks**

Parameter	Description
None	None

**Result** Long

**Example:**  
See the "BackupServerAddress" property example.

## 7.13. TxTimeout, TCPIPStationInterface Property

**Syntax** Long = station.TxTimeout

**Description** This property sets or returns the Timeout for the data sending. This time is in milliseconds.

**Remarks** A modification of this value will come acquired only when the socket will open. For example on the project startup before that the communication tasks are executed at least once.

Parameter	Description
None	None

**Result** Long

**Example:**  
See the "RxTimeout" property example.





# 8. RASStationInterface

---

## 8.1. DisconnectAfterSecs, RASStationInterface Property

---

**Syntax** Long = station.DisconnectAfterSecs

**Description** This property sets or returns the time in milliseconds after which the modem connection of the RAS Station will hang up. The counter of the time will start from the moment in which all the communication tasks are not in use, and it will be reset if at least one task returns in use before the expiration of the time.

**Remarks** A modification of this value will come acquired only to the following call.

Parameter	Description
None	None

**Result** Long

**Example:**  
See the "PhoneBookEntry" property example.

## 8.2. EnableRASCallOnThisStation, RASStationInterface Property

---

**Syntax** Boolean = station.EnableRASCallOnThisStation

**Description** This property sets or returns the enable to execute the modem connection for the Station.

**Remarks** A modification of this value will come acquired only to the following call.

Parameter	Description
None	None

**Result** Boolean  
True = RAS call enable  
False = RAS call disable

**Example:**  
See the "PhoneBookEntry" property example.

## 8.3. EndConnectionTime, RASStationInterface Property

---

- Syntax**            Date = station.EndConnectionTime
- Description**      This property returns the date and time of end of the last modem connection of the Station.
- Remarks**          This property is read-only.

Parameter	Description
None	None

**Result**            Date

**Example:**  
See the "**StartConnectionTime**" property example.

## 8.4. GetBaseStationInterface, RASStationInterface Function

---

- Syntax**            GetBaseStationInterface
- Description**      This function used in this interface return an object "**TCPIPStationInterface**".
- Remarks**

Parameter	Description
None	None

**Result**            Object  
If Function has been executed successfully it will retrieve an object of type TCPIPStationInterface if , otherwise Nothing is returned.

**Example:**

```

Sub Main
    Dim drv As DriverInterface
    Dim station As StationInterface
    Dim RASStation As RASStationInterface
    Dim SubRASStation As RASStationInterface
    Dim TCPIPStation As TCPIPStationInterface

    Set drv = GetDriverInterface("Modbus Serial")
    Set station = drv.GetDriverStation("Station1")
    If station Is Nothing Then
        MsgBox "Station not found!", vbExclamation, "ERROR "
        Exit Sub
    End If

    Set RASStation = station.GetSubStationInterface
    If RASStation Is Nothing Then

```

```

        MsgBox "RASStation not found!",vbExclamation,"ERROR "
        Exit Sub
    End If

    Set TCPIPStation = RASStation.GetBaseStationInterface
    If TCPIPStation Is Nothing Then
        MsgBox "TCPIPStation not found!",vbExclamation,"ERROR "
        Exit Sub
    End If

    Set SubRASStation = RASStation.GetSubStationInterface
    If SubRASStation Is Nothing Then
        MsgBox "SubRASStation not found!",vbExclamation,"ERROR "
        Exit Sub
    End If

    Set drv = Nothing
    Set station = Nothing
    Set RASStation = Nothing
    Set SubRASStation = Nothing
    Set TCPIPStation = Nothing
End Sub

```

## 8.5. GetSubStationInterface, RASStationInterface Function

**Syntax**      GetSubStationInterface

**Description**      This function used in this interface return an object "**RASStationInterface**".

**Remarks**

Parameter	Description
None	None

**Result**      Object  
 If Function has been executed successfully it will retrieve an object of type RASStationInterface if , otherwise Nothing is returned.

**Example:**  
 See the "**GetBaseStationInterface**" property example.

## 8.6. GetXMLSettings, RASStationInterface Function

**Syntax**      GetXMLSettings

**Description**      Gets all settings of a station in XML format.

**Remarks**

Parameter	Description
None	None

**Result**        String

**Example:**

```

Sub Main
    Dim drv As DriverInterface
    Dim station As StationInterface
    Dim RASStation As RASStationInterface
    Dim strGetXMLSettings As String
    Set drv = GetDriverInterface("Modbus Serial")
    Set station = drv.GetDriverStation("Station1")
    If station Is Nothing Then
        MsgBox "Station not found!",vbExclamation,"ERROR "
        Exit Sub
    End If
    Set RASStation = station.GetSubStationInterface
    If RASStation Is Nothing Then
        MsgBox "RASStation not found!",vbExclamation,"ERROR "
        Exit Sub
    End If
    strGetXMLSettings = RASStation.GetXMLSettings
    MsgBox "GetXMLSettings = " & CStr(strGetXMLSettings),vbInformation,"ERROR "
    Set drv = Nothing
    Set station = Nothing
    Set RASStation = Nothing
End Sub
    
```

## 8.7. IsConnected, RASStationInterface Function

**Syntax**        IsConnected

**Description**   Gets the connection state of the RAS Station.

**Remarks**

Parameter	Description
None	None

**Result**        Boolean  
                  True = Station Connected  
                  False = Station not Connected

**Example:**

```

Sub Main
    Dim drv As DriverInterface
    Dim station As StationInterface
    Dim RASStation As RASStationInterface
    Dim bIsConnected As Boolean
    Set drv = GetDriverInterface("Modbus Serial")
    Set station = drv.GetDriverStation("Station1")
    If station Is Nothing Then
    
```

```

        MsgBox "Station not found!",vbExclamation,"ERROR "
    Exit Sub
End If
Set TAPIStation = station.GetSubStationInterface
If TAPIStation Is Nothing Then
    MsgBox "TAPIStation not found!",vbExclamation,"ERROR "
    Exit Sub
End If
bIsConnected = TAPIStation.IsConnected
MsgBox "IsConnected = " & CStr(bIsConnected),vbInformation,"ERROR "
Set drv = Nothing
Set station = Nothing
Set TAPIStation = Nothing
End Sub

```

## 8.8. LastConnectionTime, RASStationInterface Property

**Syntax**      Date = station.LastConnectionTime

**Description**    This property returns the connection time of the last modem call of the Station.

**Remarks**      This property is read-only.

Parameter	Description
None	None

**Result**      Date

**Example:**  
See the "StartConnectionTime" property example.

## 8.9. LastRASErrorNumber, RASStationInterface Property

**Syntax**      Long = station.LastRASErrorNumber

**Description**    This property returns the last error code occurs during the modem call of the Station.

**Remarks**      This property is read-only.

Parameter	Description
None	None

**Result**      Long

**Example:**  
See the "StartConnectionTime" property example.

## 8.10. LastRASErrorString, RASStationInterface Property

---

- Syntax** String = station.LastRASErrorString
- Description** This property returns a description text about the state of the modem call of the Station. This string can be an error message or a state connection message.
- Remarks** This property is read-only.

Parameter	Description
None	None

**Result** String

**Example:**  
See the "StartConnectionTime" property example.

## 8.11. NumRetries, RASStationInterface Property

---

- Syntax** Byte = station.NumRetries
- Description** This property sets or returns the number of retries to execute for the modem call of the Station when the connection fails.
- Remarks** A modification of this value will come acquired only to the following call.

Parameter	Description
None	None

**Result** Byte

**Example:**  
See the "PhoneBookEntry" property example.

## 8.12. Password, RASStationInterface Property

---

**Syntax** String = station.Password

**Description** This property sets or returns the password of the User selected for the RAS connection. This field is not used if the property "**PhoneBookEntry**" is not null.

**Remarks** A modification of this value will come acquired only to the following call.

Parameter	Description
None	None

**Result** String

**Example:**  
See the "**PhoneBookEntry**" property example.

## 8.13. PhoneBookEntry, RASStationInterface Property

**Syntax** String = station.PhoneBookEntry

**Description** This property sets or returns the name of the connection previously created and configured in the Operating System.

**Remarks** A modification of this value will come acquired only to the following call.

Parameter	Description
None	None

**Result** String

**Example:**

```

Sub Main
    Dim drv As DriverInterface
    Dim station As StationInterface
    Dim RASStation As RASStationInterface
    Dim strPhoneBook As String
    Dim strPhoneNumber As String
    Dim strUserName As String
    Dim strPassword As String
    Dim nNumRetries As Byte
    Dim lDisconnectAfter As Long
    Dim lRetryAfter As Long
    Dim bEnableRASCall As Boolean
    Dim bPromptForConn As Boolean
    Dim bShowConnDlg As Boolean

    Set drv = GetDriverInterface("Modbus TCPIP")
    Set station = drv.GetDriverStation("Station1")
    If station Is Nothing Then
        MsgBox "Station not found!", vbExclamation, "ERROR "
    Exit Sub
    End If

    Set RASStation = station.GetSubStationInterface
    
```

```

If RASStation Is Nothing Then
    MsgBox "RASStation not found!",vbExclamation,"ERROR "
    Exit Sub
End If

strPhoneBook = RASStation.PhoneBookEntry
strPhoneNumber = RASStation.PhoneNumber
strUserName = RASStation.UserName
strPassword = RASStation.Password
nNumRetries = RASStation.NumRetries
IDisconnectAfter = RASStation.DisconnectAfterSecs
lRetryAfter = RASStation.RetryAfterSecs
bEnableRASCall = RASStation.EnableRASCallOnThisStation
bPromptForConn = RASStation.PromptForConnection
bShowConnDlg = RASStation.ShowConnectionDlg

MsgBox "PhoneBookEntry = " & CStr(strPhoneBook) & vbLf & _
    "PhoneNumber = " & CStr(strPhoneNumber) & vbLf & _
    "UserName = " & CStr(strUserName) & vbLf & _
    "Password = " & CStr(strPassword) & vbLf & _
    "NumRetries = " & CStr(nNumRetries) & vbLf & _
    "DisconnectAfterSecs = " & CStr(IDisconnectAfter) & vbLf & _
    "RetryAfterSecs = " & CStr(lRetryAfter) & vbLf & _
    "EnableRASCallOnThisStation = " & CStr(bEnableRASCall) & vbLf & _
    "PromptForConnection = " & CStr(bPromptForConn) & vbLf & _
    "ShowConnectionDlg = " & CStr(bShowConnDlg),vbInformation,"ERROR "

Set drv = Nothing
Set station = Nothing
Set RASStation = Nothing
End Sub

```

## 8.14. PhoneNumber, RASStationInterface Property

---

- Syntax**        String = station.PhoneNumber
- Description**    This property sets or returns the telephone number to call. This field is not used if the property "**PhoneBookEntry**" is not null.
- Remarks**        A modification of this value will come acquired only to the following call.

Parameter	Description
None	None

**Result**        String

**Example:**  
See the "**PhoneBookEntry**" property example.



## 8.15. PromptForConnection, RASStationInterface Property

**Syntax** Boolean = station.PromptForConnection

**Description** This property allows you to show or hide the connection confirmation dialog window. If this property is true, before start connection, Supervisor will show a confirmation dialog window. Only if the user will click the OK button Supervisor will start the connection.

**Remarks** A modification of this value will come acquired only to the following call.

Parameter	Description
None	None

**Result** Boolean  
 True = show the dialog window before connecting  
 False = don't show the dialog window and start connection

**Example:**  
 See the "PhoneBookEntry" property example.

## 8.16. RetryAfterSecs, RASStationInterface Property

**Syntax** Long = station.RetryAfterSecs

**Description** This property sets or returns the time in milliseconds after which the modem connection will be retried if the previous call has failed.

**Remarks** A modification of this value will come acquired only to the following call.

Parameter	Description
None	None

**Result** Long

**Example:**  
 See the "PhoneBookEntry" property example.

## 8.17. ShowConnectionDlg, RASStationInterface Property

**Syntax** Boolean = station.ShowConnectionDlg

**Description** This property allows you to show or hide the connection status dialog window.

**Remarks** A modification of this value will come acquired only to the following call.

Parameter	Description
None	None

**Result** Boolean  
 True = show dialog window  
 False = hide dialog window

**Example:**  
 See the "PhoneBookEntry" property example.

## 8.18. StartConnectionTime, RASStationInterface Property

---

**Syntax** Date = station.StartConnectionTime

**Description** This property returns the date and time of begin of the last modem connection of the Station.

**Remarks** This property is read-only.

Parameter	Description
None	None

**Result** Date

**Example:**

```

Sub Main
    Dim drv As DriverInterface
    Dim station As StationInterface
    Dim RASStation As RASStationInterface
    Dim dStartConn As Date
    Dim dEndConn As Date
    Dim dTotalConn As Date
    Dim dLastConn As Date
    Dim lLastError As Long
    Dim strLastErrorString As String
    Set drv = GetDriverInterface("Modbus TCPIP")
    Set station = drv.GetDriverStation("Station1")
    If station Is Nothing Then
        MsgBox "Station not found!", vbExclamation, "ERROR "
        Exit Sub
    End If
    Set RASStation = station.GetSubStationInterface
    If RASStation Is Nothing Then
        MsgBox "RASStation not found!", vbExclamation, "ERROR "
        Exit Sub
    End If
    dStartConn = RASStation.StartConnectionTime
    dEndConn = RASStation.EndConnectionTime
    dTotalConn = RASStation.TotalConnectionTime
    dLastConn = RASStation.LastConnectionTime
    
```

```

ILastError = RASStation.LastRASErrorNumber
strLastErrorString = RASStation.LastRASErrorString
MsgBox "StartConnectionTime = " & CStr(dStartConn) & vbLf & _
      "EndConnectionTime = " & CStr(dEndConn) & vbLf & _
      "TotalConnectionTime = " & CStr(dTotalConn) & vbLf & _
      "LastConnectionTime = " & CStr(dLastConn) & vbLf & _
      "LastTAPIError = " & CStr(ILastError) & vbLf & _
      "LastTAPIErrorString = " & CStr(strLastErrorString),vbInformation,"ERROR "
Set drv = Nothing
Set station = Nothing
Set RASStation = Nothing
End Sub

```

## 8.19. TotalConnectionTime, RASStationInterface Property

**Syntax** Date = station.TotalConnectionTime

**Description** This property returns the total connection time of the modem call of the Station. The time is the sum of all calls executed from the start of the Project.

**Remarks** This property is read-only.

Parameter	Description
None	None

**Result** Date

**Example:**  
See the "StartConnectionTime" property example.

## 8.20. UserName, RASStationInterface Property

**Syntax** String = station.UserName

**Description** This property sets or returns the name of the User selected for the RAS connection. This User must be authenticated by the Server. This field is not used if the property "PhoneBookEntry" is not null.

**Remarks** A modification of this value will come acquired only to the following call.

Parameter	Description
None	None

**Result** String

**Example:**  
See the "PhoneBookEntry" property example.

## MOVICON 1



Movicon ist eine HMI/SCADA-Software für das Betriebssystem Microsoft Win32.  
Movicon ist von Progea entwickelt und hergestellt. © 1992 – 2008 Alle Rechte vorbehalten  
Kein Teil dieser Dokumentation oder des Programms darf ohne die schriftliche Erlaubnis von Progea in irgendeiner Form reproduziert oder übertragen werden.  
Informationen in diesem Dokument können ohne Ankündigung geändert werden und sind nicht für den Hersteller bindend.



Via S.Anna, 88/E  
41100 Modena - Italy  
Tel. +39 059 451060  
Fax +39 059 451061  
Email: info@progea.com  
Http://www.progea.com



Via XX Settembre, 30  
Tecnocity Alto Milanese  
20025 Legnano (MI) Italy  
Tel. +39 0331 486653  
Fax +39 0331 455179  
Email: willems@progea.com



Progea Deutschland GmbH  
Marie-Curie-Str. 12  
D-78048 VS-Villingen  
Tel: +49 (0) 7721 / 99 25 992  
Fax: +49 (0) 7721 / 99 25 993  
info@progea.de



Progea International Ltd  
via Penate 16  
6850 Mendrisio - Switzerland  
tel +41 (91) 9676610  
fax +41 (91) 9676611  
international@progea.com



Progea USA LLC  
2800 East Enterprise Avenue  
Appleton, WI 54914  
Tel. +1 (888) 305 2999  
Fax. +1 (920) 257 4213  
info@progea.us