

# W&T OPC-Server für Netzwerk-E/A-Geräte

Version 2.10

Copyright © 1999 - 2003 Wiesemann & Theis GmbH. Basiert auf dem OPC Toolkit von FactorySoft.

Die beigefügte Datei "FSServer.dll" unterliegt dem Copyright © 1997 von FactorySoft, Inc. Sie erhalten sie ausschließlich zur Benutzung als Teil des oben genannten Softwareproduktes, auf Grundlage eines Lizenzvertrages zwischen FactorySoft, Inc. und der Wiesemann & Theis GmbH. Sie sind nicht berechtigt, diese Datei für andere Zwecke zu verwenden oder an Dritte weiterzugeben.

## Irrtum und Änderung vorbehalten

Da wir Fehler machen können, darf keine unserer Aussagen ungeprüft verwendet werden. Bitte melden Sie uns alle Ihnen bekannt gewordenen Irrtümer oder Missverständlichkeiten, damit wir diese so schnell wie möglich erkennen und beseitigen können.

Führen Sie Arbeiten an bzw. mit W&T Produkten nur aus, wenn sie hier beschrieben sind und Sie die Anleitung vollständig gelesen und verstanden haben. Eigenmächtiges Handeln kann Gefahren verursachen. Wir haften nicht für die Folgen eigenmächtigen Handelns. Fragen Sie im Zweifel lieber noch einmal bei uns bzw. Ihrem Händler nach!

<b>1</b>	<b>Allgemeines.....</b>	<b>2</b>
1.1	Über OPC-Server .....	2
1.2	Installation.....	2
1.3	Deinstallation .....	2
<b>2</b>	<b>Konfiguration.....</b>	<b>2</b>
2.1	E/A-Server .....	3
2.2	Serielle Com-Server.....	4
2.3	Web-IO Geräte einrichten .....	5
2.4	Programmooptionen.....	5
<b>3</b>	<b>Direktsteuerung über Bedienfelder .....</b>	<b>6</b>
3.1	OPC-Variablen für E/A-Server .....	6
3.2	OPC-Variablen für serielle Com-Server.....	7
<b>4</b>	<b>Zugriff über Visual Basic bzw. VBA .....</b>	<b>8</b>
4.1	Beispiel: Variablen auflisten .....	8
4.2	Beispiel: Werte auslesen.....	9
4.3	Beispiel: Werte zuweisen .....	9
<b>5</b>	<b>Ein E/A-Server, mehrere Client-Rechner .....</b>	<b>11</b>

# 1 Allgemeines

## 1.1 Über OPC-Server

OPC (OLE for Process Control) ist eine Softwareschnittstelle zum Zugriff auf Prozessdaten, auf Basis der OLE-Technologie von Microsoft. Anwendungsprogramme wie z. B. Visualisierungssysteme, die diese Schnittstelle benutzen, heißen OPC-Clients. Auf der anderen Seite der Schnittstelle stehen ihnen OPC-Server gegenüber. Das sind Gerätetreiber, die eine bestimmte Hardware in abstrakter Form als einen Satz von OPC-Variablen darstellen.

Der hier vorliegende OPC-Server implementiert die Spezifikation OPC Data Access 2.0 und steuert Geräte der W&T Produktfamilien Digital-E/A-Server und Serielle Com-Server. Von der Architektur her handelt es sich, im Gegensatz zu der auch üblichen Implementierung mit einem Systemdienst im Hintergrund, um eine monolithische Anwendung, die sowohl den eigentlichen OPC-Server, als auch Bedienelemente für Konfiguration und Diagnose enthält.

## 1.2 Installation

Starten Sie das Setup-Programm von der Installationsdiskette (indem Sie z. B. aus dem Startmenü "Ausführen..." wählen und dann eingeben: "A:\Setup"). Es installiert und registriert den OPC-Server auf ihrem Rechner.

Der OLE-Servername, den OPC-Clients später angeben müssen, um sich mit dem Server zu verbinden, lautet: "Wiesemann-Theis.DigitalEA". Der OPC-Server startet auf solche Anforderungen hin automatisch. Zum Konfigurieren des Servers können Sie ihn aber auch von Hand ausführen. Einen entsprechenden Eintrag "W&T OPC-Server" finden Sie unter "Programme" im Startmenü.

Hinweis: Unter Windows 95 wird die Installation möglicherweise mit Meldungen über fehlende Funktionen in der OLE32.dll fehlschlagen. Eine aktuellere Version dieser DLL, die das Problem behebt, ist in dem Update-Paket "DCOM für Windows 95" enthalten, das Microsoft zum kostenlosen Download im Internet anbietet, unter der Adresse [http://www.microsoft.com/com/dcom/dcom95/dcom1\\_3.asp](http://www.microsoft.com/com/dcom/dcom95/dcom1_3.asp).

## 1.3 Deinstallation

Den OPC-Server entfernen können Sie über die Systemsteuerungskomponente "Software". Er steht dort in der Liste unter "W&T OPC-Server für Digital-EA".

# 2 Konfiguration

Die beiden im folgenden beschriebenen Dialogfelder begegnen Ihnen sowohl, wenn Sie einen neuen Com-Server oder E/A-Server zur Konfiguration hinzufügen, als auch wenn Sie die Angaben zu einem existierenden Server nachträglich ändern wollen.

## 2.1 E/A-Server

**Hostname oder IP-Adresse:** Muss mit der IP-Adresse übereinstimmen, die am Gerät eingestellt wurde (vier Zahlen, durch Punkte getrennt, wie z. B. 172.16.232.77). Falls der Adresse über DNS oder einen ähnlichen Namensdienst auch ein Hostname zugeordnet wurde, können Sie auch diesen Namen statt der IP-Adresse verwenden.

**TCP-Portnummer:** Ist bei Geräten der alten 50000er Serie normalerweise (voreingestellt ab Werk) 49153. Bei den neuen Web-IO's (z. B. Typ 57630) überprüfen Sie bitte die entsprechende Einstellung im Web-Menü des Gerätes.

**System-Passwort:** Betrifft nur Web-IO Geräte. Geben Sie hier das Konfigurator-Passwort an, sofern für das Gerät eines festgelegt wurde. Falls Sie sich über das korrekte Passwort unsicher sind, achten Sie bitte nach Beenden des Dialogs auf die Statusmeldung für das Gerät, in der ggf. auf Probleme hingewiesen wird ("Passwort erforderlich" bzw. "falsches Passwort").

**Fortlaufende Gerätenummer:** Diese Nummer wird beim Hinzufügen eines neuen E/A-Servers oder Com-Servers automatisch zugewiesen und soll garantieren, dass alle Geräte verschiedenen Namen erhalten (Box1, Box2, Com3, Box4 usw.). Änderungen von Hand können sinnvoll sein, um z. B. den E/A-Ports und dem seriellen Port eines E/A-Servers dieselbe Gerätenummer zuzuweisen. Die resultierenden Gerätenamen wären dann z. B. "Box1" und "Com1".

**Aktualisierung mindestens alle ... Sek.:** Manche Ereignisse (insbesondere Änderungen an seinen binären Eingängen) meldet der E/A-Server von sich aus, für andere Daten ist man auf regelmäßiges aktives Abfragen angewiesen (betrifft u. a. die Ausgangsströme bei Geräten mit Messfunktion und die Ereigniszähler der Web-IO's). Den Abstand zwischen zwei solchen Abfragen legt dieser Wert fest, wobei die Eingabe auf ein Vielfaches von 0.1 Sekunden gerundet wird.

Diese regelmäßigen Abfragen dienen außerdem dazu, eine Störung der TCP-Verbindung zum Endgerät möglichst bald zu erkennen. Eine Störung wird dann vermutet (und die Verbindung wird zurückgesetzt), wenn zwischen zwei Anfragen keine Antwort vom Gerät eingetroffen ist. Vorsicht: Ein sehr kleiner Wert an dieser Stelle kann darum einen Verbindungsaufbau unmöglich machen.

**Nur pollen (d. h. keine Benachrichtigungen abonnieren):** Auf die automatischen Benachrichtigungen zu verzichten, hat einen einzigen Vorteil, nämlich dass der Umfang des Datenverkehrs auf dem Netzwerk nicht von äußeren Ereignissen abhängt. Der E/A-Server erzeugt aber selbst unter ungünstigsten Umständen (prellende oder schwingende Eingänge) weniger als hundert Benachrichtigungen pro Sekunde, so dass erst viele E/A-Server innerhalb ein- und desselben Netzwerks zu einem Problem werden könnten, das diese drastische Maßnahme rechtfertigt.

**Gerätetyp:** "Standard" sind die E/A-Server-Modelle 50210/50310. "Mit Messfunktion" meint die Modelle 50211/50311, bei denen sich Ausgangsströme und Gehäusetemperatur abfragen lassen. Die Funktion "Identifizieren" kann helfen, hier den richtigen Typ zu wählen, wird evtl. aber auch fehlschlagen, z. B. wenn das betreffende Gerät gerade durch einen anderen TCP-Client belegt ist. Bitte beachten Sie, dass der dritte Gerätetyp, "Web-IO", vor seiner Verwendung mit dem OPC-Server erst geeignet konfiguriert werden muss.

## 2.2 Serielle Com-Server

Die Optionen "**Hostname oder IP-Adresse**" und "**Fortlaufende Gerätenummer**" haben die gleiche Bedeutung wie bei E/A-Servern, s. o.

**TCP-Portnummer:** Ist normalerweise 8000, bzw. für die seriellen Ports B/C/D eines Mehrfach-Com-Servers: 8100, 8200, 8300. Es werden dabei in Wirklichkeit zwei TCP-Verbindungen geöffnet: Eine Datenverbindung unter der angegebenen Portnummer, sowie eine Steuerverbindung unter einer um 1094 höheren Portnummer, also 9094, 9194, 9294 oder 9394.

**Statusabfrage alle ... Sek.:** Das Eintreffen von RS232-Empfangsdaten melden Com-Server von sich aus, für alle anderen Ereignisse (einschließlich Änderungen an den Handshakeleitungen) ist man auf regelmäßige Abfragen angewiesen.

**Handshakeleitungen als E/A verwenden:** In diesem Modus können die RS232-Signale DSR/CTS/DTR/RTS über OPC-Variablen (E.0/E.1/A.0/A.1) angesprochen werden und stehen für echtes Hardware-Handshake nicht mehr zur Verfügung. Das Ganze war ursprünglich einmal für Com-Server mit dem speziellen Schnittstellenmodul 18804 gedacht und hat ansonsten keine praktische Anwendung.

**RS232-Parameter:** Ein Initialisierungsstring wie z. B. "19200,n,8,1,x", der die Schnittstellenparameter beim Start des OPC-Servers festlegt. Bedeutung der Parameter innerhalb des Strings: Baudrate, Paritätsbit (n = keins, e = gerade, o = ungerade), Anzahl Datenbits (5 - 8), Anzahl Stopbits (1 oder 2), Handshakeverfahren (x = Software, XOn/XOff, h = Hardware, RTS/CTS).

Wenn kein Initialisierungsstring angegeben ist, verwendet der OPC-Server die seriellen Einstellungen, wie er sie im Com-Server bei Verbindungsaufbau vorfindet.

**Textempfang komplett bei...:** Um den empfangenen RS232-Datenstrom für OPC als String-Variable darstellen zu können, muss er zunächst in Pakete zerlegt werden. Paketgrenzen werden dabei entweder anhand von Pausen oder am Auftreten spezieller Zeichenfolgen erkannt. Wenn eine Zeichenfolge angegeben wird, sollte gleichzeitig die Erkennung von Pausen abgeschaltet werden (durch Eingabe eines Wertes von 0 Sekunden). Eine Kombination der beiden Kriterien ist zwar möglich, aber nicht unbedingt empfehlenswert. Zeichenfolgen werden als Dezimalzahlen angegeben, niemals direkt als String. Beispielsweise wäre also für drei Pluszeichen als Paketgrenze ("+++") die korrekte Eingabe "43,43,43".

**Beim Empfang ausfiltern:** Das Ausfiltern von Sonderzeichen findet erst statt, nachdem die empfangenen Daten bereits in Pakete zerlegt sind. Ausfiltern von Zeichen, die gleichzeitig Trennzeichen darstellen, ist also zulässig. Null-Bytes werden in jedem Fall ausgefiltert, auch wenn hier nicht ausdrücklich "0" mit angegeben wird, da sie nicht innerhalb von Strings auftauchen dürfen.

**Beim Senden anhängen:** Sinnvoll kann insbesondere das Anhängen von CR (13), LF (10) oder beiden (13,10) als Zeilenendezeichen sein.

## 2.3 Web-IO Geräte einrichten

Bevor der OPC-Server auf ein Web-IO-Gerät zugreifen kann, muss zunächst einer der beiden Binärkanäle des Geräts für diesen Zweck reserviert werden: Im Web-Menü unter "Config >> Device >> Basic Settings >> Binary 1 (bzw. 2) >> TCP Server" sollte "Application Mode: OPC Device" ausgewählt sein. An gleicher Stelle überprüfen Sie bitte auch die Einstellung "Local Port", die mit der im OPC-Server eingetragenen TCP-Portnummer übereinstimmen muss. (Empfohlen ist 49153 für Kanal 1 und 49154 für Kanal 2.)

Eine weitere Besonderheit beim Web-IO besteht darin, dass jedes Gerät diverse Verbindungen gleichzeitig zulässt, u. a. HTTP und die gerade erwähnten beiden Binärkanäle. Um Konflikte bei der Ansteuerung der Ausgänge zu vermeiden, lassen diese sich zwar auf beliebigem Weg auslesen, aber nur auf jeweils einem Weg schreiben. Welcher das ist, wird für jeden Ausgang individuell per Konfiguration festgelegt, unter "Config >> Device >> Output Mode". Hier sollten Sie also zunächst auch noch einige oder alle Ausgänge dem für den OPC-Server verwendeten Binärkanal zuordnen.

## 2.4 Programmoptionen

Über den "Optionen"-Dialog werden einige Details am Verhalten des OPC-Servers festgelegt.

**Fenster verbergen:** Spätestens dann empfehlenswert, wenn ansonsten alles konfiguriert und getestet ist. Um ein verborgenes Fenster bei Bedarf doch wieder anzuzeigen, rufen Sie über das Startmenü den OPC-Server erneut auf. Beachten Sie, dass hierdurch nicht etwa eine zweite Instanz des OPC-Servers gestartet, sondern wirklich nur der im Hintergrund laufende OPC-Server wieder sichtbar gemacht wird. Nach Abschluss aller Tests und/oder Änderungen sollten Sie darum das Fenster nicht schließen, sondern nur minimieren, woraufhin es wieder verschwindet.

**E/A-Geräte freigeben:** Freigeben bedeutet in diesem Zusammenhang, die TCP-Verbindungen zu den Geräten zu trennen, so dass andere Clients wieder darauf zugreifen können. Die Wirkung ist mit der vergleichbar, die man auch durch manuelles Deaktivieren einzelner Geräte erzielen kann. Die OPC-Variablen freigegebener Geräte existieren weiter, liefern aber keine gültigen Werte mehr.

**Watchdog** (VT\_R4, R/W) ist eine globale, d. h. keinem bestimmten E/A-Gerät zugeordnete OPC-Variable. Sie enthält einen Sekundenwert, der kontinuierlich heruntergezählt wird, sofern die entsprechende Option hier aktiviert ist. Sobald der Wert 0 erreicht, werden die E/A-Geräte freigegeben. Der Status des OPC-Servers ändert sich dann in OPC\_STATUS\_SUSPENDED, und das Auslesen einzelner Variablen liefert nur noch OPC\_QUALITY\_NOT\_CONNECTED. Der Client kann das verhindern, indem er in regelmäßigen Abständen immer wieder einen von Null verschiedenen Wert zuweist, z. B. alle 10 Sekunden den Wert 15.

Der andere Freigabemechanismus, der nur darauf achtet, ob noch OPC-Clients verbunden sind, ist offensichtlich eleganter, da er ohne zusätzlichen Aufwand auf der Client-Seite funktioniert. Er ist allerdings darauf angewiesen, dass sich alle Clients korrekt an- und wieder abmelden, und das ist nicht unbedingt garantiert. Denkbare Probleme sind fehlerhaft programmierte oder überraschend abstürzende Clients sowie unterbrochene Netzwerkverbindungen zu Clients auf einem anderen Rechner. Das Verfahren mit der Watchdog-Variable ist gegen solche Probleme unempfindlich.

**Ganz beenden, wenn bereits verborgen:** Voraussetzung für das automatische Beenden ist wie gesagt erstens, dass das Programmfenster nicht sichtbar ist. Außerdem beendet sich das Programm niemals selber in einer Situation, wo Beenden von Hand zu einer Rückfrage führen würde: Also weder wenn noch OPC-Clients verbunden sind, noch solange ungespeicherte Änderungen an der Gerätekonfiguration vorliegen.

### 3 Direktsteuerung über Bedienfelder

Für jeden Server in der Liste können Sie ein Bedienfeld öffnen, das im Wesentlichen die selben Zugriffsmöglichkeiten bietet wie die OPC-Schnittstelle. Insbesondere sind die einzelnen Steuerelemente mit den Namen der entsprechenden OPC-Variablen beschriftet.

Nützlich ist das vor allem, um sich mit dem Verhalten der Endgeräte vertraut zu machen, und um z. B. bei seriellen Schnittstellen die Wirkung der diversen Filteroptionen zu sehen. Für eine echte Fernsteuerung der Geräte ist diese Funktion dagegen vermutlich nicht sinnvoll, da sich nicht mehr als ein Bedienfeld gleichzeitig öffnen lässt.

#### Verwendete Abkürzungen

##### OLE-Datentypen:

VT\_BOOL: Binärwert  
VT\_I2, VT\_I4: Ganzzahl  
VT\_R4: Fließkommazahl  
VT\_BSTR: Zeichenkette

##### Zugriffsrechte:

R/W: lesen und schreiben  
R: nur lesen  
W: nur schreiben

#### 3.1 OPC-Variablen für E/A-Server

Die vollständigen Variablennamen beginnen mit dem Gerätenamen des E/A-Servers, dem sie zugeordnet sind. Am konkreten Beispiel eines Servers namens "Box1" sieht das Ganze so aus:

**Box1.E.0 — Box1.E.11** (VT\_BOOL, R): Zustand der Eingänge.

**Box1.A.0 — Box1.A.11** (VT\_BOOL, R/W): Zustand der Ausgänge.

Darüber hinaus gibt es diese Variablen nur für E/A-Server mit Messfunktion:

**Box1.I.0 — Box1.I.11** (VT\_R4, R): Ausgangsströme, in Ampere.

**Box1.T** (VT\_R4, R): Gerätetemperatur, in Grad Celsius.

**Box1.U** (VT\_R4, R): Versorgungsspannung, in Volt.

**Box1.F** (VT\_BOOL, R/W): Fehler, Ausgänge wegen Überlast abgeschaltet. Zurücksetzen der Variable schaltet wieder ein.

Und diese für Web-IO:

**Box1.N.0 — Box1.N.11** (VT\_I4, R/W): Ereigniszähler für die Eingänge E.0 — E.11. Es werden sowohl steigende als auch fallende Flanke gezählt, d. h. ein einfacher Puls am Eingang erhöht den Zählerstand um 2. Durch Hineinschreiben des Wertes 0 lassen die Zähler sich zurücksetzen.

Je nach Modell besitzen manche E/A-Server außerdem noch eine serielle Schnittstelle. Falls Sie die ebenfalls über OPC ansprechen möchten und die dazu notwendigen Variablen hier vermissen: Die gibt es tatsächlich nicht. Aus Sicht des OPC-Servers stellt eine solche serielle Schnittstelle einen eigenständigen Com-Server dar. Um dennoch ans Ziel zu gelangen, legen Sie für das betreffende Gerät bitte zwei Server mit derselben IP-Adresse an, von denen der eine ein E/A-Server ist, der andere vom Typ "seriell".

### **3.2 OPC-Variablen für serielle Com-Server**

Die Gerätenamen für serielle Com-Server lauten Com1, Com2, usw., und die Variablen im einzelnen sind:

**Com1.TxD** (VT\_BSTR, W): RS232-Sendedaten, zugewiesene Werte werden über die serielle Schnittstelle ausgegeben.

**Com1.RxD** (VT\_BSTR, R): RS232-Empfangsdaten (das zuletzt von der seriellen Schnittstelle eingetroffene Textpaket).

**Com1.N** (VT\_I4, R/W): Paketzähler, wird bei jedem Empfang eines Textpakets um 1 erhöht.

Sofern die Handshakeleitungen für E/A konfiguriert sind, gibt es außerdem:

**Com1.E.0** (VT\_BOOL, R): binärer Eingang (DSR, bzw. Pin 4 am E/A-Modul 18804).

**Com1.E.1** (VT\_BOOL, R): binärer Eingang (CTS, bzw. E/A-Pin 7).

**Com1.A.0** (VT\_BOOL, R/W): binärer Ausgang (DTR, bzw. E/A-Pin 6).

**Com1.A.1** (VT\_BOOL, R/W): binärer Ausgang (RTS, bzw. E/A-Pin 8).

Die folgenden Variablen geben Zugriff auf die Schnittstellenparameter, wie sie auch über den RS232-Initialisierungsstring bei der Gerätekonfiguration eingestellt werden können:

**Com1.Line.Baud** (VT\_I4, R/W): Baudrate, in Bit pro Sekunde.

**Com1.Line.Bits** (VT\_I2, R/W): Anzahl Datenbits pro Zeichen (5, 6, 7 oder 8).

**Com1.Line.Stop** (VT\_I2, R/W): Anzahl Stopbits (1 oder 2).

**Com1.Line.Parity** (VT\_I2, R/W): Paritätsbit: 0 = keins, 1 = ungerade, 2 = gerade Parität

**Com1.Line.Flow** (VT\_BOOL, R/W): soll ein Handshake-Protokoll zur seriellen Flusskontrolle verwendet werden?

**Com1.Line.Hard** (VT\_BOOL, R/W): Hardware- oder Software-Handshake? (Nicht verfügbar bei seriellen Ports, deren Handshakeleitungen für E/A konfiguriert sind.)

## 4 Zugriff über Visual Basic bzw. VBA

Dieser OPC-Server bietet, abgesehen von dem grundlegenden Custom-Interface, nur ein Automatisierungs-Interface nach dem neuen OPC-Standard 2.0 an. Falls Sie mit dem alten OPC Automatisierungs-Standard 1.0 vertraut sind: Die Unterschiede zwischen den beiden Versionen sind erheblich. Außerdem ist zur Benutzung des neuen Interface mindestens Visual Basic Version 5.0 bzw. Office 97 erforderlich. Die komplette Schnittstellen-Spezifikation finden Sie auf der Homepage der OPC Foundation, <http://www.opcfoundation.org>. Für einen ersten Eindruck sind aber auch vielleicht die folgenden Beispiele (Excel-Makros) hilfreich.

Damit Visual Basic das OPC-Interface benutzen kann, muss in der Liste der Verweise der Eintrag "OPC Automation 2.0" aktiviert sein. (Zu dieser Liste gelangen Sie bei Visual Basic 6.0 über den Menüpunkt "Projekt/Verweise...", bei Excel 97 über den Menüpunkt "Extras/Verweise..." im Visual Basic Editor.)

### 4.1 Beispiel: Variablen auflisten

Dieses Beispiel benutzt den Namensraum-Browser, um alle verfügbaren Variablen des OPC-Servers zu ermitteln, und trägt sie in Spalte 1 der aktuellen Excel-Tabelle ein.

Option Base 1

```
Sub OpcGetNames()  
    ' Trägt die verfügbaren Variablennamen in Spalte 1 ein  
    Dim TheOpcServer As OPCServer  
    Dim MyBrowser As OPCBrowser  
    Set TheOpcServer = New OPCServer  
    TheOpcServer.Connect ("Wiesemann-Theis.DigitalEA")  
    Set MyBrowser = TheOpcServer.CreateBrowser  
  
    Dim i As Integer  
    ' Inhalt von Spalte 1 zunächst löschen.  
    Columns("A").ClearContents  
    MyBrowser.ShowLeafs (True)  
    For i = 1 To MyBrowser.Count  
        Cells(i, 1) = MyBrowser.Item(i)  
    Next i  
  
    Set MyBrowser = Nothing  
    TheOpcServer.Disconnect  
    Set TheOpcServer = Nothing  
End Sub
```

## 4.2 Beispiel: Werte auslesen

Dieses Beispiel liest für alle Variablennamen, die es in Spalte 1 der aktuellen Excel-Tabelle vorfindet, die Eigenschaften "Wert", "Maßeinheit" und "Beschreibung", und trägt sie in die benachbarten Spalten ein.

Eine besondere Bedeutung hat dabei die Eigenschaft "Signalqualität", die im wesentlichen aussagt, ob der OPC-Server einen gültigen Wert für eine bestimmte Variable hat. Ein möglicher und leider gar nicht so unwahrscheinlicher Grund für Probleme in dieser Hinsicht kann sein, dass der OPC-Server gerade erst gestartet ist (automatisch aufgrund der Anforderung durch das Makro), und noch keine TCP-Verbindung zu seinen Endgeräten aufbauen konnte.

Option Base 1

```
Sub OpcUpdate()  
    ' Fragt zu allen Variablennamen aus Spalte 1 Beschreibung  
    ' und aktuellen Inhalt ab  
    Dim TheOpcServer As OPCServer  
    Set TheOpcServer = New OPCServer  
    TheOpcServer.Connect ("Wiesemann-Theis.DigitaleA")  
  
    Dim PropertyIDs(5) As Long  
    Dim Data() As Variant  
    Dim Errors() As Long  
    Dim i, j As Integer  
    PropertyIDs(1) = 3      ' OPC_PROP_QUALITY  
    PropertyIDs(2) = 2      ' OPC_PROP_VALUE  
    PropertyIDs(3) = 100    ' OPC_PROP_UNIT  
    PropertyIDs(4) = 101    ' OPC_PROP_DESC  
    PropertyIDs(5) = 4      ' OPC_PROP_TIME  
    Columns("B").ClearContents  
    Columns("E:F").ClearContents  
    i = 1  
    While Cells(i, 1) <> ""  
        TheOpcServer.GetItemProperties Cells(i, 1), 5, PropertyIDs, Data,  
Errors  
        For j = 2 To 5  
            Cells(i, j) = Data(j)  
        Next j  
        If Data(1) = 20 Then      ' OPC_QUALITY_LAST_KNOWN  
            Cells(i, 6) = "VERALTET"  
        ElseIf Data(1) <> 192 Then ' OPC_QUALITY_GOOD  
            Cells(i, 6) = "FEHLER"  
            Range(Cells(i, 2), Cells(i, 3)).ClearContents  
        End If  
        n = n + 1  
    Wend  
  
    TheOpcServer.Disconnect  
    Set TheOpcServer = Nothing  
End Sub
```

## 4.3 Beispiel: Werte zuweisen

Hier wird aus Spalte 1 der aktuellen Excel-Tabellenzeile ein Variablenname gelesen, aus Spalte 2 ein Wert, und der Wert an die Variable zugewiesen.

Anders als das Auslesen von Werten ist dieser Vorgang nicht über die "Item Properties" des OPC-Servers realisierbar, sondern nur direkt über das entsprechende OPCItem-Objekt. Übrigens sollte aus Effizienzgründen normalerweise auch das Lesen von Variablen auf diese Weise erfolgen, und nicht mit der in Beispiel 2 gezeigten Methode. Und anders als in diesem Makro würde man die OPCItem-Objekte normalerweise auch nicht für jeden Zugriff erst anlegen und dann sofort wieder freigeben.

Option Base 1

```
Sub OpcWrite()  
    ' Für die aktuelle Zeile dem Variablennamen (erste Spalte)  
    ' einen Wert (aus der zweiten Spalte) zuweisen.  
    Dim TheOpcServer As OPCServer  
    Dim MyGroup As OPCGroup  
    Dim MyItem As OPCItem  
    Set TheOpcServer = New OPCServer  
    TheOpcServer.Connect ("Wiesemann-Theis.DigitalEA")  
    Set MyGroup = TheOpcServer.OPCGroups.Add("group")  
  
    Set MyItem = MyGroup.OPCItems.AddItem(Cells(ActiveCell.Row, 1), 1234)  
    MyItem.Write (Cells(ActiveCell.Row, 2))  
  
    TheOpcServer.OPCGroups.Remove (MyGroup.ServerHandle)  
    TheOpcServer.Disconnect  
    Set TheOpcServer = Nothing  
End Sub
```

## 5 Ein E/A-Server, mehrere Client-Rechner

Die TCP-Verbindung, die der OPC-Server zu einem ihm zugeordneten E/A-Server oder Com-Server aufbaut, gibt ihm exklusiven Zugriff auf das Gerät. Eine Konfiguration, in der OPC-Server von mehreren Rechnern aus auf ein- und denselben E/A-Server zugreifen, ist darum nicht sinnvoll und würde nicht zuverlässig funktionieren.

Einen vergleichbaren Effekt erzielt man stattdessen mit einem einzelnen OPC-Server, auf den Clients von mehreren Rechnern aus zugreifen, über DCOM (Distributed COM). Beachten Sie nur, dass Sie für die Benutzung von DCOM erst noch Zugriffsrechte erteilen müssen. Das geschieht mit dem Programm "DcomCnfg.exe", das zu Windows NT 4.0 standardmäßig dazugehört.

Um DCOM auch unter Windows 95 benutzen zu können, müssen Sie erst die Update-Pakete DCOM95.EXE und DCM95CFG.EXE installieren. Ob das bereits geschehen ist, können Sie feststellen, indem Sie von der Eingabeaufforderung aus versuchen, DcomCnfg.exe auszuführen. Wenn das gelingt, können Sie in dem Programm auch gleich noch die Registerkarte "Default Security" aufschlagen und dort das Häkchen "Enable remote connection" setzen. Und beachten Sie, dass es unter Windows 95 nicht möglich ist, einen OLE-Server auf einen DCOM-Aufruf von außerhalb hin starten zu lassen. Der OPC-Server müsste auf einem Win95-Rechner also von Hand gestartet werden, bevor sich OPC-Clients von anderen Rechnern aus mit ihm verbinden können.

Ausführliche Informationen zum Thema DCOM finden Sie auch in der Microsoft Knowledge Base, u. a. in den Artikeln:

Q176799, INFO: Using DCOM Config (DCOMCNFG.EXE) on Windows NT  
Q165101, HOWTO: Use Win95 as a DCOM Server  
Q182248, HOWTO: Use DCOM Config (DCOMCNFG.EXE) with Windows 95  
Q158508, FAQ: COM Security Frequently Asked Questions  
Q174024, FAQ: DCOM95 Frequently Asked Questions

Eine von vielen Methoden zum Auffinden dieser Artikel ist die Suchmaschine auf Microsofts Support-Seite, <http://support.microsoft.com>. Wählen Sie dort als Kategorie "Englische Knowledge Base" aus und geben Sie als Suchbegriff eine Artikelnummer wie "Q165101" ein.