

# IP-S7-Link .Net Advanced COM

© by TIS

Der IP-S7-Link .Net Advanced COM ist eine Bibliothek, die das Softwaretreiber-Framework IP-S7-Link .Net Advanced zur Kommunikation und zum Austausch von Daten mit speicherprogrammierbaren Steuerungen (SPS) im industriellen Automatisierungsraum und in anderen Branchen nutzt. Die Bibliothek stellt eine Anwendungsprogrammierschnittstelle zur Interaktion mit SPS-Geräten über den Internetprotokollstack (IP) für Simatic S7-Controller zur Verfügung. Diese Schnittstelle wurde mit dem Component Object Model (COM) von Microsoft implementiert. Dadurch kann die Bibliothek in Visual Basic 6 (VB6), Visual Basic Script (VBScript) und Visual Basic für Applikationen (VBA) verwendet werden. Dort enthält VBA alle Microsoft Office-Produkte, die VBA-Makroprogramme wie Microsoft Excel unterstützen.

Die Dokumentation für die Bibliothek enthält eine Referenz für die Klassenbibliothek, einen konzeptionellen Überblick, Schritt-für-Schritt-Verfahren und Informationen mit Beispielen. Um die Informationen zu finden, die Sie interessieren, finden Sie nachfolgend die wichtigsten Themenbereiche.

## Kapitelübersicht

### [IP-S7-Link .Net Advanced COM Essentials](#)

Bietet einen schnellen Einblick in die technischen Begriffe der Kommunikation mit SPS-Geräten.

### [Getting started with the IP-S7-Link .Net Advanced COM](#)

Bietet einen umfassenden Überblick über die erweiterte COM-Bibliothek IP-S7-Link .Net und Links zu weiteren Ressourcen.

### [Getting started with the IP-S7-Link .Net Advanced COM in Microsoft Excel](#)

Bietet Schritt für Schritt Anweisungen, wie die IP-S7-Link .NET Advanced-COM-Bibliothek in Microsoft Excel mit Visual Basic für Applikationen (VBA) verwendet werden, um VBA-Makros zu entwickeln.

### [IP-S7-Link .Net Advanced COM Code Snippets](#)

Stellt eine Sammlung von Instant-Szenario-basierten Code-Snippets zur direkten Sprung in die Lösung für die häufigsten Anforderungen während der Entwicklung.

### [IP-S7-Link .Net Advanced COM Frequently Asked Questions \(FAQs\)](#)

Bietet alle Antworten auf alle häufig gestellten Fragen, die typischerweise nach demselben Verfahren gelöst werden können.

# IP-S7-Link .Net Advanced COM Grundlagen

© by TIS

## Was ist das IP-S7-Link .Net Advanced COM?

Das IP-S7-Link .NET Advanced COM ist eine Bibliothek, die das Software-Treiber-Framework IP-S7-Link .Net Advanced verwendet, das eine vollständige Implementierung des Internetprotokolls (IP) zur Kommunikation und zum Austausch von Daten mit programmierbarer Logik bereitstellt Steuerungen (SPS) ermöglicht.

Das Framework realisiert einen fließenden Übergang von PC-spezifischen Daten zu SPS-spezifischen Datenformaten und umgekehrt.

Der implementierte Kommunikationsstapel realisiert mittels komplexen Optimierungsalgorithmen eine Reduzierung/Optimierung der Dateninteraktion mit der SPS. Die Bibliothek stellt nahezu eine eins zu eins Schnittstelle von COM zum Framework bereit. Dies geschieht mittels einer vollständig objektorientierten API, die auch auf einen fließenden Übergang von COM-spezifischen Daten zu SPS-spezifischen Datenformaten und umgekehrt achtet. Dadurch wird die gesamte Logik in fließende Klasseninterfaces gepackt, die auf gemeinsame Automatisierungsszenarien in COM-basierten Applikationen abzielen und es sind keine Kenntnisse über den Stack hinter dem Treiber erforderlich.

## Verwandte Themen

Titel	Beschreibung
<a href="#">SIMATIC controllers from Siemens</a>	Einen kurzen Überblick über die Automatisierungssysteme der Siemens AG.
<a href="#">Technical documentation SIMATIC Controller</a>	Bietet eine Reihe von Dokumenten zu den verschiedenen Automatisierungslösungen der Siemens AG.
<a href="#">SIMATIC Programming with STEP 7</a>	Die gesamte Dokumentation mit STEP 7 zu entwickeln.

# Erste Schritte mit dem IP-S7-Link .Net Advanced COM

© by TIS

Das IP-S7-Link .Net Advanced COM ist eine Bibliothek, die das Softwaretreiber-Framework IP-S7-Link .Net Advanced verwendet, das den gesamten IP-Stack für Simatic S7-Controller implementiert. Während die Bibliothek einfache Lese- / Schreibmethoden bereitstellt, zielt sie auch auf verschiedene erweiterte Szenarien ab. Die Verwendung dieser Bibliothek bietet nicht nur die Möglichkeit, programmierbare Logikcontroller-Adressen (SPS-Adressen) an Ihre COM-Anwendung zu binden. Es bietet auch Mechanismen, um Variablen oder ganze Strukturen mit einer oder mehreren SPS-Adressen Ihrer SPS zu verknüpfen und diese gleichzeitig zu lesen und zu schreiben.

## Operand

Name	Abkürzung (Siemens, DE)	Abkürzung(IEC)
Eingang	E	I
Ausgang	A	Q
Merker	M	M
Peripherie	P	P
Zähler	Z	C
Datenbaustein	DB	DB
Timer	T	16

## Datentypen

Name	Operand	Bits	Reichweite	Beschreibung	Array
BOOL	X	1	0 bis 1	Bit-Wert wahr (1) oder falsch (0)	x
BYTE	B	8	0 to 255	Unsigned 8-Bit Ganzzahl	x
WORD	W	16	0 to 65.535	Unsigned 16-Bit Ganzzahl (Word)	x
DWORD	D	32	0 to $2^{32}-1$	Unsigned 32-bit Ganzzahl (Double Word)	x
CHAR	B	8	A+00 to A+ff	Unsigned 8-Bit-Zeichen im ASCII-CODE	x
INT	W	16	-32.768 to 32.767	Signed 16-Bit-Ganzzahl	x
DINT	D	32	$-2^{31}$ to $2^{31}-1$	Signed 32-Bit-Ganzzahl	x
REAL	D	32	+ -1.5e-45 to + -3.4e38	IEEE754 32-bit single precision floating point Zahl	x
S5TIME	W	16	00.00:00:00.100 to 00.02:46:30.000	binär codierte Dezimalzahl (BCD), die eine Zeitspanne repräsentiert	
TIME	D	32	00.00:00:00.000 to 24.20:31:23.647	Signed 16-Bit-Ganzzahl, die eine Zeitspanne in Millisekunden darstellt	
TIME_OF_DAY	D	32	00.00:00:00.000 to 00.23:59:59.999	unsigned 16-Bit-Ganzzahl, die eine Zeitspanne in Millisekunden darstellt	
DATE	W	16	01.01.1990 to 31.12.2168	unsigned 16-Bit-Ganzzahl, die ein Datum in Tagen repräsentiert	

Name	Operant	Bits	Reichweite	Beschreibung	Array
DATE_AND_TIME	D	64	00:00:00.000 01.01.1990 to 23:59:59.999 31.12.2089	binär codierte Dezimalzahl (BCD), die ein Datum und eine Uhrzeit repräsentiert	
S7String	B	beliebig	A+00 to A+ff	ASCII-Code, max. 254 Bytes	

Die Variablen setzen sich aus Operand und Datentyp zusammen. Beispiele:

Beispiele	Datentyp	Bsp. Siemens	Bsp. IEC
Eingang Byte 1, Bit 0	BOOL	E 1.0	I 1.0
Ausgang Byte 1, Bit 7	BOOL	A 1.7	Q 1.7
Merker Byte 10, Bit 1	BOOL	M 10.1	M 10.1
Datenbaustein 1, Byte 1, Bit 0	BOOL	DB1.DBX 1.0	DB1.DBX 1.0
Eingang Byte 1	BYTE	EB 1	IB 1
Ausgang Byte 10	BYTE	AB 10	QB 10
Merker Byte 100	BYTE	MB 100	MB 100
Peripherie Eingang Byte 0	BYTE	PEB 0	PIB 0
Peripherie Ausgang Byte 1	BYTE	PAB 1	PQB 1
Datenbaustein 1, Byte 1	BYTE	DB1.DBB 1	DB1.DBB 1

Datenbaustein 1, Datenbaustein Typ bool, Adresse 1.0 → DB1.DBX 1.0

Datenbaustein 1, Datenbaustein Typ Byte, Adresse 1 → DB1.DBB 1

Peripherie Eingang, Typ DWORD, Adresse 0 → PED 0

Merkhilfe:

DB#.DBB # = Datenbaustein#.Datenbaustein Byte #

DB#.DBW # = Datenbaustein#.Datenbaustein Wort #

DB#.DBD # = Datenbaustein#.Datenbaustein Doppelwort #

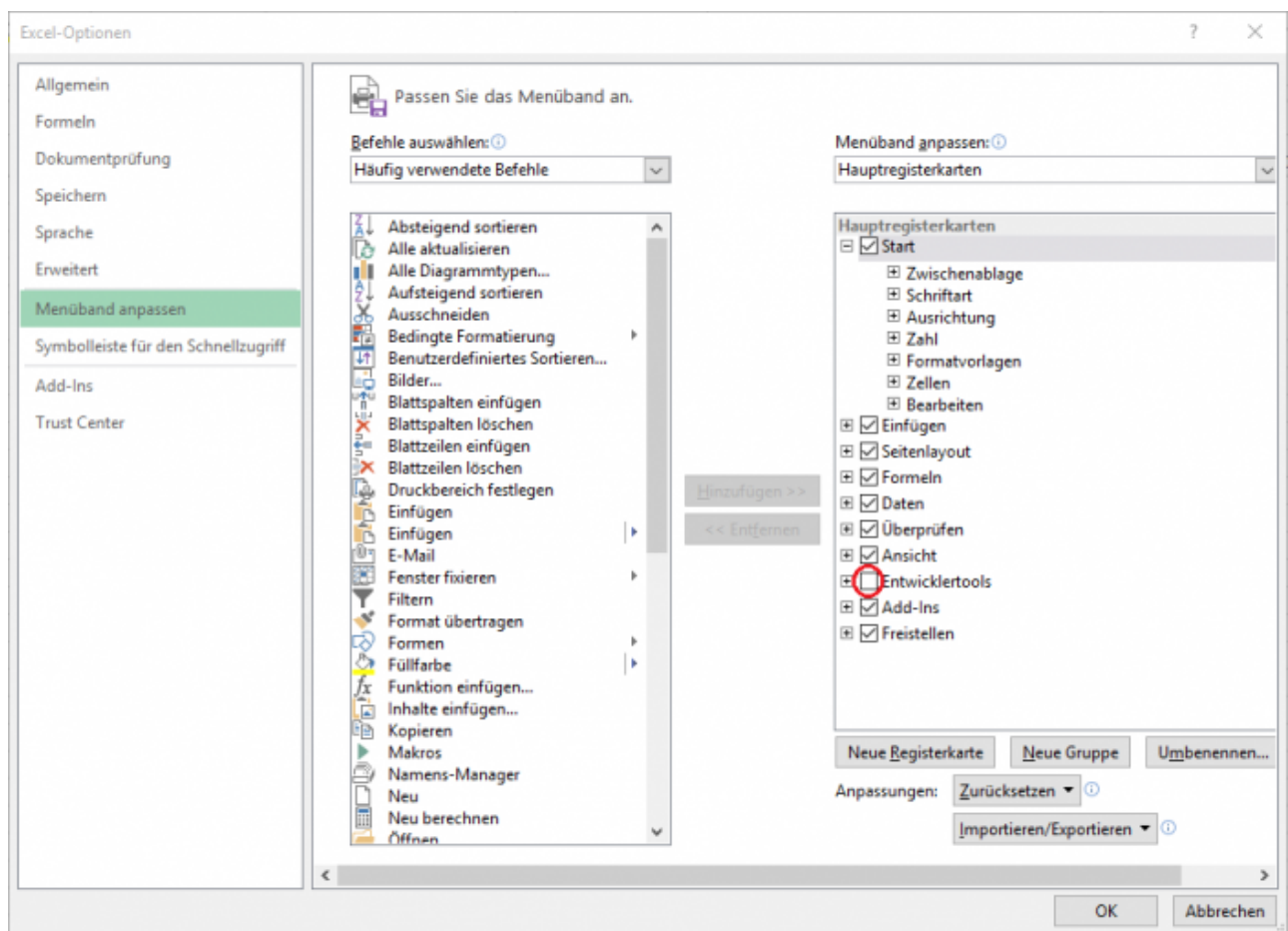
# = Adresse

# Erste Schritte mit dem IP-S7-Link .Net Advanced COM in Microsoft Excel

© by TIS

Nach der Installation des IP-S7-Link .Net Advanced COM mit dem Windows Installer-Paket müssen Sie die folgenden Schritte ausführen, um die Bibliothek in Ihrer Microsoft Excel-Arbeitsmappe zu integrieren und die API in Ihren VBA-Makros verwenden zu können:

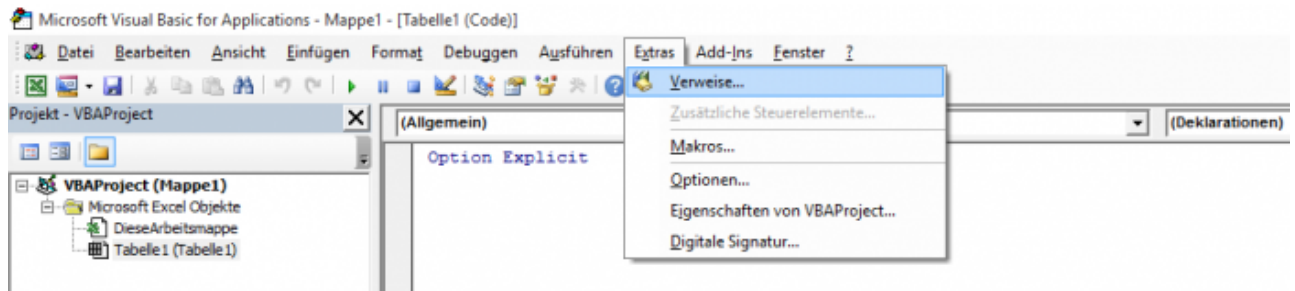
1. Vergewissern Sie sich, dass Excel die Entwicklertools-Menüleiste anzeigt. Gehen Sie dazu zu den Excel-Optionen und wählen Sie „Menüband anpassen“ aus. Aktivieren Sie anschließend das Menü der „Entwicklertools“.



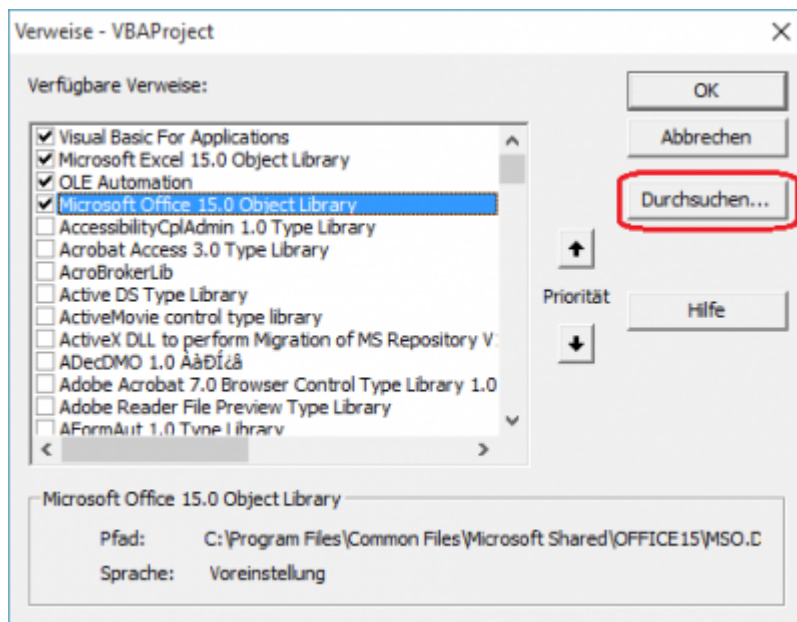
2. Wählen Sie die Menüleiste der Entwicklertools aus. Anschließend klicken Sie auf Code anzeigen. Dies öffnet die integrierte Entwicklungsumgebung (IDE), um Ihre Visual Basic für Applikationen (VBA) Makros zu entwickeln.



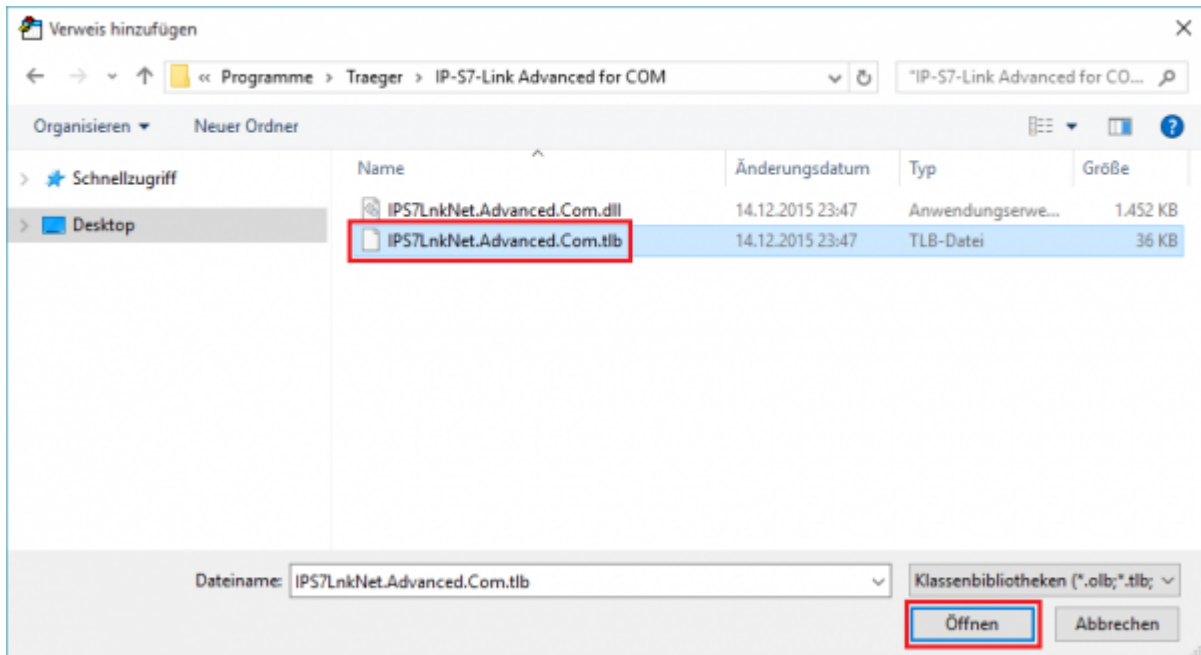
3. Wählen Sie den Menüpunkt „Extras“. Klicken Sie dann auf den Menüpunkt „Verweise“. Dies öffnet den Verweis-Dialog. Mithilfe dieses Dialogfelds können Sie Ihrer Excel-Arbeitsmappe zusätzliche Bibliotheken hinzufügen, um ihre Funktionalität bei der Entwicklung Ihres VBA-Makros zu nutzen.



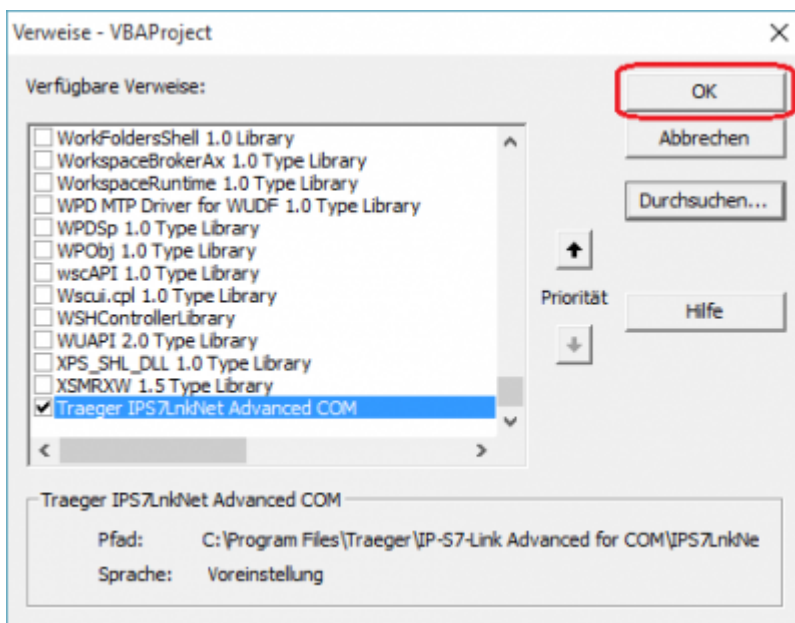
4. Wählen Sie die Schaltfläche „Durchsuchen“, um die Typbibliotheksdatei (\*.tlb) aufzurufen, die die Bibliothek (\*.dll) beschreibt. Diese Datei wird von der IDE verwendet, um die von der Bibliothek bereitgestellten Klassen, Methoden usw. zu erkennen.



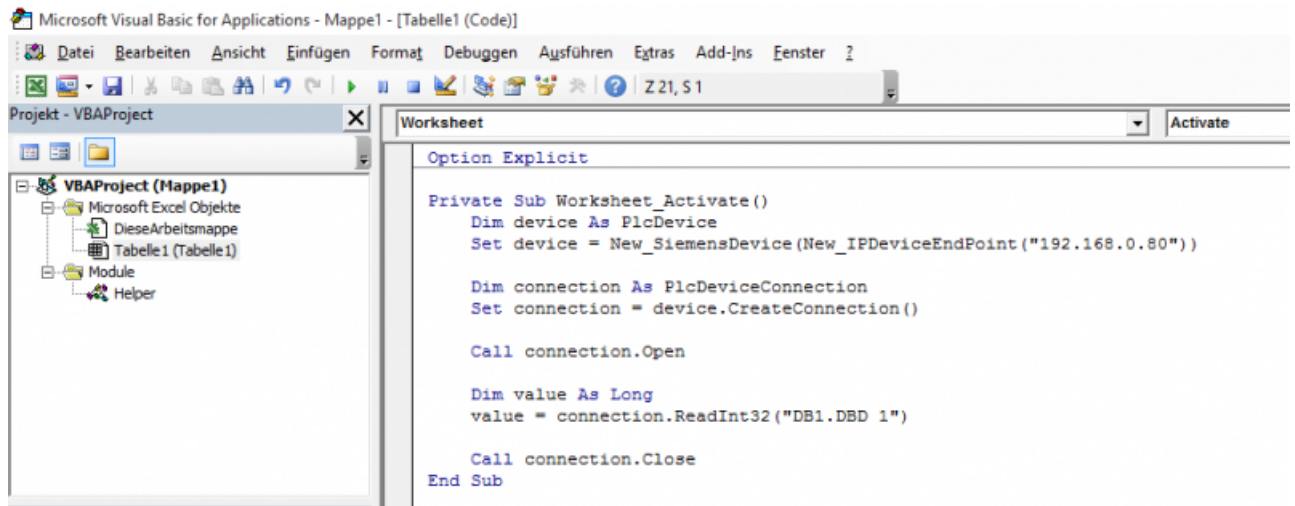
5. Navigieren Sie zu dem Installationsordner den Sie während der Einrichtung ausgewählt haben, das Standard-Installationsverzeichnis ist: „C:\Programme\IP-S7-Link Advanced for COM“. Wenn der entsprechende Installationsordner geöffnet wurde, wählen Sie die IPS7LnkNet.Advanced.Com.tlb Datei aus und bestätigen mit dem Öffnen-Button.



6. Nach dem Sie die IP57LnkNet.Advanced.Com.tlb ausgewählt haben, werden Sie nun einen Eintrag in der Referenzliste mit dem Namen „IP57LnkNet Advanced COM“ finden, die bereits aktiviert sein sollte. Falls es nicht aktiviert ist, aktivieren Sie dies. Bestätigen Sie den neuen Verweis, indem Sie auf die Schaltfläche OK klicken.



7. Nachdem die gewünschte Referenz hinzugefügt wurde, können Sie nun mit unseren Samples beginnen.





# IP-S7-Link .Net Advanced COM Code Snippets

© by TIS

Die folgenden Codeausschnitte bedeuten, dass Sie die Datei Helper.bas verwenden, die in den Samples enthalten ist und mit der Installation der Bibliothek installiert wird.

## Device Provider

Verwenden Sie einen der verschiedenen SPS-Geräteanbieter, indem Sie eine Instanz der entsprechenden PlcDevice-Klassenableitungen instanziiieren.

```
Dim device As PlcDevice
Set device = New_SiemensDevice(New_IPDeviceEndPoint("192.168.0.80"))
```

## Device Konfiguration

Nachdem eine Instanz von (z. B.) die SiemensDevice-Klasse erstellt wurde, verwenden Sie einfach die von der Klasse bereitgestellte Property, um die entsprechenden Geräte-Metadaten einzurichten.

```
Dim device As SiemensDevice
Set device = New_SiemensDevice(New_IPDeviceEndPoint("192.168.0.80"))

device.Type = SiemensDeviceType_S71200
device.ChannelType = SiemensChannelType_OperationPanel
```

In den meisten Szenarien ist es schon genug, um ein Gerät mit dem entsprechenden Gerätetyp im Konstruktor des Geräts zu erstellen, da der Default-Kanal meistens den gemeinsamen Anforderungen entspricht.

```
Dim device As SiemensDevice
Set device = New_SiemensDevice(New_IPDeviceEndPoint("192.168.0.80"),
SiemensDeviceType_S71200)
```

## Device End Point

Das Framework bietet die Möglichkeit, verschiedene Typen von end points über die PlcDeviceEndPoint-Klasse zu definieren. Die am häufigsten verwendete End point-Implementierung ist die IPDeviceEndPoint-Klasse. Mit dieser Klasse können Sie die IP-Adresse und optional die Rack- und Steckplatznummer der SPS angeben.

```

Dim endPoint As PlcDeviceEndPoint
Set endPoint = New_IPDeviceEndPoint("192.168.0.80")

' Alternatively also specify the rack.
Dim endPointWithRack As PlcDeviceEndPoint
Set endPointWithRack = New_IPDeviceEndPoint("192.168.0.80", 0)

' Alternatively also specify the rack and slot.
Dim endPointWithRackSlot As PlcDeviceEndPoint
Set endPointWithRackSlot = New_IPDeviceEndPoint("192.168.0.80", 0, 2)

```

## Device Connection

Nach dem Erstellen einer Instanz eines der PlcDevice-Klassen-Derivate erstellen Sie einfach eine neue Verbindung, die mit dem dargestellten Gerät verknüpft ist. Wenn Sie eine Instanz mit dieser Factory-Methode erstellen, wird die vom Provider abhängige Implementierung der PlcDeviceConnection-Klasse zurückgegeben.

```

Dim device As PlcDevice
Set device = New_SiemensDevice(New_IPDeviceEndPoint("192.168.0.80"))

Dim connection As PlcDeviceConnection
Set connection = device.CreateConnection()

```

## Verbindungsstatus behandeln

Um den aktuellen Konnektivitätsstatus der PlcDeviceConnection-Klasseninstanz abzurufen, verwenden Sie die State-Property, um das entsprechende PlcDeviceConnectionState-Enumerationmember zu erhalten.

```

If (connection.State = PlcDeviceConnectionState_Connected) Then
    ' ...
End If

```

Um die Zustandsübergänge zu behandeln, verwenden Sie eines oder mehrere der Status spezifischen Events der PlcDeviceConnection-Klasseninstanz.

```

Private WithEvents m_connection As PlcDeviceConnection

```

Um das Event zu behandeln, deklarieren Sie den übereinstimmenden Event-Handler wie folgt:

```

Private Sub m_connection_Connected(ByRef sender As Object, ByVal e As IComEventArgs)
    Debug.Print "Connection established!"
End Sub

```

## Werte lesen

Verwenden Sie zum Lesen eines einzelnen Werts eine der Read-Methoden der PlcDeviceConnection-Klasse.

```

Dim value As Long
value = connection.ReadInt32("DB1.DB1 1")

```

Um ein Array von Werten zu lesen, verwenden Sie die zusätzlichen Leseüberladungen:

```
Dim values() As Long
values = connection.ReadInt32Array("DB1.DBD 1", 3)
```

## Werte schreiben

Um einen einzelnen Wert zu schreiben, verwenden Sie eine der Write-Methoden der PlcDeviceConnection-Klasse.

```
Call connection.WriteInt32("DB1.DBD 1", 123)
```

Um ein Array von Werten zu schreiben, verwenden Sie die zusätzlichen Write-Überladungen:

```
Dim values(3) As Long
values(0) = 123
values(1) = 456
values(2) = 789

Call connection.WriteInt32Array("DB1.DBD 1", values)
```

# IP-S7-Link .Net Advanced COM Frequently Asked Questions (FAQs)

© by TIS

## Microsoft Excel Exceptions

### **Situation:**

Excel erkennt einen importierten Typ oder eine Methode nicht mehr.

### **Reason**

Durch Deinstallieren und Neuinstallieren einer Excel-importierten Bibliothek kann dies die Bibliotheksregistrierung beeinträchtigen, so dass Excel möglicherweise nicht mehr in der Lage ist, einige in der Bibliothek enthaltene Typen- und Methodeninformationen aufzulösen.

### **Lösung**

- Öffnen Sie das Dialogfeld Referenzen.
- Suchen Sie die importierte Bibliothek, deren Typ oder Methode von Excel hervorgehoben wurde.
- Entfernen Sie die Referenz aus der Excel-Arbeitsmappe in der Bibliothek.
- Übernehmen Sie die Entfernung mit OK.
- Speichern Sie die Excel-Arbeitsbuch-Änderungen (dies ist wichtig).
- Dann öffnen Sie das Dialogfenster Referenzen.
- Wählen Sie die zuvor entfernte Bibliothek.
- Übernehmen Sie diesen Schritt mit OK.
- Speichern Sie die Änderungen der Excel-Arbeitsmappe.
- Nun sollte alles wieder wie erwartet funktionieren.

# Inhaltsverzeichnis

<b>Kapitelübersicht</b>	1
<b>Was ist das IP-S7-Link .Net Advanced COM?</b>	2
<b>Verwandte Themen</b>	2
Operand	3
Datentypen	3
<b>Device Provider</b>	9
<b>Device Konfiguration</b>	9
<b>Device End Point</b>	9
<b>Device Connection</b>	10
<b>Verbindungsstatus behandeln</b>	10
<b>Werte lesen</b>	10
<b>Werte schreiben</b>	11
<b>Microsoft Excel Exceptions</b>	12

