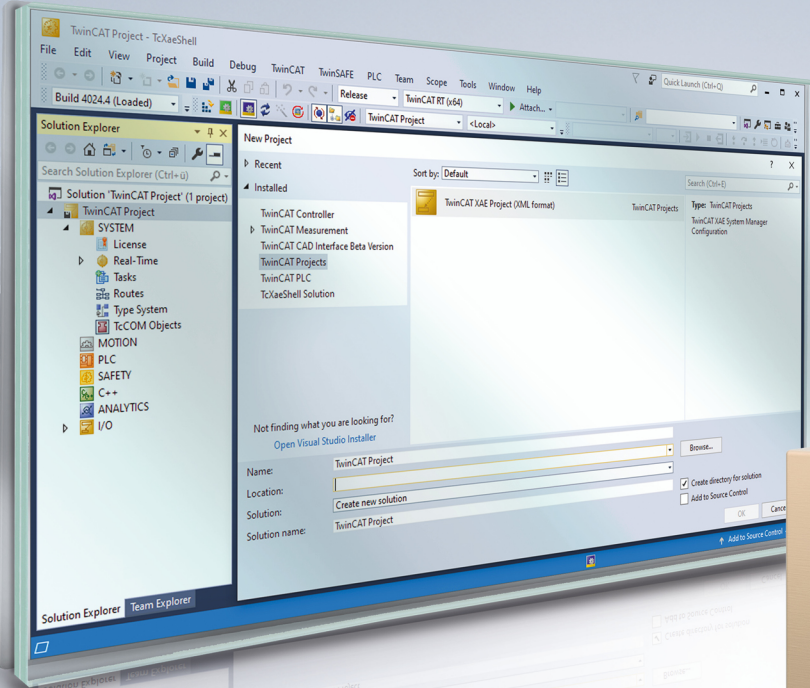


BECKHOFF New Automation Technology

Handbuch | DE

TF8020

TwinCAT 3 | BACnet



Inhaltsverzeichnis

1	Vorwort.....	9
1.1	Hinweise zur Dokumentation	9
1.2	Zu Ihrer Sicherheit.....	10
1.3	Hinweise zur Informationssicherheit	11
2	Einführung	12
2.1	Übersicht	13
2.2	Systemvoraussetzungen	13
3	Übersicht über BACnet-Eigenschaften	14
3.1	Objekttypen Input, Output, Value	14
3.2	Die am häufigsten verwendeten BACnet-Eigenschaften	14
3.2.1	Object_Identifier	14
3.2.2	Object_Name	15
3.2.3	Object_Type	16
3.2.4	Present_Value.....	16
3.2.5	Description	16
3.2.6	Device_Type	16
3.2.7	Status_Flags	16
3.2.8	Event_State.....	16
3.2.9	Reliability	17
3.2.10	Out_of_Service	17
3.2.11	Update_Interval.....	17
3.2.12	Units	17
3.2.13	Min_Pres_Value.....	17
3.2.14	Max_Pres_Value.....	17
3.2.15	Resolution	17
3.2.16	COV_Increment	17
3.2.17	Time_Delay	18
3.2.18	Notification_Class	18
3.2.19	High_Limit	18
3.2.20	Low_Limit.....	18
3.2.21	Deadband.....	18
3.2.22	Limit_Enable	18
3.2.23	Event_Enable.....	18
3.2.24	Acked_Transitions.....	19
3.2.25	Notify_Type	19
3.2.26	Event_Time_Stamps.....	19
3.2.27	Event_Message_Texts.....	19
3.2.28	Profile_Name	19
3.2.29	Event_Message_Texts_Config	19
3.2.30	Event_Detection_Enable.....	19
3.2.31	Event_Algorithm_Inhibit_Ref.....	19
3.2.32	Event_Algorithm_Inhibit	19
3.2.33	TimeDelay_Normal	19
3.2.34	Reliability_Evaluation_Inhibit	20

3.2.35	Property_List	20
3.2.36	Priority_Array	20
3.2.37	Relinquish_Default	20
3.2.38	Inactive_Text	20
3.2.39	Active_Text	20
3.2.40	Change_Of_State_Time	20
3.2.41	Change_Of_State_Count	20
3.2.42	Time_Of_State_Count_Reset	20
3.2.43	Elapsed_Active_Time	21
3.2.44	Time_Of_Active_Time_Reset	21
3.2.45	Alarm_Value	21
3.2.46	Minimum_Off_Time	21
3.2.47	Minimum_On_Time	21
3.2.48	Feedback_Value	21
3.2.49	Number_Of_States	21
3.2.50	State_Text	21
3.2.51	Alarm_Values	21
3.2.52	Fault_Values	21
3.2.53	Date_List	22
3.2.54	Weekly_Schedule	22
3.2.55	Exception_Schedule	22
3.3	Zum Verständnis des PICS-Dokuments	22
4	Quickstart	23
4.1	Anlegen des TwinCAT-Projekts	23
4.1.1	Auswahl des Zielsystems	24
4.1.2	Anlegen eines BACnet-Adapters und -Servers	29
4.1.3	Einstellungen für IP-Adresse	31
4.1.4	Anpassen der BACnet-Server-Einstellungen	33
4.1.5	Persistenz	34
4.2	Anlegen des BACnet SPS-Projekts	34
4.2.1	Anlegen des SPS-Projekts	35
4.2.2	Testen des BACnet-Servers	43
4.2.3	Testen von BACnet mit dem Systemmanager	44
4.2.4	Testen von BACnet mit einem BACnet Explorer	46
5	SPS-Bibliothek: Tc3_BACnetRev14	51
5.1	DUTs	51
5.1.1	Aufzählungen	51
5.1.2	Schnittstellen	51
5.1.3	Typen	51
5.2	GVLs	51
5.2.1	Version	51
5.2.2	BACnet_Globals	52
5.2.3	BACnet_Param	52
5.3	POUs	53
5.3.1	Namenskonventionen	54

5.3.2	FB_BACnet_Adapter.....	54
6	Programmierung eines BACnet-Servers	60
6.1	BACnet-Objekt POU.....	60
6.1.1	Funktionsbausteine ohne Suffix	61
6.1.2	Primitive Value-Objekttypen.....	62
6.2	Typische BACnet-Szenarien	63
6.2.1	Befehlspriorisierung	63
6.2.2	Event Reporting	64
6.2.3	Scheduling	64
6.2.4	Trendaufzeichnung	65
6.2.5	Eventaufzeichnung.....	65
6.2.6	Regelkreise	65
6.2.7	TimeSynchronization.....	66
6.2.8	Abrufen von Diagnoseinformationen.....	67
6.2.9	Scannen anderer BACnet-Devices	67
6.2.10	Erstellen einer strukturierten Ansicht (AKS).....	68
6.2.11	Verknüpfung von Hardware über Attribute.....	70
6.3	Wichtige Hinweise zur Benutzung der Bibliothek.....	71
6.3.1	Deklaration von Eigenschaften beim Start.....	71
6.4	Parameter-Dialoge	72
6.4.1	BACnet-Funktionsbausteine immer zyklisch aufrufen.....	73
6.4.2	Aufruf von BACnet-Funktionsbausteinen mit der gleichen Zykluszeit.....	73
6.5	Berechnung des Routerspeichers.....	74
6.5.1	Beispiel für die Berechnung des erforderlichen Routerspeichers	74
6.6	Spezifische Funktionen	75
6.6.1	Generierung einer EDE-Datei	75
6.6.2	Zykluszeit-Überschreitungszähler	76
6.6.3	Aktivieren / Deaktivieren von Eigenschaften.....	76
6.6.4	Schreibgeschützte Eigenschaften.....	76
6.6.5	Freigabe des Schreibzugriffs durch die SPS	76
6.6.6	Hinzufügen eines Empfängers zu einer Empfängerliste	77
6.6.7	Verwendung von UTF-8-Zeichen	77
6.7	FB_BACnet_Server.....	77
6.8	FB_BACnet_Device	78
6.9	IO Code.....	78
6.10	Empfohlener Arbeitsablauf / BACnet-Persistenz	79
6.11	Persistenz	79
6.12	Uhrzeitsynchronisation.....	84
6.13	Empfohlene Zykluszeit der PLC-Task.....	86
7	Programmierung eines BACnet-Clients.....	87
7.1	Schreibmodus	87
7.2	Lesemodus.....	87
7.2.1	Automatikbetrieb	87
7.2.2	Anwenden des Lesemodus auf das gesamte Peer-Gerät	88
7.2.3	Anwenden des Lesemodus auf einzelne Objekte	88

7.2.4	COV-Reporting.....	89
7.3	Client POU's	89
7.4	FB Code	91
7.5	Aufrufen der Funktion FB Code	91
7.5.1	Aufruf von FB Code aus dem Dialog Scan	91
7.5.2	Aufruf von FB Code aus dem Dialog Cyclic Data	92
7.6	Der Dialog Client FB Code	92
7.6.1	Das Fenster Select Objects	93
7.6.2	Unterstützte Services	93
7.6.3	Service-Einstellungen	94
7.6.4	Das Fenster Settings.....	95
7.6.5	Das Fenster FB Code	95
7.6.6	Verwendung des von FB_Code erstellten FB	96
7.6.7	FB_Code verwenden, um Gerätevorlagen zu erstellen	97
7.7	Client-Variablen.....	98
7.8	Entfernte Zeitplanobjekte	98
7.9	Azyklisches Lesen.....	99
7.9.1	Beispiel FB_BACnetRM_ReadProperty	99
7.9.2	Beispiel FB_BACnetRM_ReadPropertyEx.....	99
7.10	Azyklisches Schreiben	100
7.10.1	Beispiel FB_BACnetRM_WriteProperty	100
7.10.2	Beispiel FB_BACnetRM_WritePropertyEx.....	100
7.11	Überwachung einer Client-Verbindung	101
7.11.1	Beispiel: Die Verbindung wurde erfolgreich hergestellt.....	101
7.11.2	Beispiel: Die Verbindung ist unterbrochen	102
7.12	Verwendung von ReadPropertyMultiple.....	103
8	Dynamic Object Manager	105
8.1	FB_BACnet_DynObjectManager	105
8.2	Zyklische Aufrufe.....	105
8.3	Vordefinierter Objektpool	105
8.4	Beispiel.....	106
8.5	Initialisierung der dynamischen Objekte abschließen	107
8.6	Anlegen und Löschen eigener BACnet Funktionsbausteine (FB).....	107
9	Beispiele	109
9.1	Variablendeklaration	109
9.2	BACnet Properties	109
9.3	Verknüpfung mit dem Attribut 'TcLinkTo'.....	110
9.4	Property Auswahl und Schreibschutz	111
9.5	Prioritätssteuerung	112
9.6	Prioritäten zurücksetzen.....	113
9.7	Regelkreise und Loop Objekte	113
9.8	Alarmempfänger einrichten	114
9.9	Empfang von Alarmen und Events aus anderen Geräten.....	115
9.10	Vorwarngrenzen.....	116
9.11	Kalender und Zeitschaltfunktionen.....	117

9.12	Logging Objekte	120
9.13	Verarbeitung des Logspeichers in der SPS	121
9.14	Primitive Value-Objekte	122
9.15	Structured View-Objekte	124
9.16	Arrayinitialisierung	125
10	Support und Service	127

1 Vorwort

1.1 Hinweise zur Dokumentation

Diese Beschreibung wendet sich ausschließlich an ausgebildetes Fachpersonal der Steuerungs- und Automatisierungstechnik, das mit den geltenden nationalen Normen vertraut ist.

Zur Installation und Inbetriebnahme der Komponenten ist die Beachtung der Dokumentation und der nachfolgenden Hinweise und Erklärungen unbedingt notwendig.

Das Fachpersonal ist verpflichtet, stets die aktuell gültige Dokumentation zu verwenden.

Das Fachpersonal hat sicherzustellen, dass die Anwendung bzw. der Einsatz der beschriebenen Produkte alle Sicherheitsanforderungen, einschließlich sämtlicher anwendbaren Gesetze, Vorschriften, Bestimmungen und Normen erfüllt.

Disclaimer

Diese Dokumentation wurde sorgfältig erstellt. Die beschriebenen Produkte werden jedoch ständig weiterentwickelt.

Wir behalten uns das Recht vor, die Dokumentation jederzeit und ohne Ankündigung zu überarbeiten und zu ändern.

Aus den Angaben, Abbildungen und Beschreibungen in dieser Dokumentation können keine Ansprüche auf Änderung bereits gelieferter Produkte geltend gemacht werden.

Marken

Beckhoff®, TwinCAT®, TwinCAT/BSD®, TC/BSD®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC®, XTS® und XPlanar® sind eingetragene und lizenzierte Marken der Beckhoff Automation GmbH.

Die Verwendung anderer in dieser Dokumentation enthaltenen Marken oder Kennzeichen durch Dritte kann zu einer Verletzung von Rechten der Inhaber der entsprechenden Bezeichnungen führen.

Patente

Die EtherCAT-Technologie ist patentrechtlich geschützt, insbesondere durch folgende Anmeldungen und Patente:

EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702

mit den entsprechenden Anmeldungen und Eintragungen in verschiedenen anderen Ländern.

EtherCAT®

EtherCAT® ist eine eingetragene Marke und patentierte Technologie lizenziert durch die Beckhoff Automation GmbH, Deutschland

Copyright

© Beckhoff Automation GmbH & Co. KG, Deutschland.

Weitergabe sowie Vervielfältigung dieses Dokuments, Verwertung und Mitteilung seines Inhalts sind verboten, soweit nicht ausdrücklich gestattet.

Zuwiderhandlungen verpflichten zu Schadenersatz. Alle Rechte für den Fall der Patent-, Gebrauchsmuster- oder Geschmacksmustereintragung vorbehalten.

1.2 Zu Ihrer Sicherheit

Sicherheitsbestimmungen

Lesen Sie die folgenden Erklärungen zu Ihrer Sicherheit.
Beachten und befolgen Sie stets produktspezifische Sicherheitshinweise, die Sie gegebenenfalls an den entsprechenden Stellen in diesem Dokument vorfinden.

Haftungsausschluss

Die gesamten Komponenten werden je nach Anwendungsbestimmungen in bestimmten Hard- und Software-Konfigurationen ausgeliefert. Änderungen der Hard- oder Software-Konfiguration, die über die dokumentierten Möglichkeiten hinausgehen, sind unzulässig und bewirken den Haftungsausschluss der Beckhoff Automation GmbH & Co. KG.

Qualifikation des Personals

Diese Beschreibung wendet sich ausschließlich an ausgebildetes Fachpersonal der Steuerungs-, Automatisierungs- und Antriebstechnik, das mit den geltenden Normen vertraut ist.

Signalwörter

Im Folgenden werden die Signalwörter eingeordnet, die in der Dokumentation verwendet werden. Um Personen- und Sachschäden zu vermeiden, lesen und befolgen Sie die Sicherheits- und Warnhinweise.

Warnungen vor Personenschäden

GEFAHR

Es besteht eine Gefährdung mit hohem Risikograd, die den Tod oder eine schwere Verletzung zur Folge hat.

WARNUNG

Es besteht eine Gefährdung mit mittlerem Risikograd, die den Tod oder eine schwere Verletzung zur Folge haben kann.

VORSICHT

Es besteht eine Gefährdung mit geringem Risikograd, die eine mittelschwere oder leichte Verletzung zur Folge haben kann.

Warnung vor Umwelt- oder Sachschäden

HINWEIS

Es besteht eine mögliche Schädigung für Umwelt, Geräte oder Daten.

Information zum Umgang mit dem Produkt



Diese Information beinhaltet z. B.:
Handlungsempfehlungen, Hilfestellungen oder weiterführende Informationen zum Produkt.

1.3 Hinweise zur Informationssicherheit

Die Produkte der Beckhoff Automation GmbH & Co. KG (Beckhoff) sind, sofern sie online zu erreichen sind, mit Security-Funktionen ausgestattet, die den sicheren Betrieb von Anlagen, Systemen, Maschinen und Netzwerken unterstützen. Trotz der Security-Funktionen sind die Erstellung, Implementierung und ständige Aktualisierung eines ganzheitlichen Security-Konzepts für den Betrieb notwendig, um die jeweilige Anlage, das System, die Maschine und die Netzwerke gegen Cyber-Bedrohungen zu schützen. Die von Beckhoff verkauften Produkte bilden dabei nur einen Teil des gesamtheitlichen Security-Konzepts. Der Kunde ist dafür verantwortlich, dass unbefugte Zugriffe durch Dritte auf seine Anlagen, Systeme, Maschinen und Netzwerke verhindert werden. Letztere sollten nur mit dem Unternehmensnetzwerk oder dem Internet verbunden werden, wenn entsprechende Schutzmaßnahmen eingerichtet wurden.

Zusätzlich sollten die Empfehlungen von Beckhoff zu entsprechenden Schutzmaßnahmen beachtet werden. Weiterführende Informationen über Informationssicherheit und Industrial Security finden Sie in unserem <https://www.beckhoff.de/secguide>.

Die Produkte und Lösungen von Beckhoff werden ständig weiterentwickelt. Dies betrifft auch die Security-Funktionen. Aufgrund der stetigen Weiterentwicklung empfiehlt Beckhoff ausdrücklich, die Produkte ständig auf dem aktuellen Stand zu halten und nach Bereitstellung von Updates diese auf die Produkte aufzuspielen. Die Verwendung veralteter oder nicht mehr unterstützter Produktversionen kann das Risiko von Cyber-Bedrohungen erhöhen.

Um stets über Hinweise zur Informationssicherheit zu Produkten von Beckhoff informiert zu sein, abonnieren Sie den RSS Feed unter <https://www.beckhoff.de/secinfo>.

2 Einführung



Die neu entwickelte TwinCAT-Bibliothek *Tc3_BACnetRev14* implementiert einen vollständig objektorientierten Engineering- und Konfigurationsprozess, um eine komfortable Projektierung von Beckhoff-BACnet-Steuerungen zu ermöglichen.

Für die Verwendung der Bibliothek sind keine OOP-Kenntnisse (objektorientierte Programmierung) erforderlich. Für fortgeschrittene Anwender bietet diese Architektur jedoch flexibel erweiterbare Optionen.

Bislang wurde mit der älteren Bibliothek *Tc2_BACnetRev12* die Verbindung zwischen den SPS-Variablen und dem BACnet-Supplement über das Automapping-Verfahren hergestellt. Die BACnet-Funktionsbausteine wurden durch Kommentare ergänzt, die die Objekteigenschaften beschreiben und von der Map- oder Remap-Funktion im System Manager geparkt werden. Dieses Verfahren ist in TwinCAT 4024 weiterhin verfügbar und kann durch Hinzufügen der Bibliothek *Tc2_BACnetRev12* genutzt werden. Bitte beachten Sie, dass diese Funktion in zukünftigen TwinCAT-Versionen möglicherweise nicht mehr unterstützt wird!

Die Verwendung der beiden Bibliotheken *Tc3_BACnetRev14* und *Tc2_BACnetRev12* in einem Projekt wird nicht unterstützt.

BACnet® is a registered trademark of ASHRAE.

2.1 Übersicht

Die Komponenten werden von zwei Bibliotheken bereitgestellt:

Tc3_BACnetRev14: Diese Bibliothek enthält Funktionsbausteine, Datentypen, globale Variablen und Parameter, die für das Engineering von BACnet-Objekten benötigt werden. Diese Bibliothek wird mit der TwinCAT-Installation (4024.11 oder höher) als kompilierte Bibliothek (ohne Quellcode) bereitgestellt. Der nicht kompilierte Quellcode wird auf Anfrage zur Verfügung gestellt (E-Mail: buildingautomation@beckhoff.com).

Bitte beachten Sie, dass das Laden nicht kompilierter Bibliotheken einige Zeit in Anspruch nimmt (die Bibliothek ist in diesem Fall mit einem Uhrensymbol gekennzeichnet). Warten Sie, bis das Uhrensymbol verschwunden ist, bevor Sie die Bibliothek benutzen.

Tc3_BA2_Common: Diese Bibliothek enthält alle Funktionsbausteine, Datentypen, globalen Variablen und Parameter, die von BACnet und TwinCAT 3 Building Automation (Tc3BA) häufig verwendet werden. So sind z. B. technische Einheiten und ein PID-Regler in dieser Bibliothek verfügbar.

Bitte fügen Sie diese Bibliothek zusätzlich zu *Tc3_BACnet_Rev14* hinzu, wenn Sie diese Optionen nutzen möchten.



Einige Funktionsbausteine können zusätzliche Bibliotheken erfordern:

Tc2_Utilities: Enthält allgemeine Hilfsfunktionen, z. B. zum Zugriff auf das Betriebssystem, zum Erstellen von CSV-Dateien oder zum Konvertieren und Formatieren von Daten (z. B. UTF-8-Zeichensatzunterstützung).

Tc2_SUPS: Enthält Funktionen zur Verwendung einer unterbrechungsfreien Stromversorgung (USV).

2.2 Systemvoraussetzungen

Zielsystem: TwinCAT XAR 4024.17 oder höher

Engineering-PC: TwinCAT XAE 4024.17 oder höher.

3 Übersicht über BACnet-Eigenschaften

BACnet-Objekte werden mit Hilfe von Eigenschaften zusammengestellt, die entweder im BACnet-Standard spezifiziert sind oder proprietär sein können. Einige Eigenschaften sind obligatorisch, während andere optional sein können. Einige Eigenschaften müssen beschreibbar sein, andere können bei Bedarf beschreibbar gemacht werden. Der Konformitätscode im PICS-Dokument beschreibt, welche Eigenschaften zu welcher Gruppe gehören:

R = Erforderlich

O = Optional

W = Beschreibbar

Die Spalte "CC" beschreibt die Anforderungen durch den BACnet-Standard.
"WA" beschreibt die Schreibzugriff-Optionen in TwinCAT.

Die folgende Tabelle zeigt ein Beispielobjekt (Analog-Eingang, es sind nicht alle Eigenschaften dargestellt) aus dem PICS-Dokument (das Dokument kann hier heruntergeladen werden: www.beckhoff.com/bacnet).

Analog Input

Property	Data Type	CC	WA
Object_Identifier	BACnetObjectIdentifier	R	R
Object_Name	CharacterString	R	W
Object_Type	BACnetObjectType	R	R
Present_Value	REAL	R	R
Description	CharacterString	O	W
Device_Type	CharacterString	O	W
Status_Flags	BACnetStatusFlags	R	R
Event_State	BACnetEventState	R	R
Reliability	BACnetReliability	O	R

3.1 Objekttypen Input, Output, Value

Analoge, binäre und Multistate-Objekte sind die am häufigsten verwendeten Objekte. Analog-Objekte stellen eine Fließkomma-Information (REAL) dar, Binary-Objekte eine digitale Information (inaktiv/aktiv) und Multistate-Objekte einen Satz von mehreren Zuständen.

Input-Objekttypen werden typischerweise verwendet, um einen physischen Hardware-Eingang zu repräsentieren, der mit dem Gerät verbunden ist, z. B. ein Temperatur- oder Helligkeitssensor.

Output-Objekttypen repräsentieren typischerweise physikalische Ausgänge wie einen 0-10V-Ausgang oder einen Prozentsatz eines Ventilaktors.

Value-Objekttypen werden verwendet, um virtuelle Informationen wie einen Sollwert oder einen Regelparameter darzustellen.

3.2 Die am häufigsten verwendeten BACnet-Eigenschaften

Dieses Kapitel erklärt die Bedeutung der am häufigsten verwendeten BACnet-Objekteigenschaften. Eine vollständige Liste der von TwinCAT unterstützten Objekte, der unterstützten Eigenschaften dieser Objekte und des implementierten Konformitätscodes finden Sie im PICS-Dokument.

3.2.1 Object_Identifier

Diese Eigenschaft besteht aus zwei Elementen, einem 10-Bit `Object_Type` und einem 22-Bit `Object_Instance`. Der `Object_Type` ist im BACnet-Standard festgelegt:

```
AnalogInput :=0
AnalogOutput :=1
AnalogValue :=2
BinaryInput :=3
BinaryOutput :=4
BinaryValue :=5
Calendar :=6
Command :=7
Device :=8
EventEnrollment :=9
File :=10
Group :=11
Loop :=12
MultiStateInput :=13
MultiStateOutput :=14
NotificationClass :=15
Program :=16
Schedule :=17
Averaging :=18
MultiStateValue :=19
TrendLog :=20
LifeSafetyPoint :=21
LifeSafetyZone :=22
Accumulator :=23
PulseConverter :=24
EventLog := 25
GlobalGroup := 26
TrendLogMultiple := 27
LoadControl := 28
StructuredView := 29
AccessDoor := 30
unassigend31 := 31
AccessCredential := 32
AccessPoint := 33
AccessRights := 34
AccessUser := 35
AccessZone := 36
CredentialDataInput := 37
NetworkSecurity := 38
BitStringValue := 39
CharacterStringValue := 40
DatePatternValue := 41
DateValue := 42
DateTimePatternValue := 43
DateTimeValue := 44
IntegerValue := 45
LargeAnalogValue := 46
OctetStringValue := 47
PositiveIntegerValue := 48
TimePatternValue := 49
TimeValue := 50
NetworkPort := 51
AlertEnrollment := 52
Channel := 53
```

Die `Object_Instance` wird bei der Erstellung des Objekts angegeben und muss pro Objekttyp eindeutig sein. In der TwinCAT 3-Bibliothek **Tc3_BACnetRev14** beginnen die Instanznummern standardmäßig bei 10000. Der höchste 22-Bit-Wert = 4194303 kann nicht Objektinstanzen zugewiesen werden. Dieser Wert ist vom BACnet-Standard als Platzhalter für den Fall reserviert, dass die tatsächliche Objektinstanznummer nicht bekannt ist.

Im Falle des `Device_Objects` muss die `Object_Instance` innerhalb des gesamten BACnet-Internetworks (über alle BACnet Data-Link-Layer) eindeutig sein.

3.2.2 Object_Name

Diese Eigenschaft enthält den Namen des Objekts. In vielen Fällen enthält diese Eigenschaft das Anlagenkennzeichnungssystem (AKS), d. h. es handelt sich um eine projekt- bzw. kundenspezifische Namenskonvention für Datenpunkte.

Der `Object_Name` muss mindestens ein druckbares Zeichen enthalten, das innerhalb des Geräts eindeutig sein muss. Im Falle des `Object_Name` des Device-Objekts muss diese Eigenschaft innerhalb des gesamten BACnet-Internetworks (über alle BACnet Data-Link-Layer) eindeutig sein. Der Standard-Zeichensatz in TwinCAT ist UTF-8. Die maximale Länge dieses Strings in TwinCAT ist 255.

3.2.3 Object_Type

Diese Eigenschaft stellt die 10-Bit-Typ-Information des BACnet Object Identifier dar. Wenn diese Eigenschaft angefordert wird, enthält die Antwort nur den Objekttyp. Wird die Eigenschaft ObjectIdentifier angefordert, wird der 32-Bit-Identifier (10-Bit-Typ und 22-Bit-Instanznummer) in der Antwort zurückgegeben.

3.2.4 Present_Value

Diese Eigenschaft ist sehr wahrscheinlich die wichtigste Eigenschaft. Sie stellt den aktuellen Prozesswert des Objekts dar.

Der Datentyp dieser Eigenschaft ist:

Analog-Objekte: REAL (32-Bit Fließkommazahl einfacher Genauigkeit gemäß ANSI-IEEE 754)

Binary-Objekte: ENUMERATED 0=inaktiv, 1=aktiv *

Multistate-Objekte: UNSIGNED INT

* Bitte beachten Sie, dass die Bibliothek einen BOOL-Datentyp verwendet und eine automatische Konvertierung in die BACnet-Aufzählung bietet.

3.2.5 Description

Diese optionale Eigenschaft kann eine zusätzliche Textinformation zur Beschreibung des Objekts enthalten. Der Standard-Zeichensatz ist UTF-8. Die maximale Länge in TwinCAT beträgt 255 Zeichen.

3.2.6 Device_Type

(Verfügbar in Ein- und Ausgangsobjekttypen)

Diese optionale Texteneigenschaft kann verwendet werden, um den Hardwaretyp anzugeben, z. B. einen PT1000-Sensor. Sie kann weitere Informationen wie die Artikelnummer oder ähnliche Informationen über die angeschlossene Hardware enthalten. Der Standard-Zeichensatz ist UTF-8. Die maximale Länge in TwinCAT beträgt 255 Zeichen.

3.2.7 Status_Flags

Diese Eigenschaft steht für vier boolesche Statusflags:

IN_ALARM: Im Falle eines aktiven Alarms wird dieses Flag auf TRUE gesetzt.

FAULT: Ist die Eigenschaft Reliability ungleich NO_FAULT_DETECTED, wird dieses Flag auf TRUE gesetzt. Zu den Fehlerzuständen gehören defekte Sensoren, Verknüpfungen usw.

OVERRIDDEN: Falls die Eigenschaft Present_Value des Objekts von der Anwendung lokal überschrieben wird, wird dieses Flag auf TRUE gesetzt. Die Werte der Eigenschaften Present_Value und Reliability werden nicht mehr von der Hardware entnommen. Die Festlegung des überschriebenen Zustands erfolgt lokal durch die Anwendung.

OUT_OF_SERVICE: Dieser Wert gibt den Zustand der Eigenschaft Out_of_Service an.

3.2.8 Event_State

Diese Eigenschaft stellt den Ereigniszustand im Falle eines aktiven Alarms dar, z. B. LOW_LIMIT. Im Normalzustand oder in Abwesenheit einer mit dem Objekt verbundenen Alarmierung muss der Wert NORMAL sein.

3.2.9 Reliability

Diese Eigenschaft gibt an, ob der Wert der Eigenschaft `Present_Value` zuverlässig ist (`NO_FAULT_DETECTED`) und wenn nicht, *warum* nicht (z. B. Kurzschluss, fehlender Sensor usw.).

3.2.10 Out_of_Service

Diese boolesche Eigenschaft zeigt an, ob das Objekt in Betrieb ist oder nicht. Im Falle von `Out_of_Service = TRUE` wird `Present_Value` nicht mehr von der Hardware verfolgt und `Present_Value` wird beschreibbar, auch für Eingangsobjekttypen. Dies ermöglicht die Festlegung eines Wertes im Falle defekter Hardware.

3.2.11 Update_Interval

Diese optionale Eigenschaft gibt die maximale Zeitspanne in Hundertstelsekunden zwischen Aktualisierungen des `Present_Value` an, falls der Eingang nicht außer Betrieb ist und nicht überschrieben wird.

3.2.12 Units

Diese aufgezählte Eigenschaft gibt die technische Einheit des Objekts an. Da diese Einheiten für BACnet und für allgemeine Gebäudeautomationszwecke verwendet werden, können die aufgezählten Werte aus der Bibliothek `Tc3_BA2_Common` abgerufen werden. Die Aufzählung ist im Abschnitt [DUTs/Enumerationen/Units/E_BA_Unit](#) zu finden.

3.2.13 Min_Pres_Value

Diese Eigenschaft gibt den niedrigsten Wert in technischen Einheiten an, der für die Eigenschaft `Present_Value` des Objekts zuverlässig ermittelt werden kann. Derzeit nicht in Analog-Value-Objekten verfügbar.

3.2.14 Max_Pres_Value

Diese Eigenschaft gibt den höchsten Wert in technischen Einheiten an, der für die Eigenschaft `Present_Value` des Objekts zuverlässig ermittelt werden kann. Derzeit nicht in Analog-Value-Objekten verfügbar.

3.2.15 Resolution

Diese Eigenschaft gibt die kleinste erkennbare Änderung der Eigenschaft `Present_Value` in technischen Einheiten an.

3.2.16 COV_Increment

Diese Eigenschaft gibt die Hysterese für Change-of-Value-Benachrichtigungen (bei Wertänderung) an. Wenn der neue Wert der Eigenschaft `Present_Value` den alten Wert um `COV_Increment` übersteigt, werden die Empfänger, die dieses Objekt abonniert haben, über die Wertänderung informiert.

Bitte beachten Sie: Ein `COV_Increment` von Null bedeutet, dass jede (auch kleine) Änderung gemeldet wird, was zu einer Nachrichtenflut im Netzwerk führen kann!

3.2.17 Time_Delay

Diese Eigenschaft gibt den Mindestzeitraum in Sekunden an, den `Present_Value` außerhalb des durch die Eigenschaften `High_Limit` und `Low_Limit` definierten Bereichs liegen muss, bevor ein `TO-OFFNORMAL`-Ereignis generiert wird, oder innerhalb desselben Bereichs, einschließlich der Eigenschaft `Deadband`, bevor ein `TO-NORMAL`-Ereignis generiert wird.

3.2.18 Notification_Class

Diese Eigenschaft gibt die Instanznummer des `Notification Class` Objekts für die Event-Notification-Distribution an.

3.2.19 High_Limit

Diese Eigenschaft gibt den Grenzwert an, den `Present_Value` überschreiten muss, bevor ein Ereignis generiert wird.

3.2.20 Low_Limit

Diese Eigenschaft gibt den Grenzwert an, unter den `Present_Value` fallen muss, bevor ein Ereignis generiert wird.

3.2.21 Deadband

Diese Eigenschaft gibt einen Bereich zwischen den Eigenschaften `High_Limit` und `Low_Limit` an, in dem `Present_Value` liegen muss, damit ein `TO-NORMAL`-Ereignis generiert wird.

Für die Ereigniserzeugung gelten die folgenden Bedingungen:

- `Present_Value` muss unter dem `High_Limit` minus `Deadband` liegen und
- `Present_Value` muss den `Low_Limit` plus `Deadband` überschreiten und
- `Present_Value` muss für einen Mindestzeitraum, der in der Eigenschaft `Time_Delay` angegeben ist, innerhalb dieses Bereichs bleiben und
- entweder das `HighLimitEnable`- oder das `LowLimitEnable`-Flag muss in der Eigenschaft `Limit_Enable` gesetzt sein und
- das `TO-NORMAL`-Flag muss in der Eigenschaft `Event_Enable` gesetzt sein.

3.2.22 Limit_Enable

Diese Eigenschaft gibt zwei Flags an, die die Einhaltung der oberen Alarmgrenze und unteren Alarmgrenze getrennt voneinander aktivieren und deaktivieren.



Bitte beachten Sie:

Bis zur BACnet Revision 12 gab es im BACnet Standard einen Fehler in der Beschreibung der Property Description. Bis zur Version 12 war nur Event Reporting aktiviert/deaktiviert. Ab Revision 13 wurde im BACnet-Standard diese Festlegung geändert. Nun wird die Beobachtung dieser Eigenschaften aktiviert / deaktiviert.

3.2.23 Event_Enable

Diese Eigenschaft gibt drei Flags an, die die Meldung der Ereignisse `TO-OFFNORMAL`, `TO-FAULT` und `TO-NORMAL` separat aktivieren und deaktivieren.

3.2.24 Acked_Transitions

Diese Eigenschaft gibt drei Flags an, die den Empfang von Bestätigungen für die Ereignisse `TO-OFFNORMAL`, `TO-FAULT` und `TO-NORMAL` separat anzeigen.

3.2.25 Notify_Type

Diese Eigenschaft gibt an, ob es sich bei den vom Objekt erzeugten Benachrichtigungen um Ereignisse oder Alarme handeln soll.

3.2.26 Event_Time_Stamps

Diese Eigenschaft gibt die Zeitpunkte der letzten Ereignisbenachrichtigungen für die Ereignisse `TO-OFFNORMAL`, `TO-FAULT` bzw. `TO-NORMAL` an.

Bitte beachten Sie: Der Property-Wert kann eine Auswahl von `DateTime` (meist verwendet), `Time` (selten verwendet) oder `Sequence Number` (falls das Gerät keine Uhr unterstützt) sein.

3.2.27 Event_Message_Texts

Diese Eigenschaft gibt den Meldungstext der letzten Ereignisbenachrichtigungen für die Ereignisse `TO-OFFNORMAL`, `TO-FAULT` bzw. `TO-NORMAL` an.

3.2.28 Profile_Name

Diese Eigenschaft ist der Name eines Objektprofils, dem dieses Objekt entspricht. Um die Eindeutigkeit zu gewährleisten, muss ein Profilname mit einem Vendor Identifier Code (415 für Beckhoff Automation) im Basis-10-Integer-Format beginnen, gefolgt von einem Bindestrich.

3.2.29 Event_Message_Texts_Config

Diese Eigenschaft enthält die Zeichenketten, die die Grundlage für den Parameter "Message Text" für die Ereignisbenachrichtigungen der Ereignisse `TO_OFFNORMAL`, `TO_FAULT` und `TO_NORMAL` bilden.

3.2.30 Event_Detection_Enable

Diese Eigenschaft gibt an, ob Intrinsic Reporting im Objekt aktiviert ist oder nicht, und steuert, ob das Objekt von den Ereigniszusammenfassungsservices berücksichtigt wird oder nicht.

3.2.31 Event_Algorithm_Inhibit_Ref

Diese Referenz gibt die Eigenschaft an, die den Wert der Eigenschaft `Event_Algorithm_Inhibit` steuert.

3.2.32 Event_Algorithm_Inhibit

Diese Eigenschaft zeigt an, ob der Ereignisalgorithmus für das Objekt deaktiviert wurde oder nicht.

3.2.33 TimeDelay_Normal

Diese Eigenschaft gibt die Mindestzeitspanne in Sekunden an, die `Present_Value` innerhalb des durch die Eigenschaften `High_Limit` und `Low_Limit` einschließlich der Eigenschaft `Deadband` definierten Bereichs bleiben muss, bevor ein `TO-NORMAL`-Ereignis generiert wird.

Bitte beachten Sie: Bis zur BACnet Revision 12 war im BACnet Standard nur `Time_Delay` für die beiden Übergänge `TO_OFFNORMAL` und `TO_NORMAL` verfügbar. Ab Revision 13 wird diese optionale Eigenschaft `TimeDelay_Normal` verwendet, um eine separate Hysterese für `TO_NORMAL`-Ereignisse festzulegen.

3.2.34 Reliability_Evaluation_Inhibit

Diese Eigenschaft gibt an, ob die Zuverlässigkeitsbeurteilung im Objekt deaktiviert ist oder nicht.

3.2.35 Property_List

Diese Eigenschaft ist ein BACnetARRAY von Eigenschaftenbezeichnern. Ein Eigenschaftenbezeichner für jede Eigenschaft, die innerhalb des Objekts existiert.

3.2.36 Priority_Array

Diese Eigenschaft ist ein schreibgeschütztes Array mit priorisierten Werten. Siehe Kapitel [Befehlspriorisierung](#) [▶ 63].

3.2.37 Relinquish_Default

Diese Eigenschaft ist der Standardwert, der für die Eigenschaft `Present_Value` verwendet wird, wenn alle Befehlsprioritätswerte in der Eigenschaft `Priority_Array` den Wert NULL haben. Siehe Kapitel [Befehlspriorisierung](#) [▶ 63].

3.2.38 Inactive_Text

(Binary-Objekte) Diese Eigenschaft stellt eine menschenlesbare Beschreibung des `INAKTIV`-Zustands dar. Wenn das Binary-Objekt beispielsweise einen Schalter darstellt, kann der inaktive Zustand als `AUS` dargestellt werden.

3.2.39 Active_Text

(Binary-Objekte) Diese Eigenschaft stellt eine menschenlesbare Beschreibung des `AKTIV`-Zustands dar. Wenn das Binary-Objekt zum Beispiel einen Schalter darstellt, kann der aktive Zustand als `EIN` dargestellt werden.

3.2.40 Change_Of_State_Time

(Binary-Objekte) Diese Eigenschaft gibt das Datum und die Uhrzeit an, zu der die letzte Zustandsänderung stattgefunden hat.

3.2.41 Change_Of_State_Count

(Binary-Objekte) Diese Eigenschaft gibt an, wie oft sich der Zustand der Eigenschaft `Present_Value` geändert hat, seit die Eigenschaft `Change_Of_State_Count` zuletzt auf einen Nullwert gesetzt wurde.

3.2.42 Time_Of_State_Count_Reset

(Binary-Objekte) Diese Eigenschaft stellt das Datum und die Uhrzeit dar, zu der die Eigenschaft `Change_Of_State_Count` zuletzt auf einen Nullwert gesetzt wurde.

3.2.43 Elapsed_Active_Time

(Binary-Objekte) Diese Eigenschaft stellt die kumulierte Anzahl der Sekunden dar, in denen `Present_Value` den Wert `ACTIVE` hatte, seit die Eigenschaft `Elapsed_Active_Time` zuletzt auf einen Nullwert gesetzt wurde.

3.2.44 Time_Of_Active_Time_Reset

(Binary-Objekte) Diese Eigenschaft stellt das Datum und die Uhrzeit dar, zu der die Eigenschaft `Elapsed_Active_Time` zuletzt auf einen Nullwert gesetzt wurde.

3.2.45 Alarm_Value

(Binary-Objekte) Diese Eigenschaft gibt den Wert an, den `Present_Value` haben muss, bevor ein Ereignis generiert wird.

3.2.46 Minimum_Off_Time

(Binary-Objekte) Diese Eigenschaft gibt die Mindestanzahl von Sekunden an, die `Present_Value` im Zustand `INACTIVE` verbleibt, nachdem ein Schreibvorgang in `Present_Value` diese Eigenschaft in den Zustand `INACTIVE` versetzt hat. Siehe Kapitel [Befehlspriorisierung](#) [▶ 63].

3.2.47 Minimum_On_Time

(Binary-Objekte) Diese Eigenschaft gibt die Mindestanzahl von Sekunden an, die `Present_Value` im Zustand `ACTIVE` verbleibt, nachdem ein Schreibvorgang in `Present_Value` diese Eigenschaft in den Zustand `ACTIVE` versetzt hat. Siehe Kapitel [Befehlspriorisierung](#) [▶ 63].

3.2.48 Feedback_Value

Diese Eigenschaft gibt den tatsächlichen Wert der von `Present_Value` kontrollierten Einheit an.

3.2.49 Number_Of_States

(Multistate-Objekte) Diese Eigenschaft definiert die Anzahl der Zustände, die `Present_Value` haben kann.

3.2.50 State_Text

(Multistate-Objekte) Diese Eigenschaft ist ein `BACnetARRAY` von Zeichenketten, die Beschreibungen aller möglichen Zustände von `Present_Value` darstellen. Die Anzahl der Beschreibungen entspricht der Anzahl der in der Eigenschaft `Number_Of_States` definierten Zustände.

3.2.51 Alarm_Values

(Multistate-Objekte) Diese Eigenschaft gibt alle Zustände an, denen `Present_Value` entsprechen muss, bevor ein `TO-OFFNORMAL`-Ereignis generiert wird.

3.2.52 Fault_Values

(Multistate-Objekte) Diese Eigenschaft gibt alle Zustände an, denen `Present_Value` entsprechen muss, bevor ein `TO-FAULT`-Ereignis generiert wird.

3.2.53 Date_List

(Calendar-Objekt) Diese Eigenschaft ist eine BACnetLIST von BACnetCalendarEntry, von denen jedes entweder ein bestimmtes Datum oder ein Datumsmuster (Date), ein Datumsbereich (BACnetDateRange) oder eine Monats-/Wochen-/Wochentagsangabe (BACnetWeekNDay) ist.

3.2.54 Weekly_Schedule

(Zeitplanobjekt) Diese Eigenschaft ist ein BACnetARRAY mit genau sieben Elementen. Jedes der Elemente 1-7 enthält einen BACnetDailySchedule. Ein BACnetDailySchedule besteht aus einer Liste von BACnetTimeValues, die die Abfolge von Zeitschaltplanaktionen an einem Wochentag beschreiben, an dem kein Exception_Schedule in Kraft ist. Montag ist das erste Element (Tag 1) und Sonntag ist das letzte Element (Tag 7).

3.2.55 Exception_Schedule

(Zeitplanobjekt) Diese Eigenschaft ist ein BACnetARRAY von BACnetSpecialEvents. Jedes BACnetSpecialEvent beschreibt eine Abfolge von Zeitschaltplanaktionen, die an einem bestimmten Tag oder an bestimmten Tagen Vorrang vor dem normalen Tagesverhalten haben. Das spezielle Ereignis kann ein Datum, einen Datumsbereich, eine WeekNDay-Information oder eine Calendar-Referenz (innerhalb desselben BACnet-Device) enthalten, um den Ausnahmezeitraum zu spezifizieren.

3.3 Zum Verständnis des PICS-Dokuments

Ein PICS (**P**rotocol **I**mplementation **C**onformance **S**tatement) ist das "BACnet-Datenblatt". Dieses Dokument beschreibt den Umfang der Implementierung von BACnet-Devices.

Das PICS-Dokument für Beckhoff-Steuerungen kann hier heruntergeladen werden: https://download.beckhoff.com/download/Document/certificates/beckhoff_bacnet_ip_pics_en_rev14_ver4.0.pdf

Das PICS beschreibt die unterstützten Objekttypen und für jeden Objekttyp, welche Eigenschaften unterstützt werden, den Lese- oder Schreibzugriff (sog. Konformitätscode) und mögliche Eigenschaftseinschränkungen.

Darüber hinaus werden auch die unterstützte Data Link Layer, Zeichensätze, Segmentierungsunterstützung usw. beschrieben.

Das dritte Element des PICS-Dokuments ist die Liste der unterstützten BIBBs (**B**ACnet **I**nteroperability **B**uilding **B**locks). Die Norm beschreibt A-seitige und B-seitige BIBBs. A-seitige BIBBs sind solche, die von einem Client ausgestellt werden, der etwas von einem B-seitigen Gerät (Server) oder an ein solches anfordert.

Um eine spezifische Funktionalität, wie z. B. Change-of-Value, zu nutzen, müssen beide Seiten, Client und Server, das entsprechende BIBB unterstützen.

Beispiel:

DS-COV-A: Client fordert Daten unter Verwendung des abonnierten Change-of-Value an.

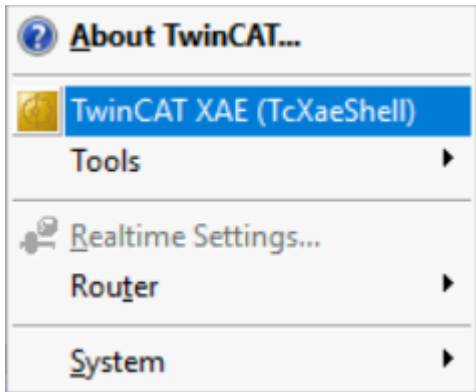
DS-COV-B: Der Server antwortet auf COV-Abonnements vom Client und stellt dem Client die Daten zur Verfügung, wenn sich der Wert geändert hat, indem er eine COV-Benachrichtigung sendet.

4 Quickstart

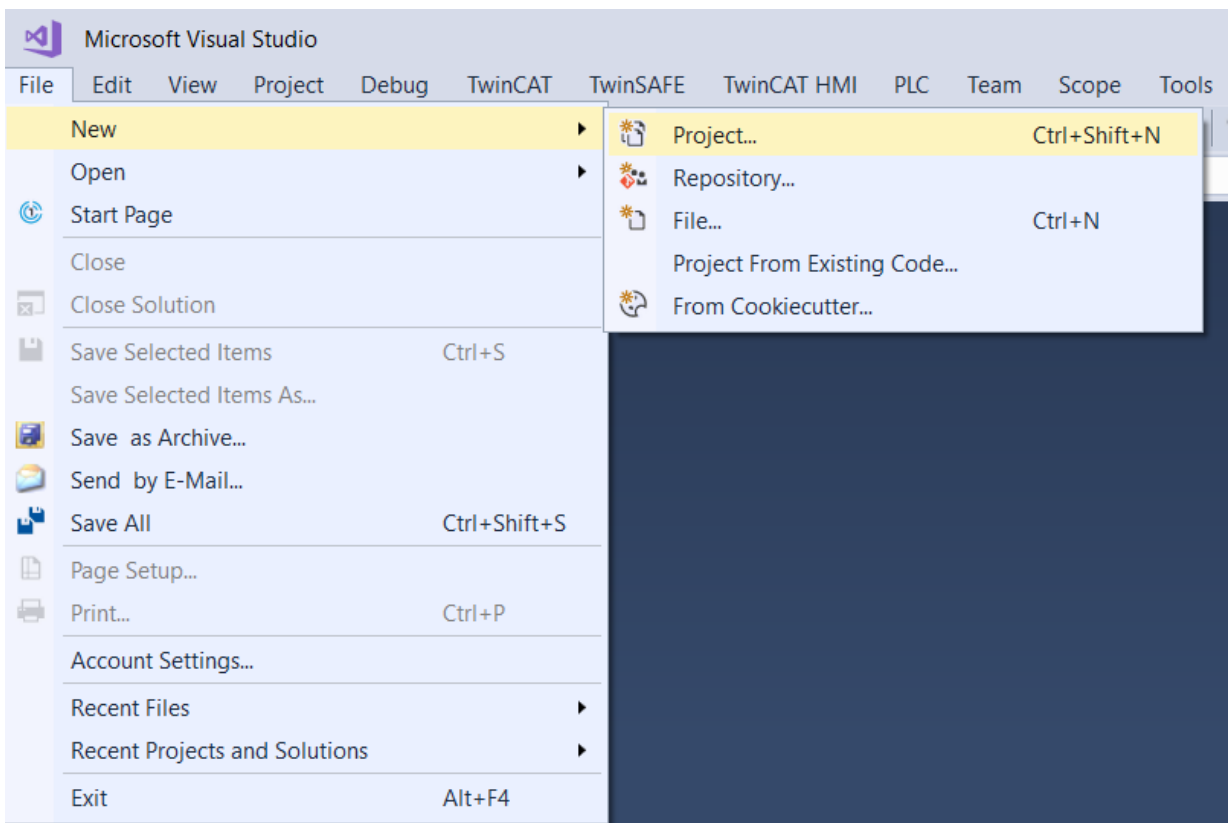
Dieses Beispiel zeigt, wie ein BACnet-Server mit einem einfachen Analog-Value-Objekt auf einer CX-Steuerung erstellt wird.

4.1 Anlegen des TwinCAT-Projekts

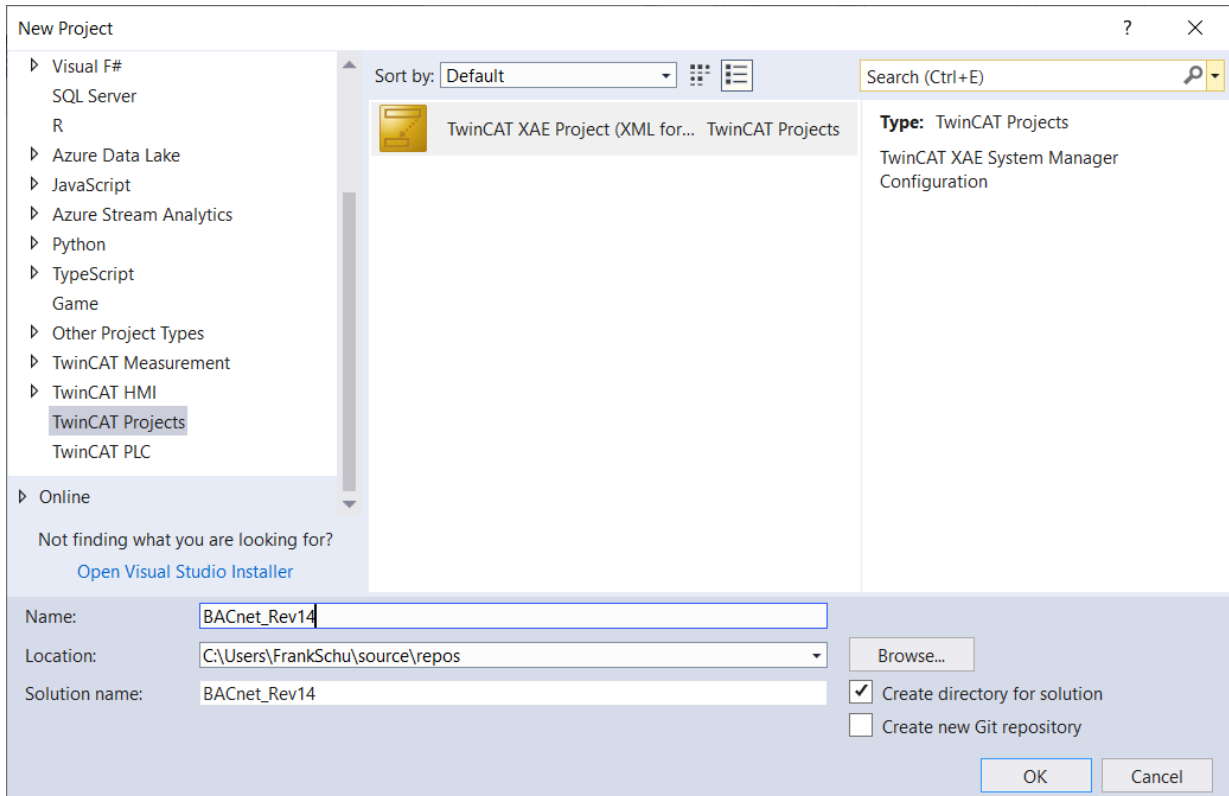
1. Starten Sie TwinCAT.



2. Legen Sie ein neues, leeres Projekt an.

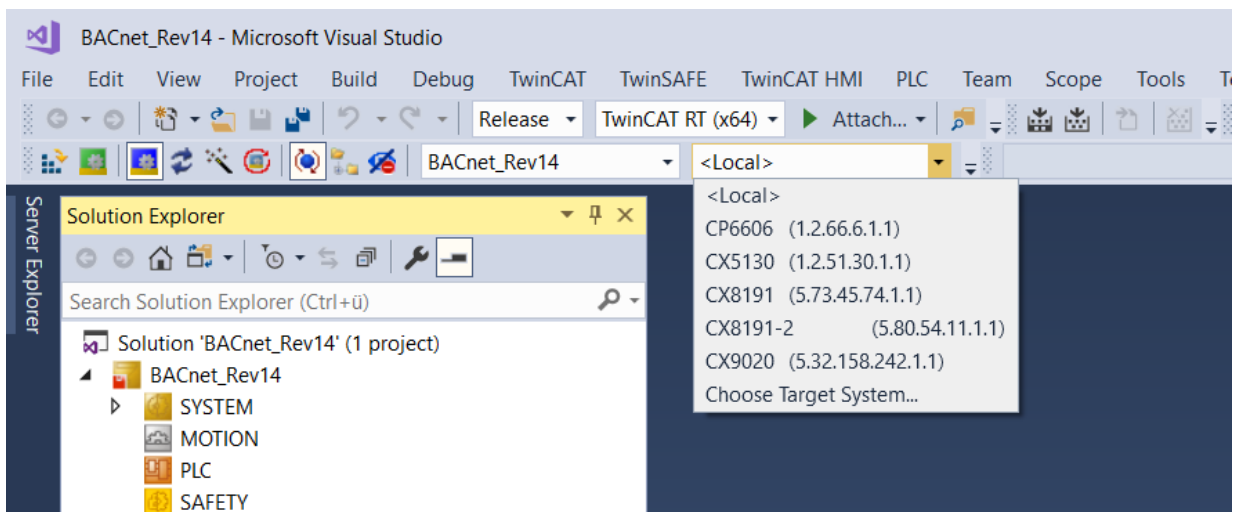


3. Wählen Sie *TwinCAT Projekte* in der Liste der Projekttypen, wählen Sie *TwinCAT XAE Project*, benennen Sie das Projekt und ändern Sie das Verzeichnis, falls erforderlich.



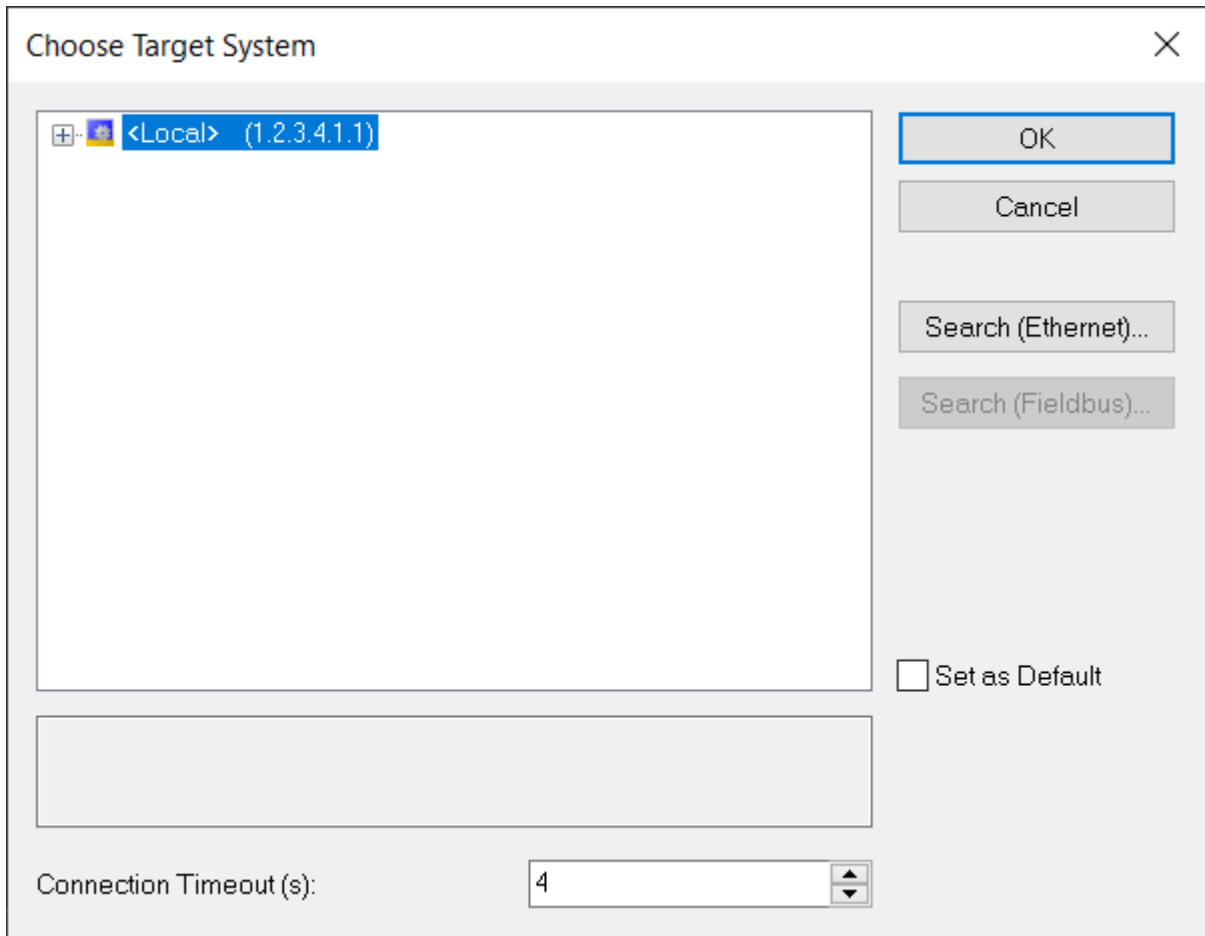
4.1.1 Auswahl des Zielsystems

1. Wählen Sie das Zielsystem aus der Liste der bereits über eine ADS-Route verbundenen Steuerungen aus.



2. Wenn keine ADS-Route zur Steuerung existiert, wählen Sie die Funktion *Auswahl des Zielsystems*.

3. Wählen Sie in diesem Dialog *Search (Ethernet)*.



4. Wählen Sie *Broadcast Search*.

5. Wählen Sie die Netzwerkadapter aus, die für die Suche nach Steuerungen verwendet werden sollen.

⇒ In dieser Liste werden alle verfügbaren Steuerungen im Netzwerk angezeigt. Der Buchstabe "x" in der Spalte *Connected* bedeutet, dass bereits eine ADS-Route eingerichtet wurde.

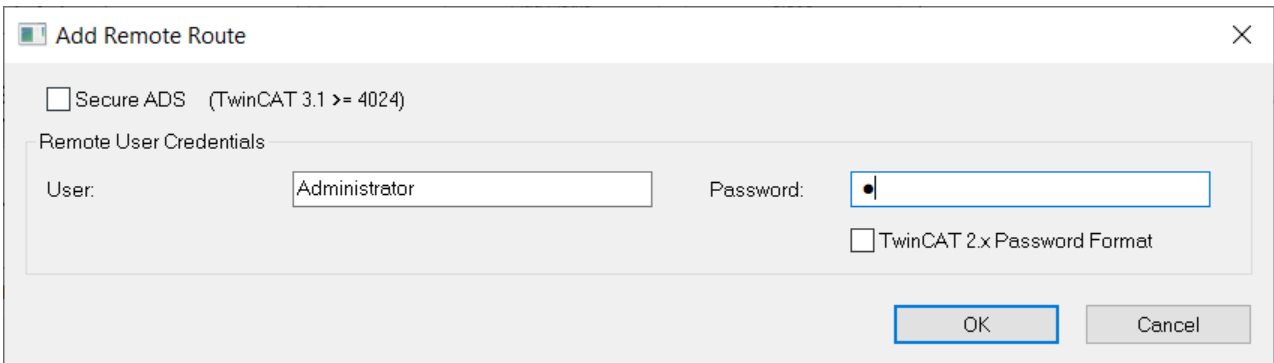
Host Name	Connected	Address	AMS NetId	TwinCAT	OS Version	Fingerprint
CX-0C07A2		192.168.2.16	192.168.2.16.1.1	2.11.2258	Win XP	
CX-16C2EC		192.168.2.222	192.168.2.222.1.1	2.11.2305	Windows 8	
CX8191	x	192.168.0.191	5.73.45.74.1.1	3.1.4024	Win CE (7.0)	C4CABBB9BE18FFF2CE429D4436EF81917DB41
CX8191-2	x	192.168.2.85	5.80.54.11.1.1	3.1.4024	Win CE (7.0)	69697C212BE8D587985648C61A0FAA716DCC4E
CX9020		192.168.0.190	5.32.158.242.1.1	3.1.4024	Win CE (7.0)	7760A88DD102DA583232FC4EA280395F560797

6. Wählen Sie die Steuerung, die für das BACnet-Projekt verwendet werden soll, wählen Sie *IP Address* im Abschnitt *Address Info* und wählen Sie *Add Route*.

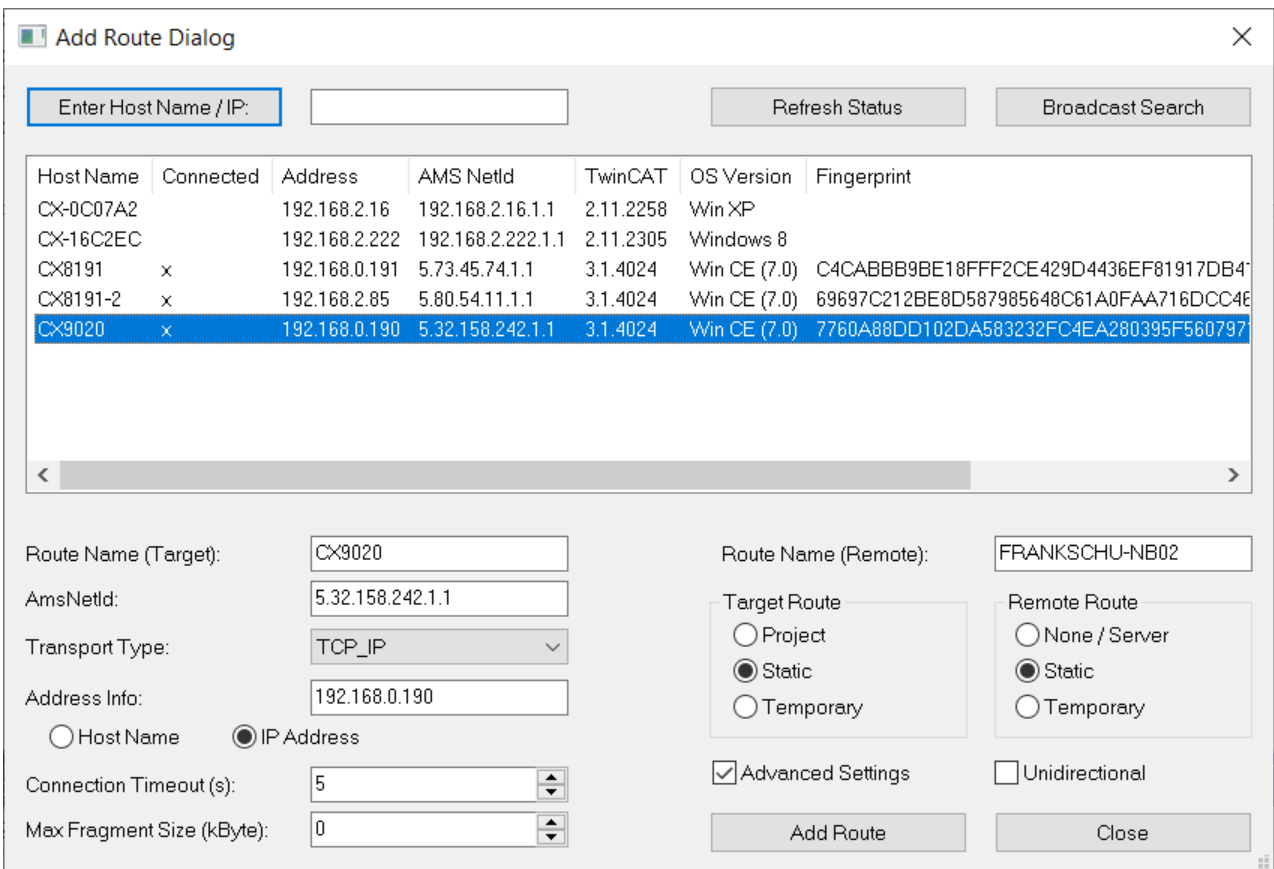


Es ist optional möglich, eine sichere ADS-Route zu erstellen. Weitere Informationen zur Verwendung von Secure ADS finden Sie im Beckhoff Information System.

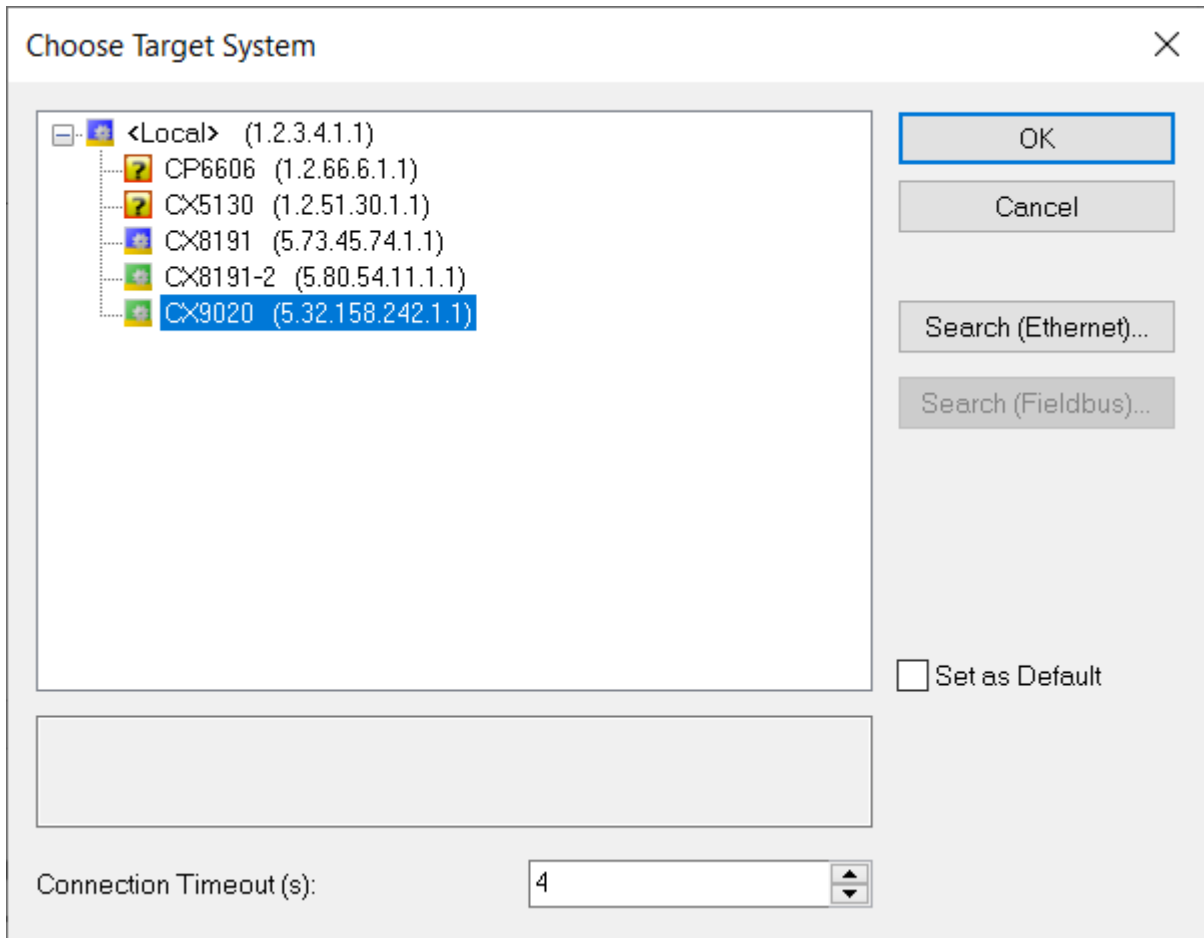
In diesem Beispiel wird eine reguläre ADS-Route verwendet. Das Standardpasswort für den Benutzer *Administrator* ist die Zahl *1*. Es wird dringend empfohlen, das Standardpasswort in ein sichereres Passwort zu ändern und in den Revisionsunterlagen des Projekts zu dokumentieren, spätestens bei Abschluss des Projekts.



Die Steuerung ist nun über ADS verbunden.

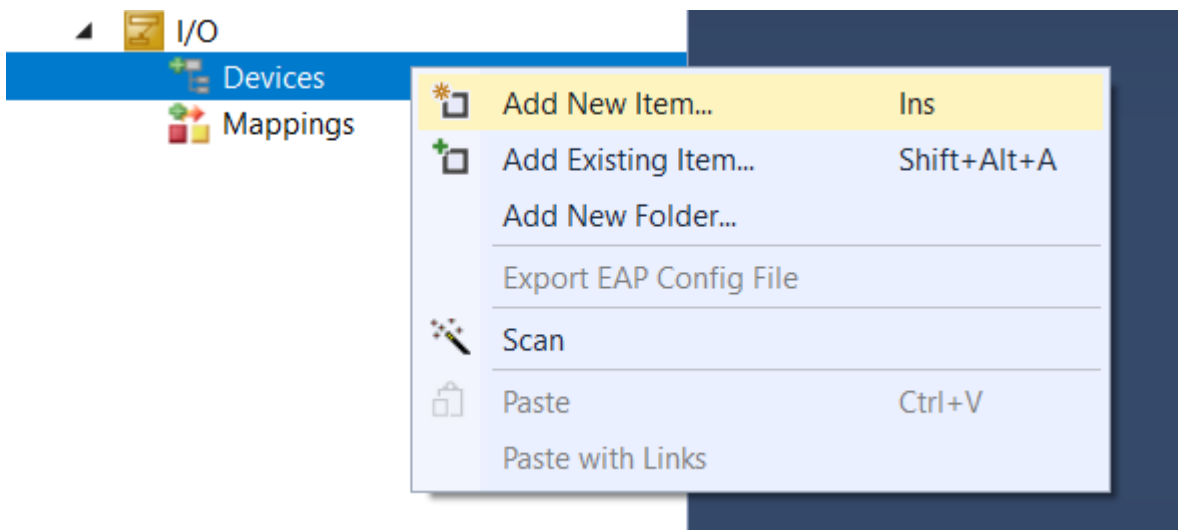


7. Schließen Sie den Dialog und wählen Sie die Steuerung aus der Liste der verfügbaren Zielsysteme aus.

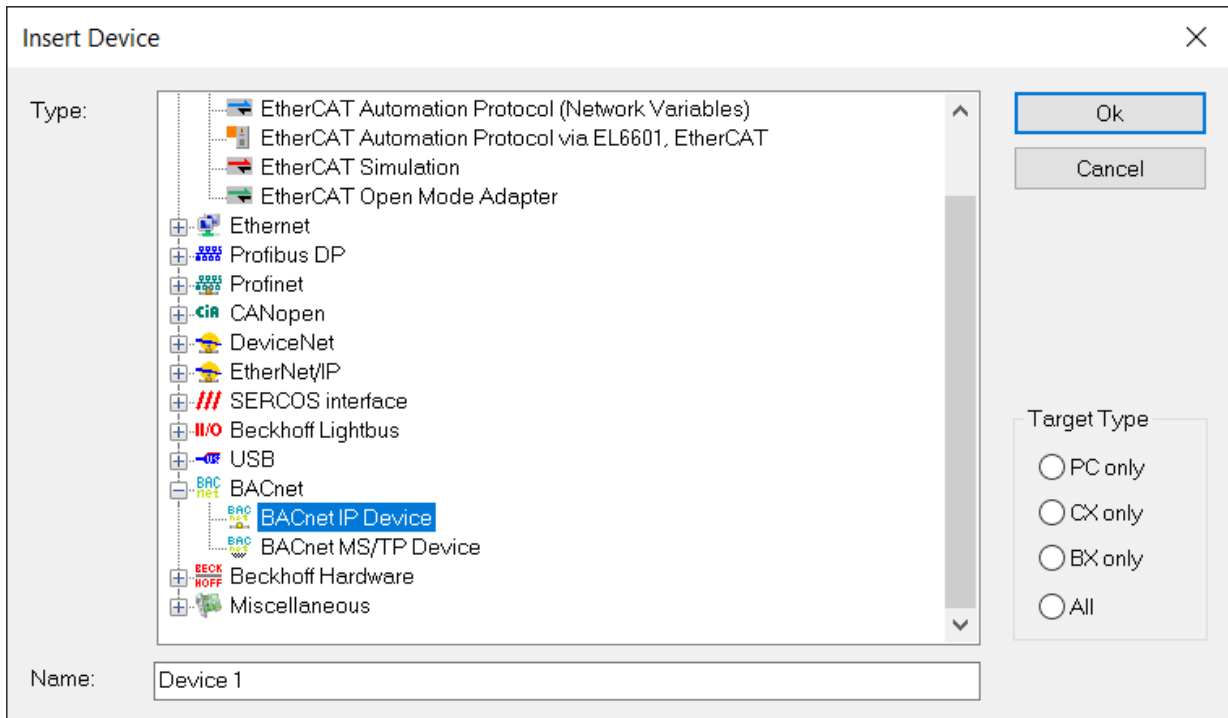


4.1.2 Anlegen eines BACnet-Adapters und -Servers

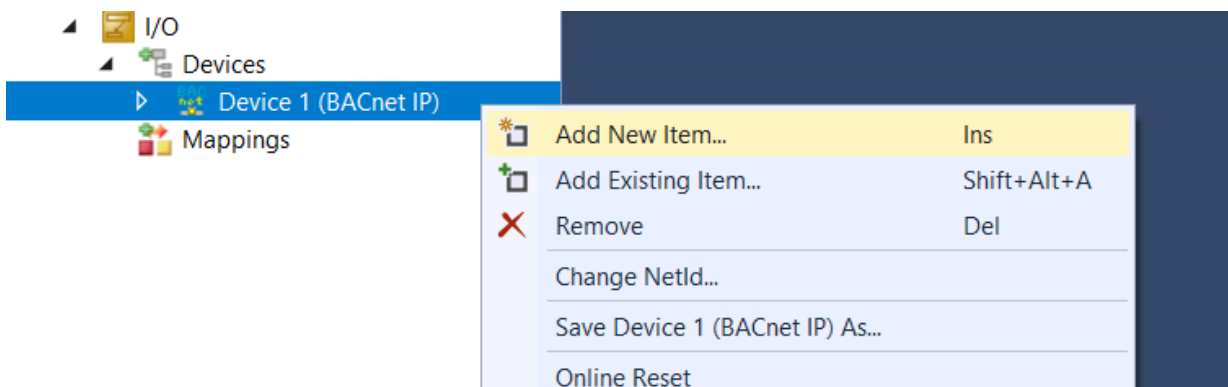
1. Klicken Sie mit der rechten Maustaste auf *Devices* und wählen Sie *Add New Item* aus.



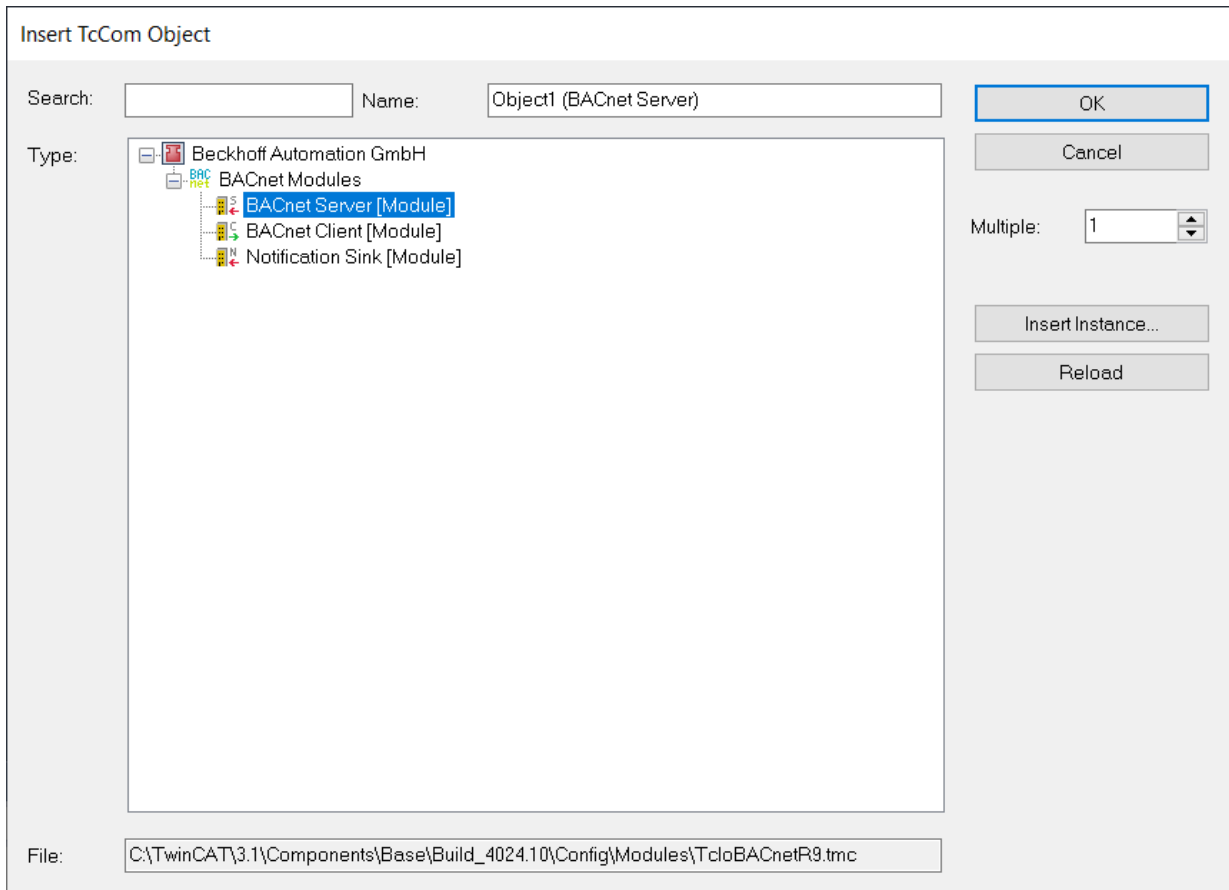
2. Wählen Sie *BACnet IP Device* aus der Liste der verfügbaren Supplements.



3. Klicken Sie mit der rechten Maustaste auf den Menüpunkt *Device 1 (BACnet IP)* und wählen Sie *Add New Item*.



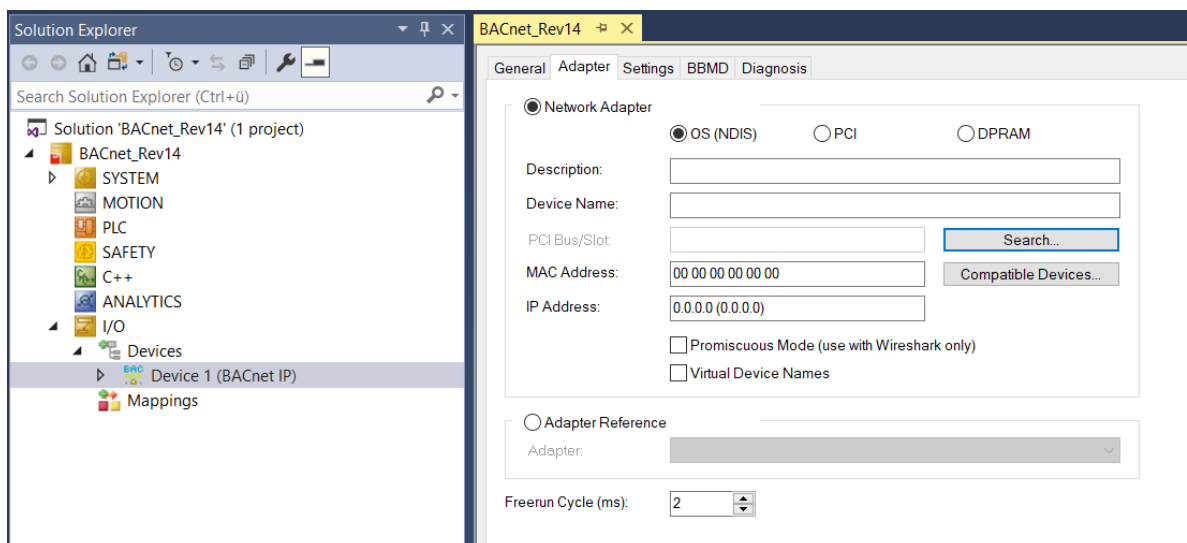
4. Wählen Sie aus der Liste der verfügbaren TCOM-Objekte *BACnet Server (Module)*.



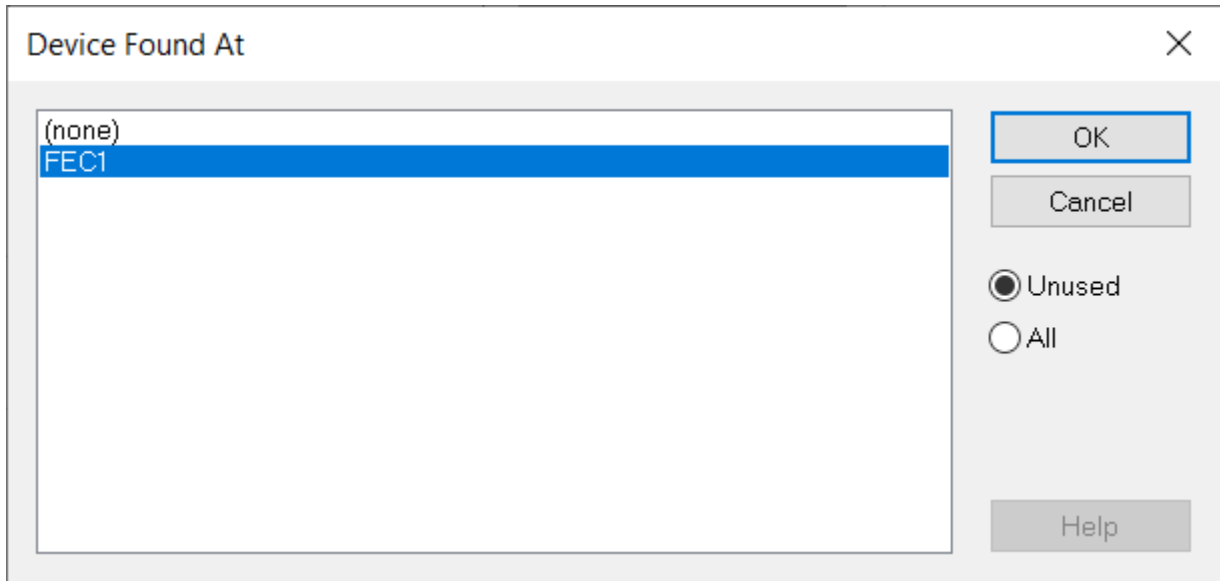
4.1.3 Einstellungen für IP-Adresse

1. Doppelklicken Sie auf den BACnet-Adapter *Device 1 (BACnet IP)*, wählen Sie die Registerkarte *Adapter* und überprüfen Sie, ob die IP-Adresse korrekt eingestellt wurde.

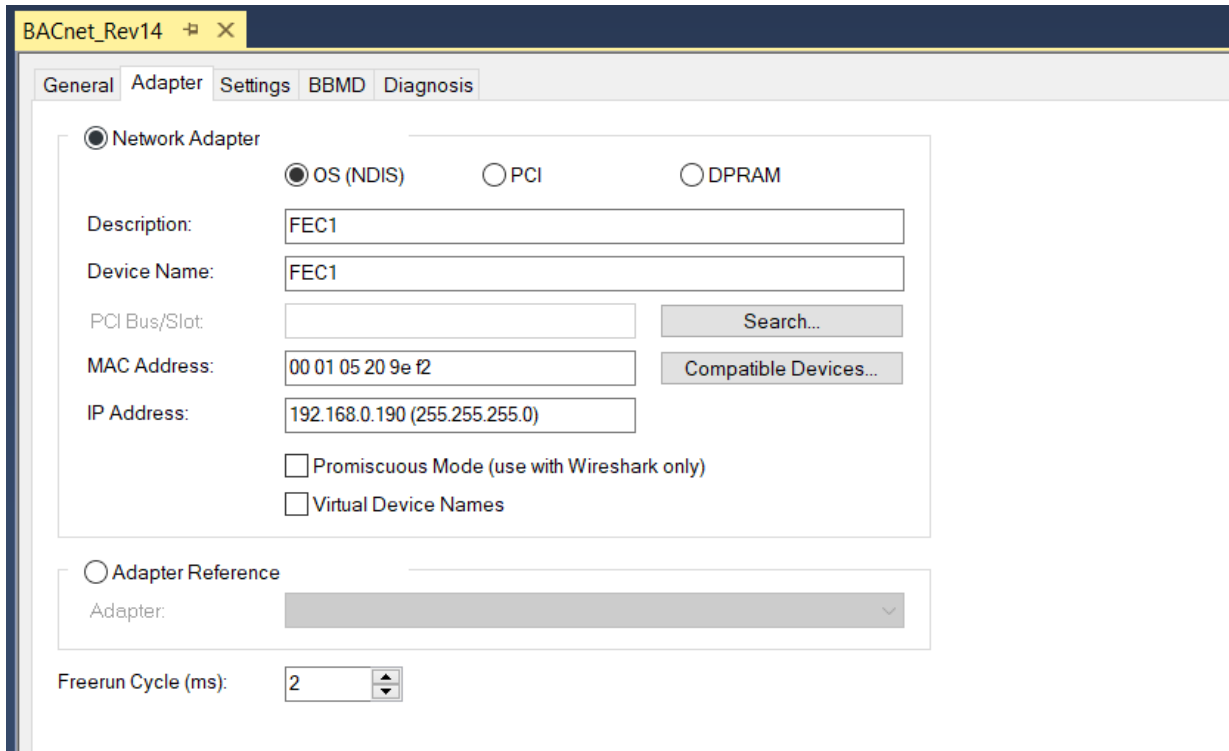
⇒ In der Abbildung unten sind die MAC-Adresse und die IP-Adresse auf Null gesetzt, es wurde kein Adapter ausgewählt.



2. Wählen Sie *Search* und wählen Sie die Netzwerkschnittstelle aus. Je nach Steuerungstyp können die Namen und die Anzahl der Netzwerkschnittstellen variieren.

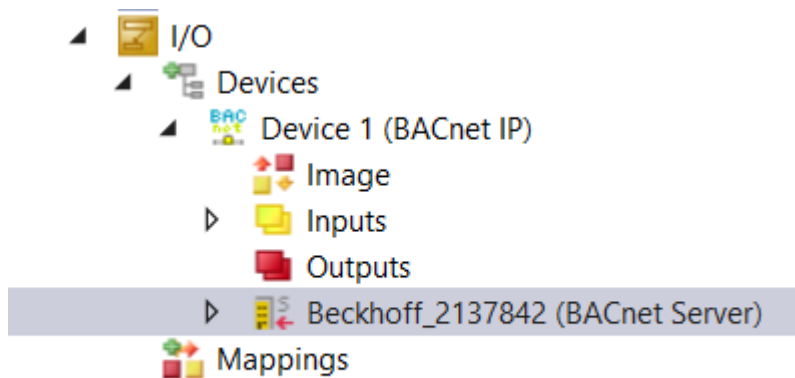


3. Prüfen Sie, ob die IP-Adresse richtig eingestellt wurde.



4.1.4 Anpassen der BACnet-Server-Einstellungen

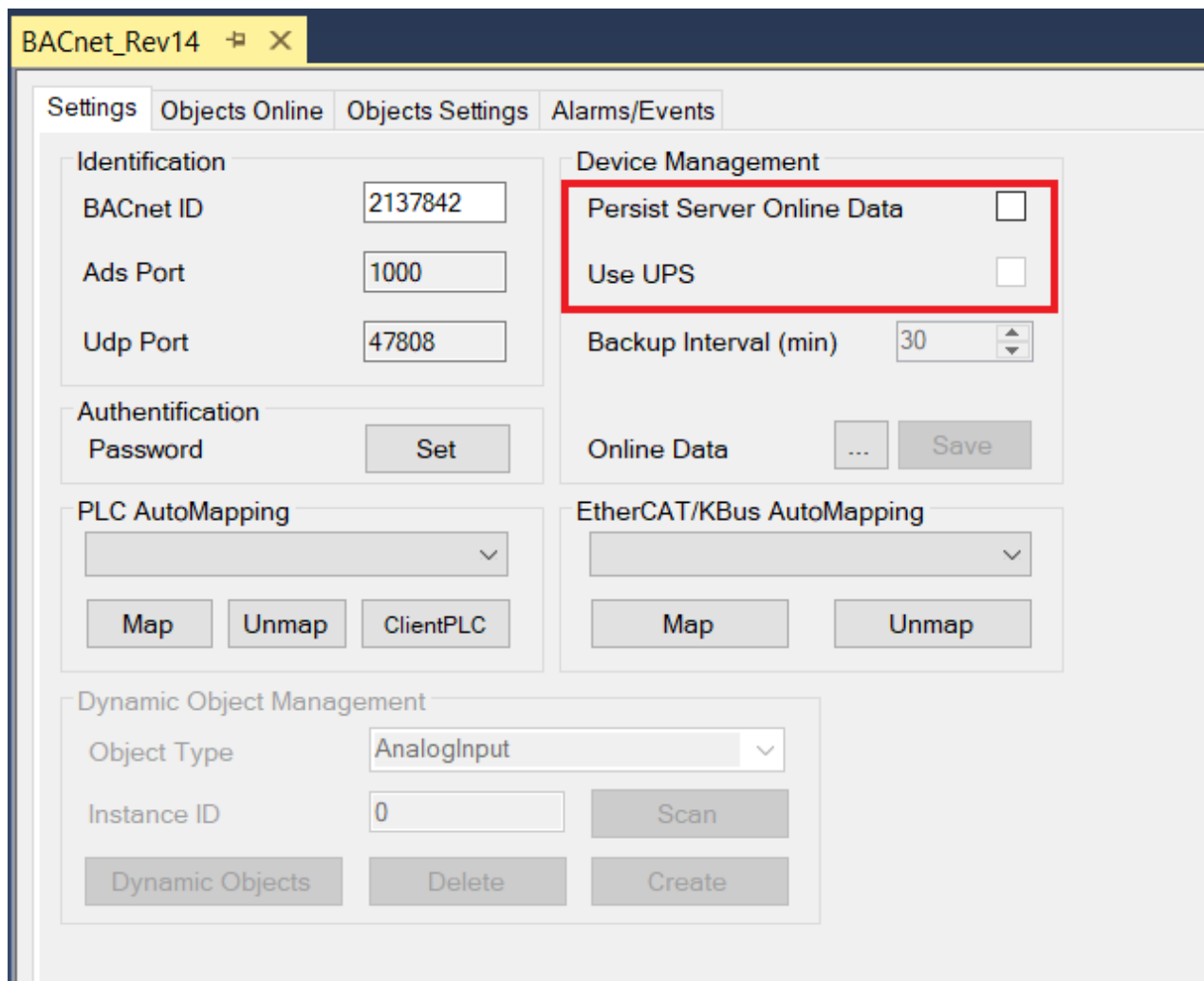
1. Doppelklicken Sie auf den BACnet-Server, der im Tree Item Device 1 angelegt wurde.



2. Verwenden Sie das Eingabefeld *BACnet ID*, um die Instanznummer des BACnet-Servers einzustellen.

i Die BACnet Instanznummer ist ein 22-Bit-Wert (gültiger Bereich von 0 - 4194302, 4194303 ist als Wildcard reserviert und kann nicht verwendet werden). In Kombination mit dem Objekttyp (10 Bit) ergibt diese Information die eindeutige Adresse des Geräts (den Device Object Identifier).

Jedes Gerät im Netz benötigt einen eindeutigen Device Object Identifier. Die Eindeutigkeit wird über das gesamte BACnet-Internetwork gefordert, d.h. über alle BACnet-Netzwerke, die durch BACnet-Router verbunden sind. Sprechen Sie mit dem Fachplaner oder anderen an der BACnet-Installation beteiligten Parteien, um eine doppelte Adressierung zu vermeiden!



4.1.5 Persistenz

Die persistente Speicherung der von BACnet-Clients (z. B. von der GLT) geänderten Werte erfolgt als Datei im TwinCAT-Boot-Verzeichnis (BACnetOnline_1010010.bootdata). Um die persistente Speicherung zu aktivieren, aktivieren Sie bitte "Persist Server Online Data" in den BACnet-Server-Einstellungen (wie oben gezeigt).

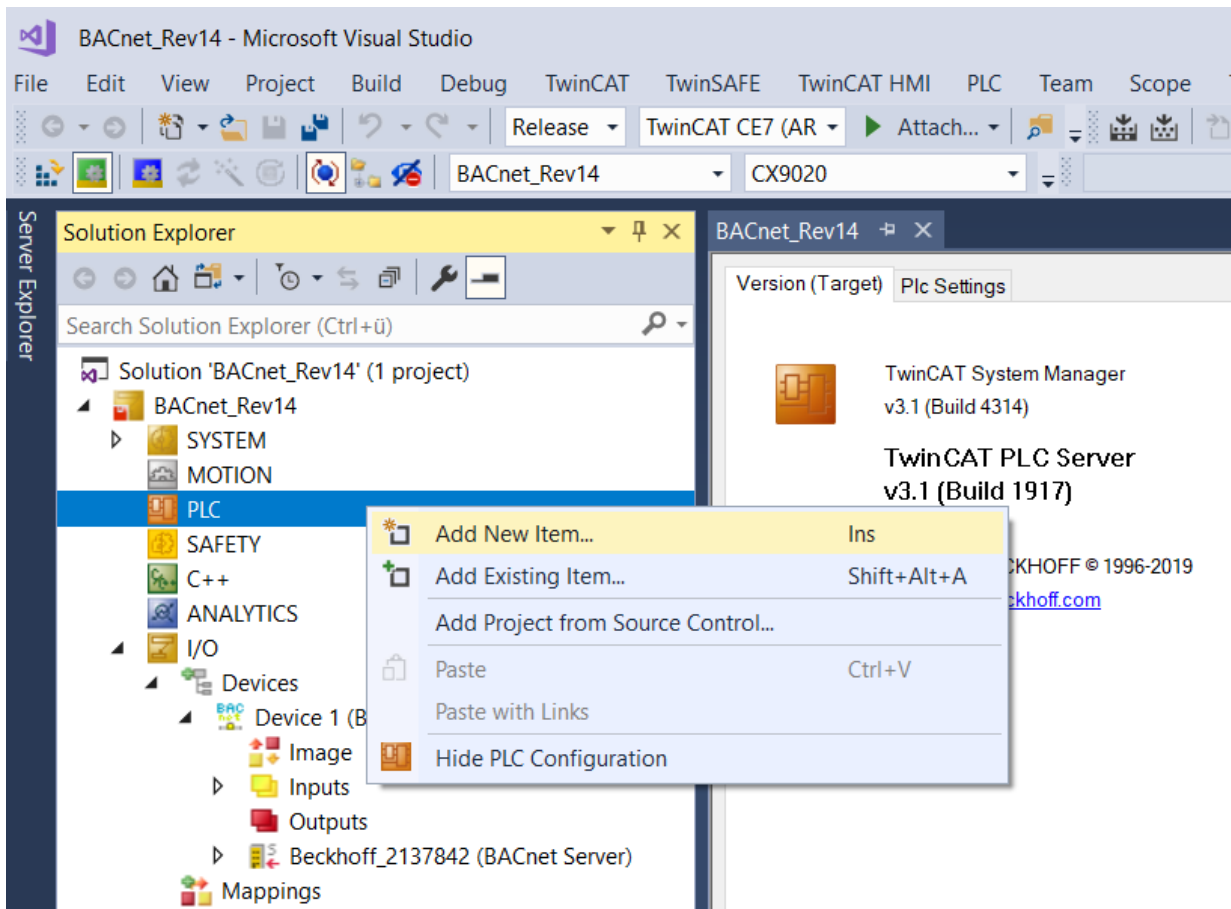
Falls eine USV in der Steuerung installiert ist, kann das Flag Use UPS aktiviert werden. In diesem Fall muss der USV-Funktionsbaustein dafür sorgen, dass bei einem Stromausfall persistente Daten geschrieben werden.

4.2 Anlegen des BACnet SPS-Projekts

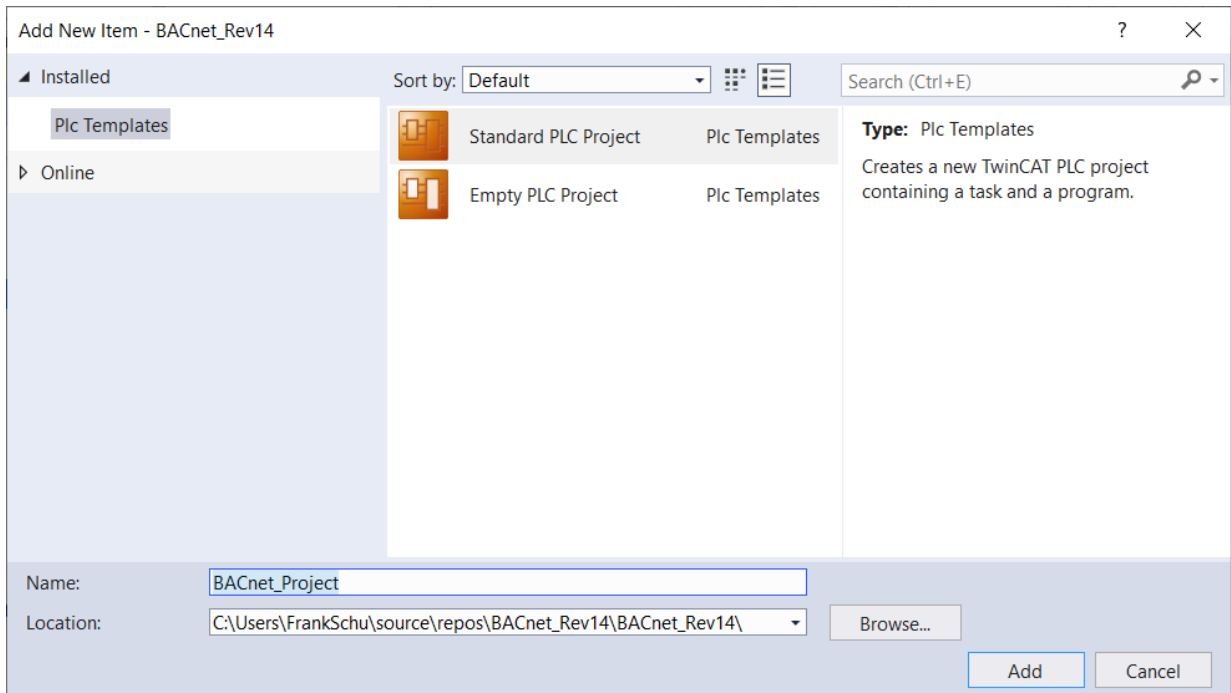
Dieses Beispiel zeigt, wie man ein erstes BACnet-Server-Projekt erstellt.

4.2.1 Anlegen des SPS-Projekts

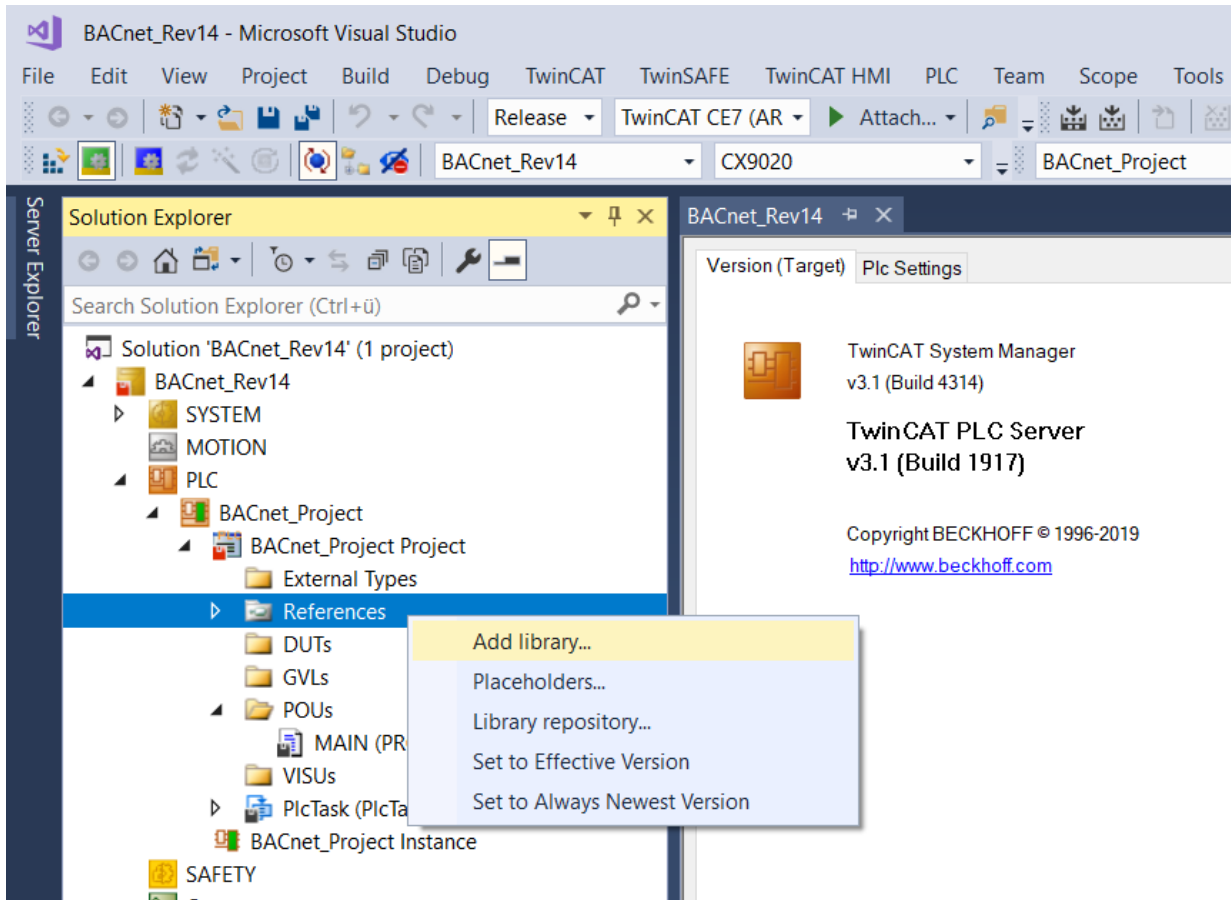
1. Klicken Sie mit der rechten Maustaste auf *PLC* und wählen Sie *Add New Item* aus.



2. Wählen Sie *Standard PLC Project* und benennen Sie das Projekt.



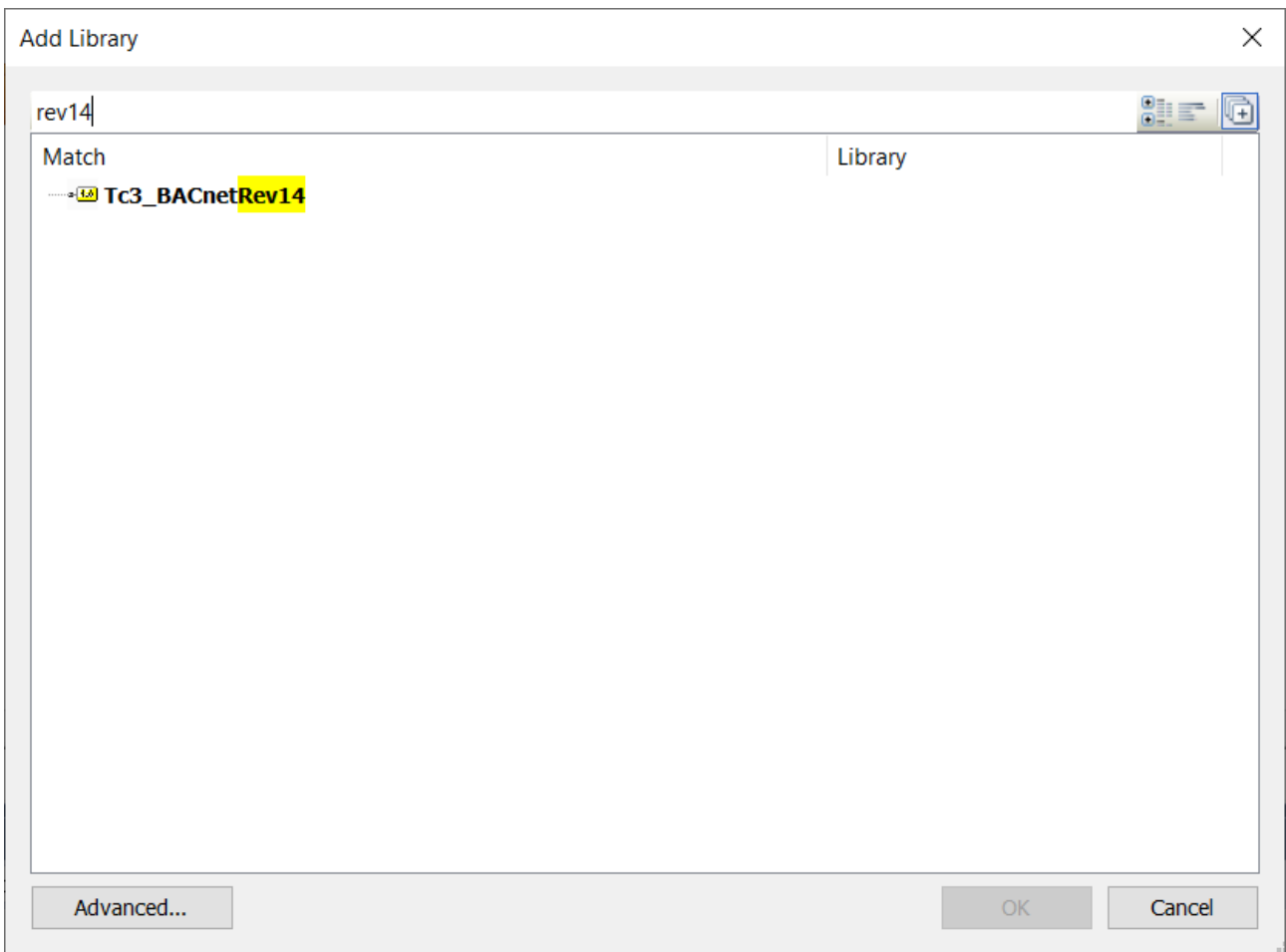
3. Klicken Sie mit der rechten Maustaste auf *References* und wählen Sie *Bibliothek hinzufügen*, um Bibliotheken zu einem SPS-Projekt hinzuzufügen.



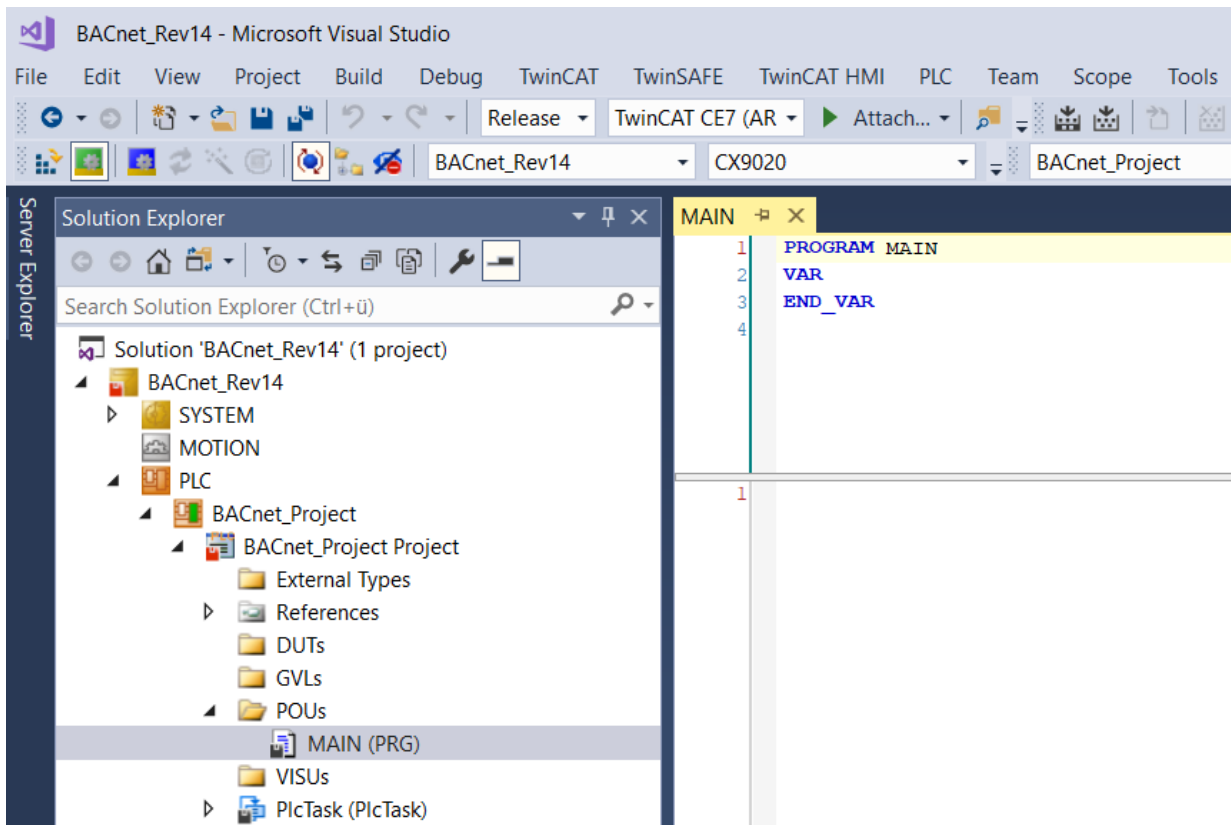
4. Wählen Sie die Bibliothek *Tc3_BACnetRev14*.



Wenn Sie den Text *rev14* in das Volltextsuchfeld eingeben, wird nur diese Bibliothek angezeigt. Markieren Sie diesen Punkt und wählen Sie *OK*.



5. Doppelklicken Sie im SPS-Projekt auf die POU MAIN.



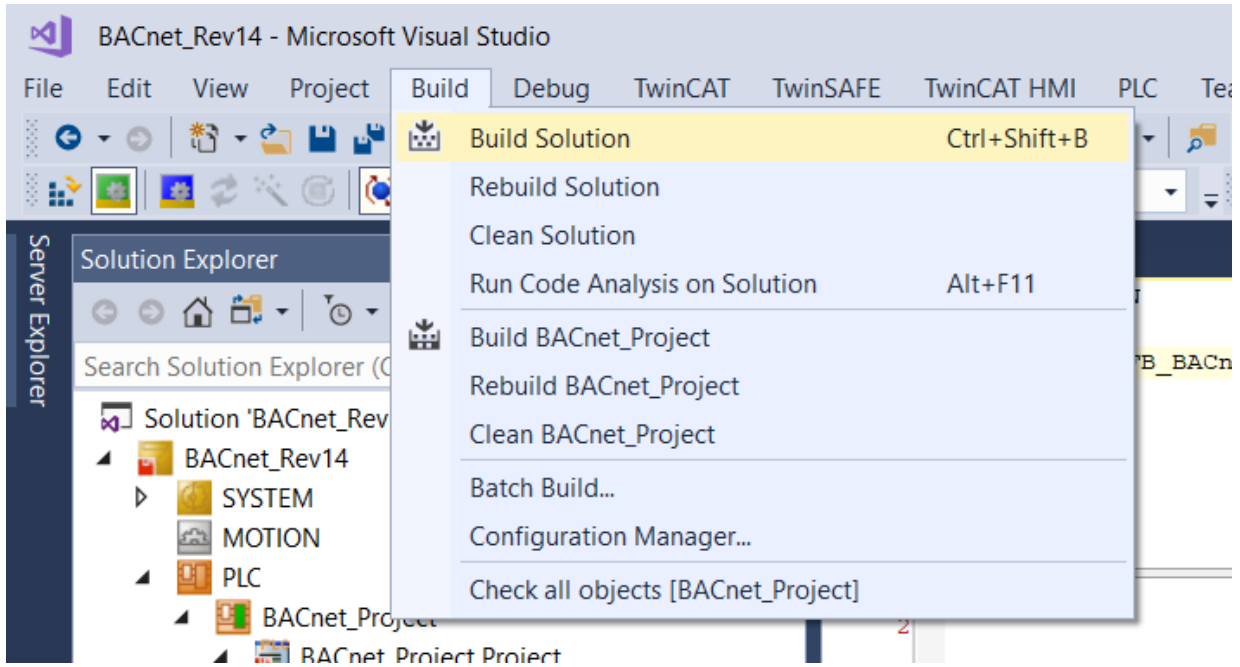
6. Fügen Sie die folgende Deklaration im Variablenfenster hinzu: *fbAv : FB_BACnet_AV,*

⇒ Dieser Befehl erzeugt eine Instanz des Funktionsbausteins FB_BACnet_AV mit dem Namen fbAv (der ein Analog-Value-Objekt darstellt).

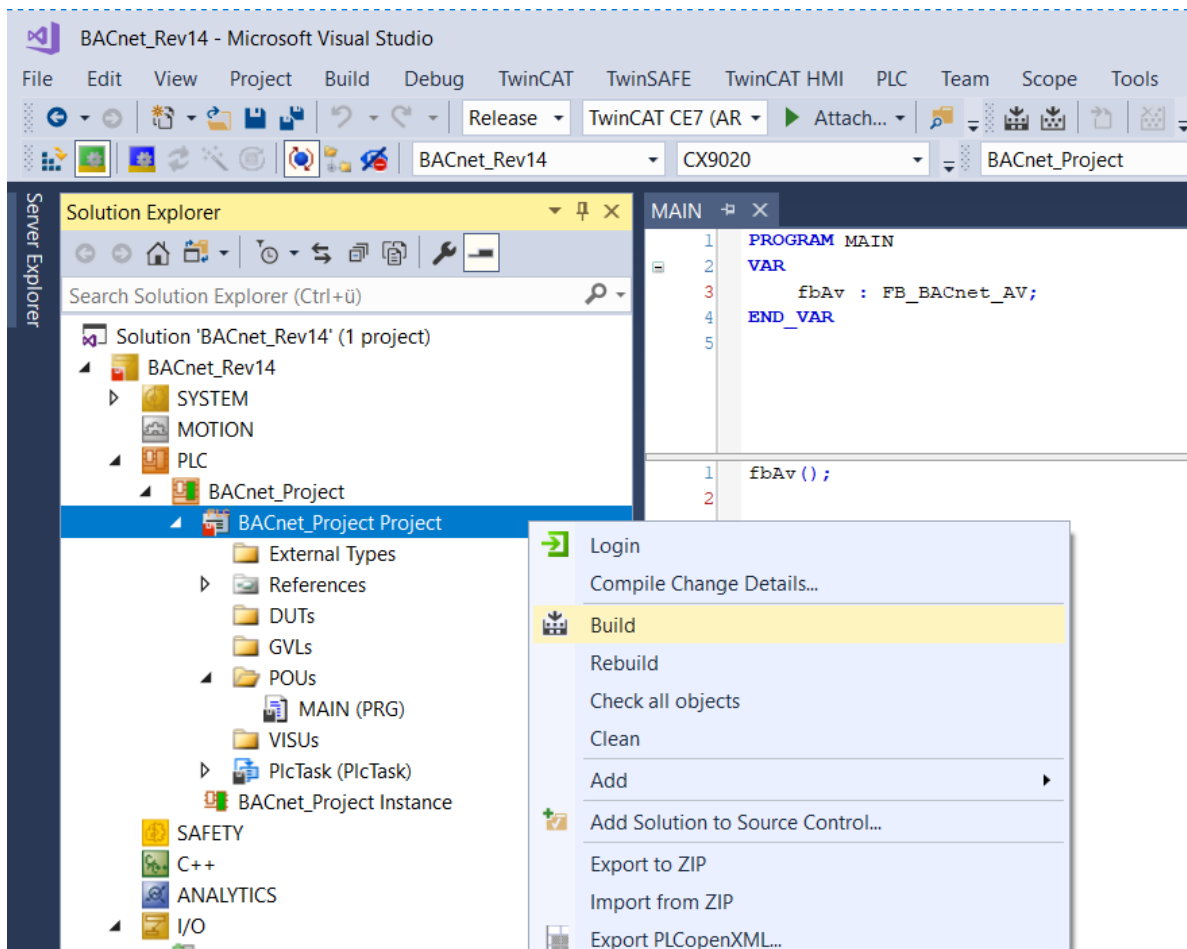
7. Fügen Sie dem Codefenster den folgenden Eintrag hinzu: `fbAv()`;

⇒ Dieser Code wird in regelmäßigen Abständen entsprechend der Zykluszeit (Standard: 10ms) aufgerufen.

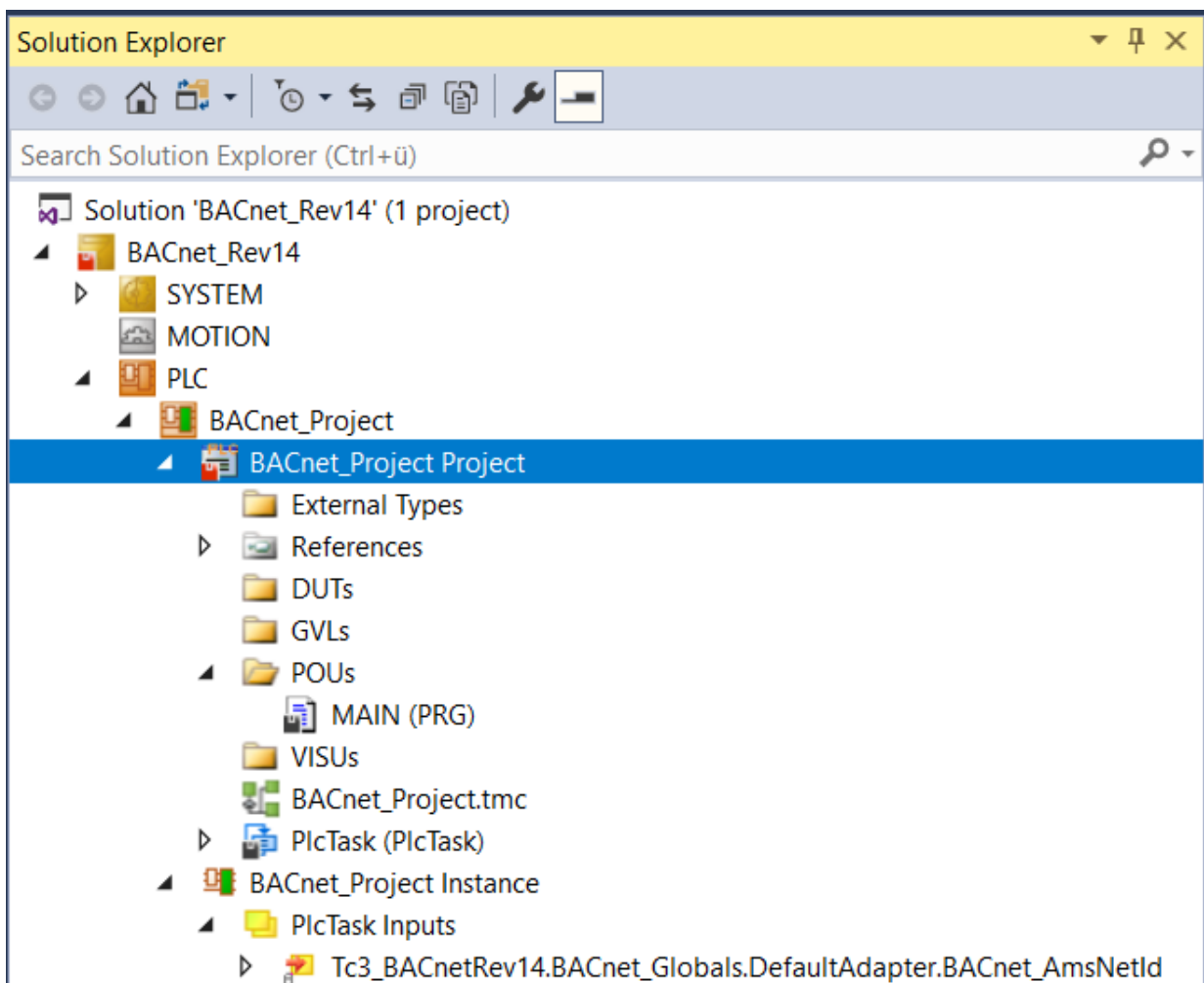
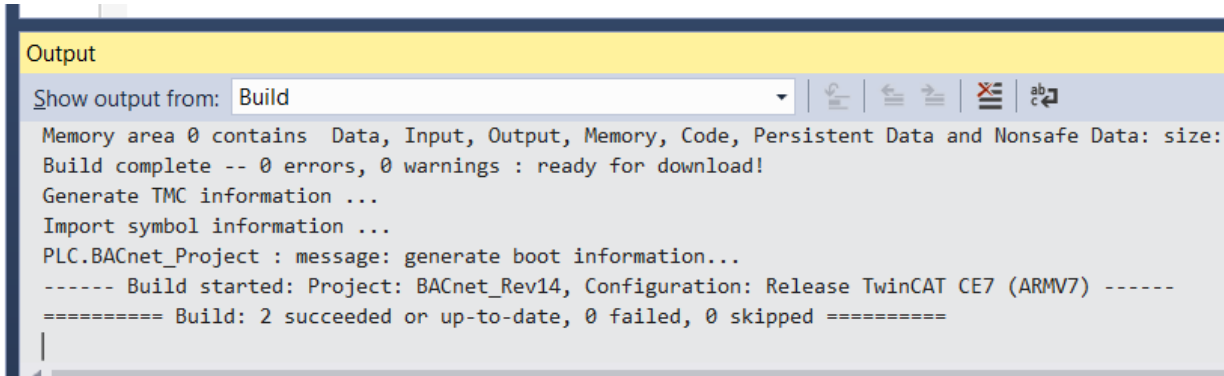
8. Übersetzen Sie das Projekt, indem Sie *Build / Build Solution* wählen.



⇒ Alternativ können Sie diese Funktion auch durch einen Rechtsklick auf den Projektnamen aufrufen.

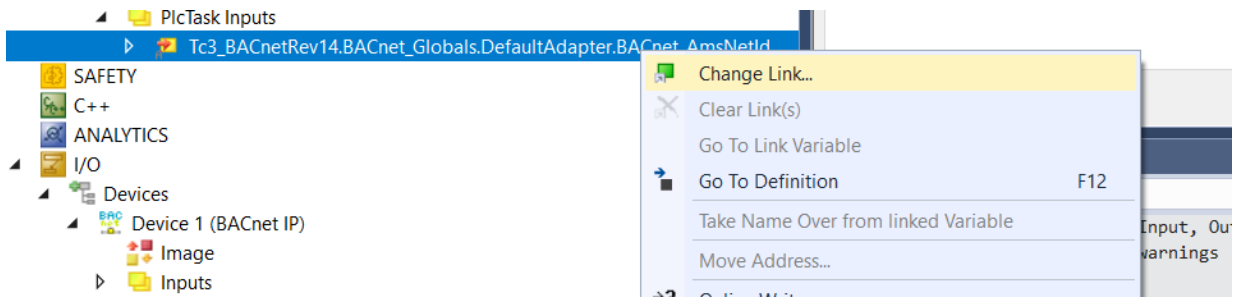


9. Prüfen Sie, ob der Build erfolgreich war und kein Fehler angezeigt wird. Wenn Fehler angezeigt werden, überprüfen Sie das Projekt und korrigieren Sie die Fehler.
10. Nach erfolgreicher Kompilierung wird im Prozessabbild die Variable `Tc3_BACnetRev14.BACnet_Globals.DefaultAdapter.BACnet_AmsNetId` angezeigt.



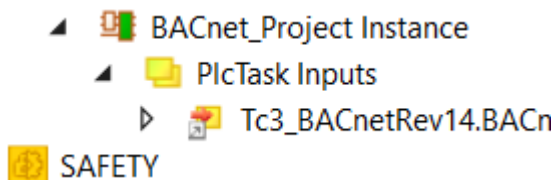
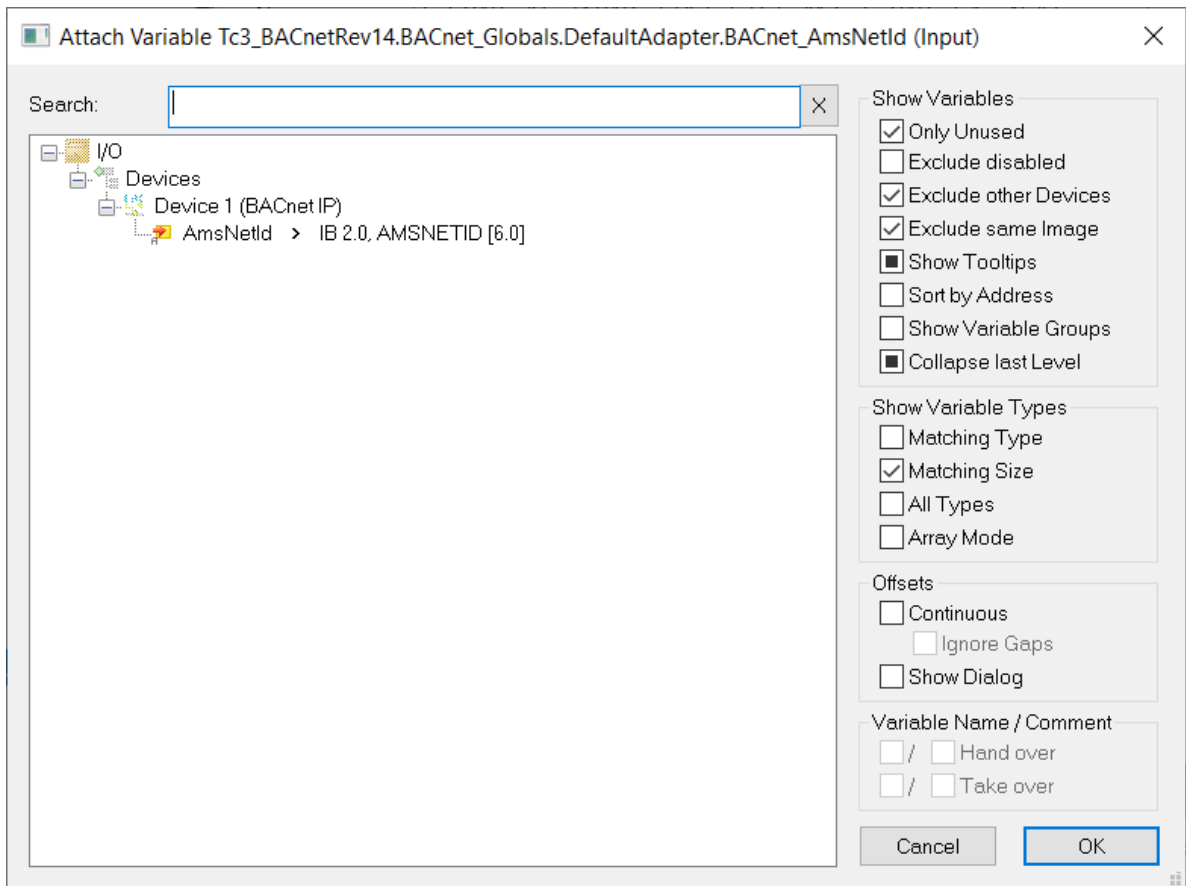
⇒ Diese Variable wird verwendet, um die BACnet-Verbindung zwischen dem SPS-Programm und dem BACnet-Adapter herzustellen.

11. Klicken Sie mit der rechten Maustaste auf diese Variable und wählen Sie *Change Link*.



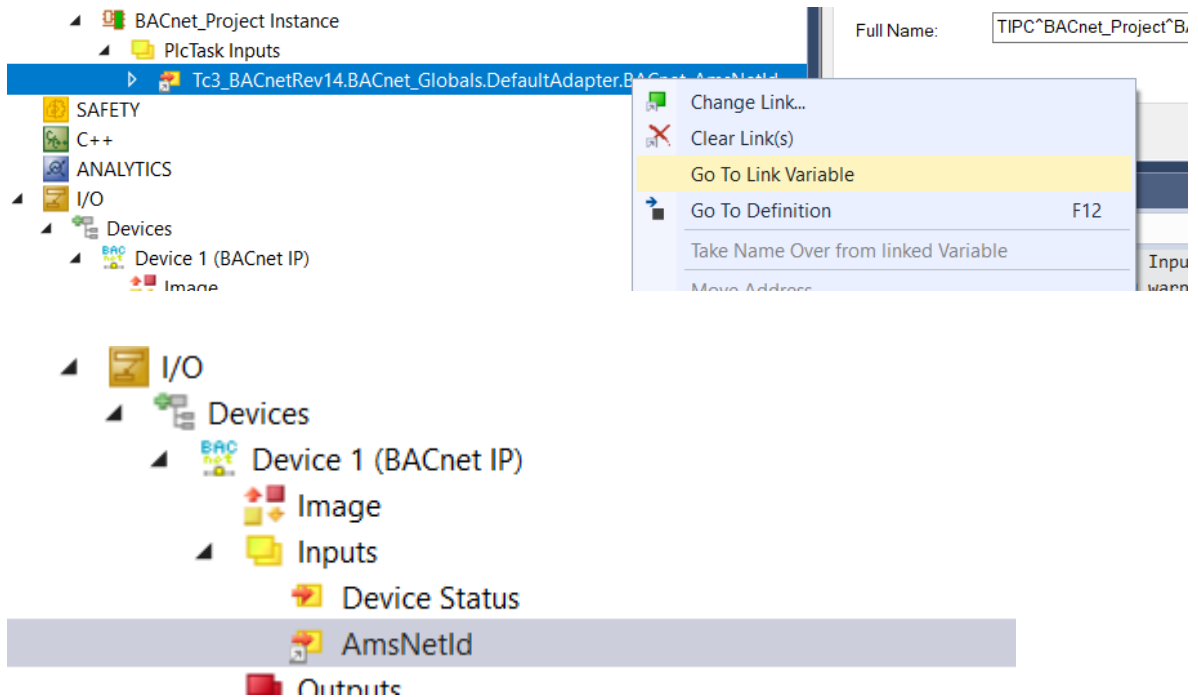
12. Wählen Sie die Variable desselben Typs im BACnet-Adapter aus und erstellen Sie mit OK eine Verknüpfung.

⇒ Die Verbindung wird durch ein Verbindungssymbol angezeigt.

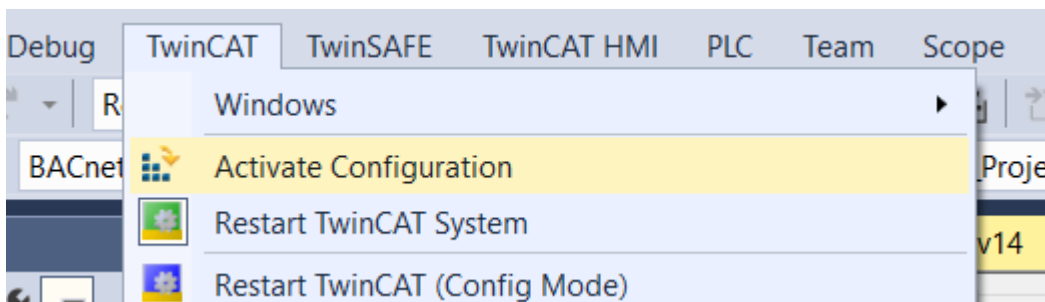


13. Sie können die Verbindung jederzeit mit der Funktion *Go To Link Variable* überprüfen.

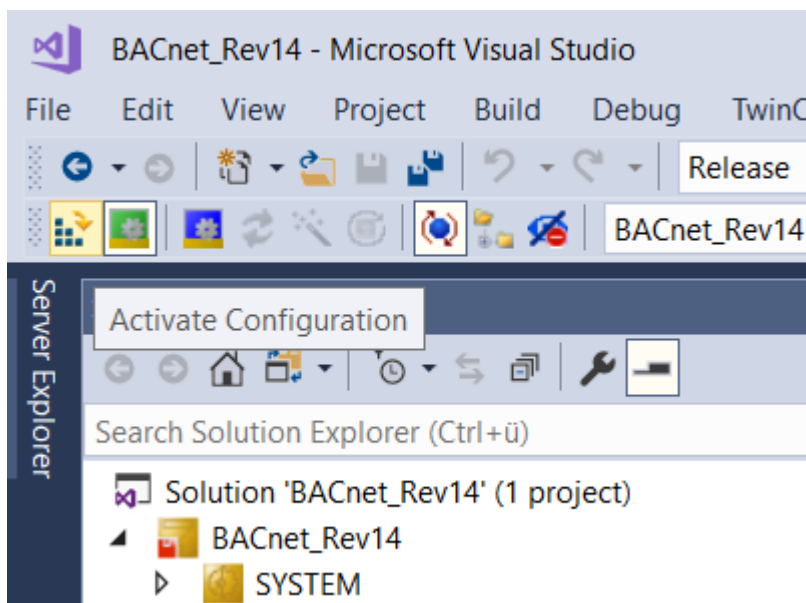
⇒ Mit dieser Funktion wird der Cursor auf die verbundene Variable gesetzt.



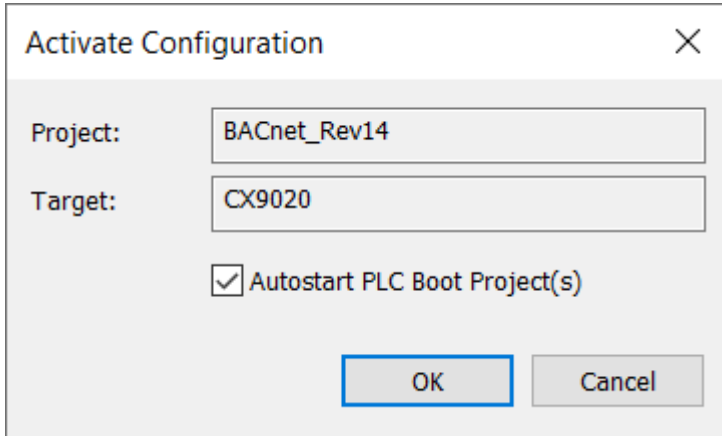
14. Aktivieren Sie das Projekt mit der Funktion *Konfiguration aktivieren* im TwinCAT-Menü.



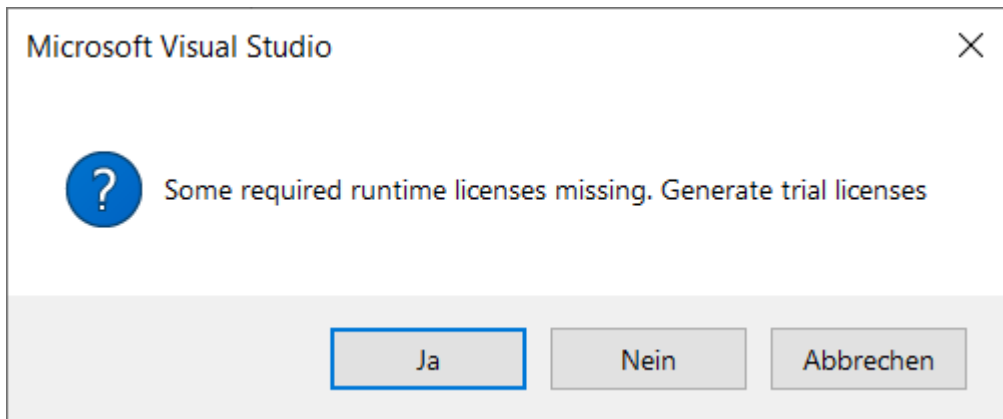
⇒ Alternativ ist diese Funktion auch in der TwinCAT-Symboleiste verfügbar.



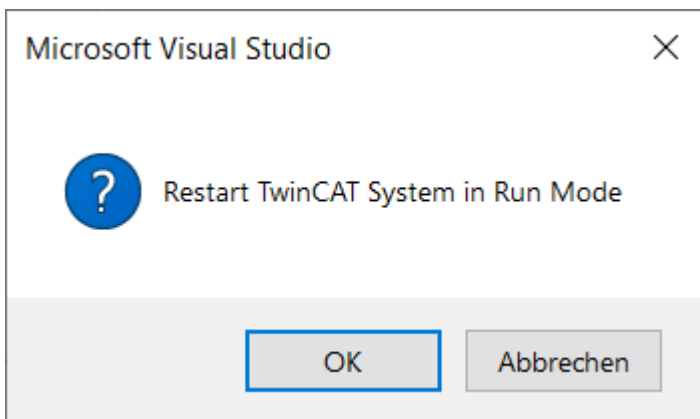
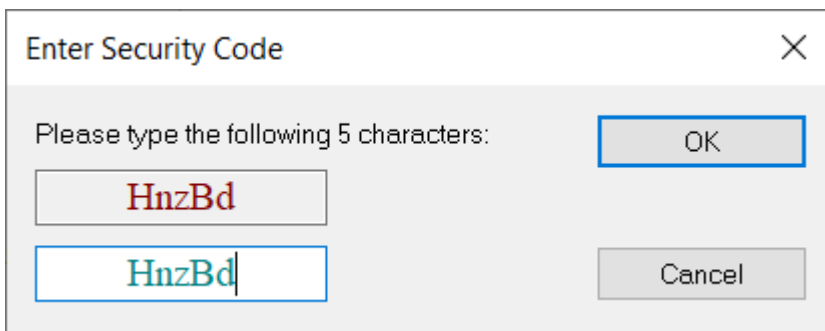
15. Prüfen Sie, ob das richtige Zielsystem ausgewählt ist, und wählen Sie OK.



⇒ Wenn auf dem Zielsystem keine Lizenzen vorhanden sind, können sogenannte Testlizenzen generiert werden. Testlizenzen sind für einen Zeitraum von 7 Tagen gültig. Das Beckhoff Information System bietet weitere Informationen zur TwinCAT-Lizenzierung.



16. Geben Sie die 5 Buchstaben aus dem oberen Feld in das untere Feld ein.
Bei der Eingabe wird zwischen Groß- und Kleinschreibung unterschieden.

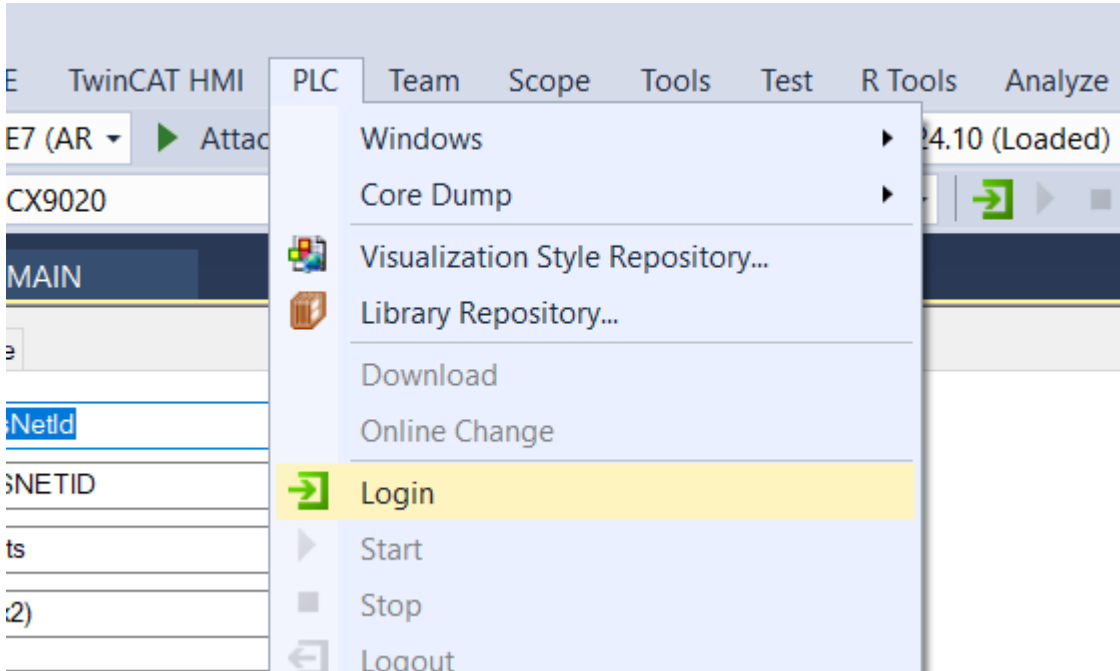


17. Um diesen Vorgang abzuschließen, fordert TwinCAT einen Neustart der SPS an. Wählen Sie OK.

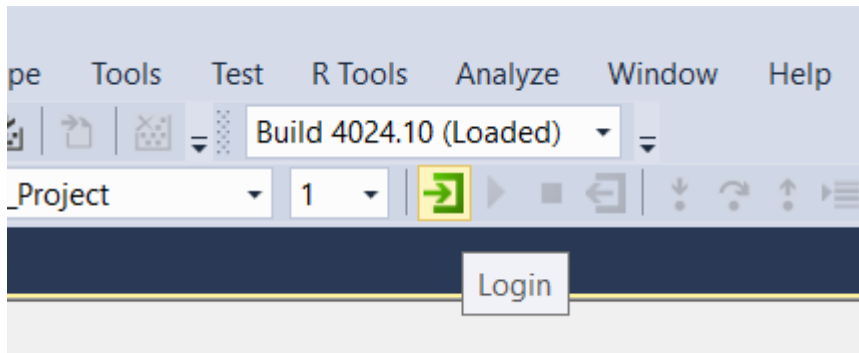
- ⇒ Mit diesen wenigen und einfachen Schritten wurde ein voll funktionsfähiger BACnet-Server erstellt. Der Server und die enthaltenen Objekte sind nun im BACnet-Netzwerk verfügbar.

4.2.2 Testen des BACnet-Servers

1. Melden Sie sich in der SPS an. Wählen Sie *Einloggen* aus dem PLC-Menü.



- ⇒ Alternativ ist diese Funktion auch über die TwinCAT-Symboleiste verfügbar.



2. Navigieren Sie zur POU MAIN, erweitern Sie die Anzeige der Variable *fbAv* mit dem Pluszeichen und überprüfen Sie den Inhalt dieses Funktionsbausteins, der das BACnet-Objekt darstellt.

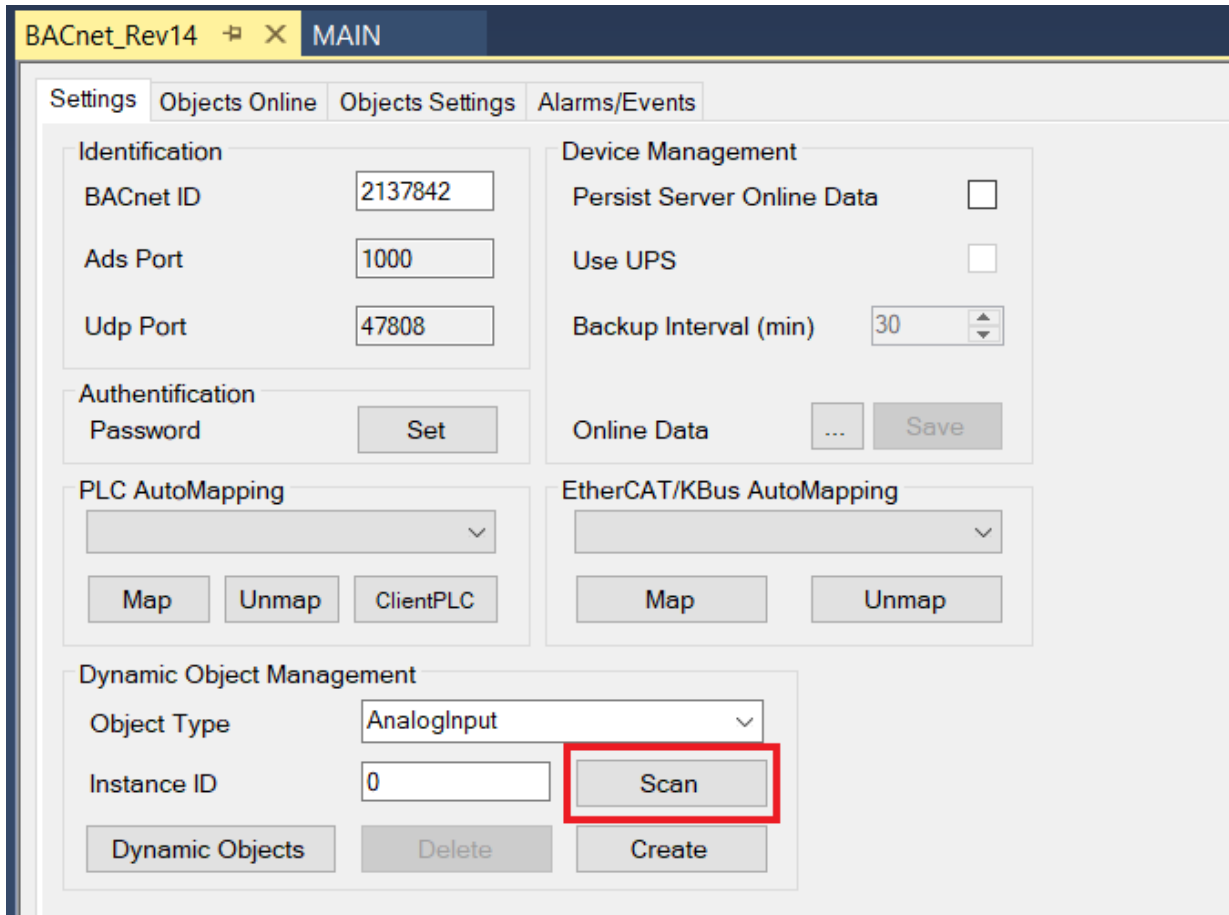
Expression	Type	Value	Prepared
fbAv	FB_BACnet_AV		
Server	REFERENCE TO...		
nObjectInstance	UDINT	10000	
bReady	BOOL	TRUE	
eObjectType	E_BACNETOBJT...	ObjAnalogVa...	
tState	TCOM_STATE	TCOM_STAT...	
eErrorID	E_BACNET_ER...	eNoError	
tObjectID	T_BACnet_Obj...	8398608	
bUpdateCOV	BOOL	FALSE	
nCycPropCnt	DINT	2	
aSyncHandlers	ARRAY [E_BA...		
nPlcSync	DINT	2	
ipSrv	ITcIoBACnetPlc...	16#D84F138C	
ipObj	ITcIoBACnetPlc...	16#D65374B8	
aReqSyncHandlers	ARRAY [E_BA...		
sObjectName	T_MaxString	'fbAv'	
sDescription	T_MaxString	''	
iParent	I_BACnet_View	16#00000000	
bObjNameChanged	BOOL	FALSE	
bDescriptionChanged	BOOL	FALSE	
sSymbolPath	T_MaxString	'BACnet_Proj...	
sSymbolName	STRING	'fbAv'	
iPrevSibling	I_BACnet_Object	16#00000000	
bEventDetectionEnable	BOOL	TRUE	
bEventAlgorithmInhibit	BOOL	FALSE	

- ⇒ Die Variable *bReady* kennzeichnet einen erfolgreichen Start dieses Objekts. In diesem Beispiel wurde keine spezifische Konfiguration der Objekteigenschaften vorgenommen, so dass die Property-Werte Standardwerte sind.

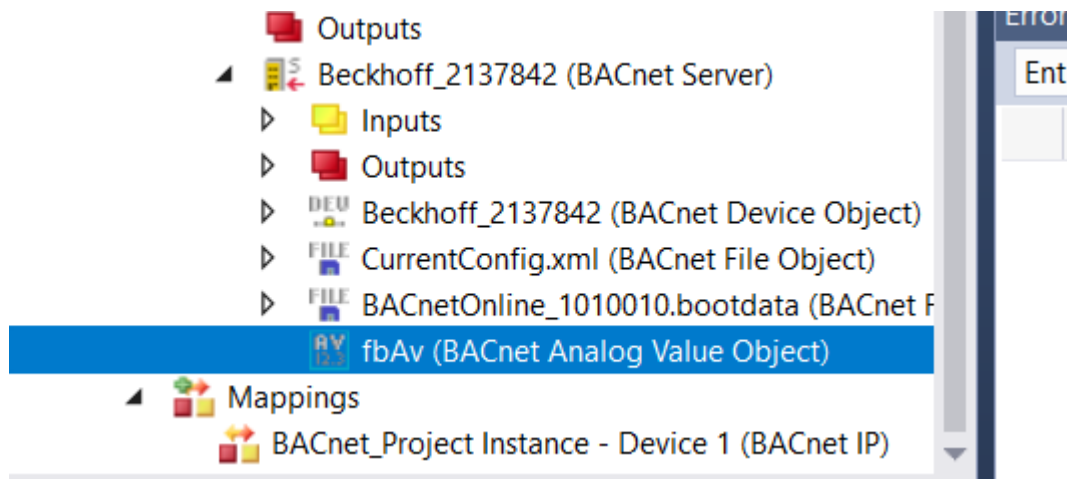
4.2.3 Testen von BACnet mit dem Systemmanager

Die von der SPS erzeugten BACnet-Objekte sind vergleichbar mit den Objekten, die von BACnet-Clients mit dem Service *Create Object* dynamisch erzeugt werden. Befolgen Sie diese Anweisungen, um dynamische Objekte im System Manager zu scannen:

1. Doppelklicken Sie auf den BACnet-Server und navigieren Sie zur Seite *Settings*. Verwenden Sie die Funktion *Scan* im Abschnitt *Dynamic Object Management*.



⇒ Der Baum unterhalb des BACnet-Servers zeigt nun das Objekt *fbAv* (dasjenige, das im obigen Beispiel erstellt wurde).



2. Doppelklicken Sie auf dieses Objekt.

Name	ID	Value	Type
ObjectIdentifier	75	AnalogValue:10000	BACnetObjectIdentifier
ObjectName	77	fbAv	ASCII_UTF8
ObjectType	79	AnalogValue	BACnetObjectType
Description	28		ASCII_UTF8
PresentValue	85	0	Real
StatusFlags	111	⚠ ⚡ ⚙ ⌚	BACnetStatusFlags
EventState	36	state_normal	BACnetEventState
Reliability	103	no_fault_detected	BACnetReliability
OutOfService	81	<input type="checkbox"/>	Bool
Units	117	Other_no_units	BACnetEngineeringUnits
CovIncrement	22	0, 1	Real
PriorityArray	87	16 Elements	BACnetPriorityValue[]
RelinquishDefault	104	0	Real
ActivePriority	517	17	UnsignedInteger
EventEnable	35	⚠ ⚡ ⌚	BACnetEventTransitionBits
NotificationClass	17	0	UnsignedInteger
NotifyType	72	alarm	BACnetNotifyType
TimeDelay [s]	113	0	UnsignedInteger
LimitEnable	52	⬇ ⬆	BACnetLimitEnable
HighLimit	45	0	Real
LowLimit	59	0	Real
Deadband	25	0	Real
AckedTransitions	0	⚠ ⚡ ⌚	BACnetEventTransitionBits
EventTimeStamps	130	{*:*:*:*:*:*:*:*:*:*:*:*:*:*:*:*:*}	BACnetTimeStamp[]
EventMessageTexts	351	{;:}	CharacterStringExtList
EventMessageTextsConfig	352	{To Off-Normal;To Fault;To Normal}	CharacterStringExtList

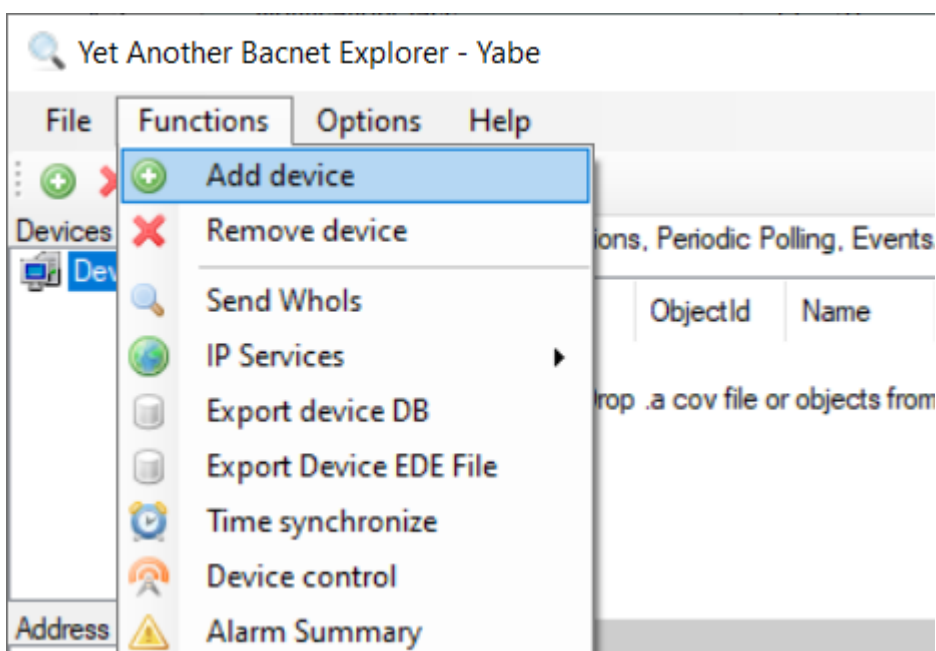
⇒ Es öffnet sich die Online-Ansicht dieses Objekts.

4.2.4 Testen von BACnet mit einem BACnet Explorer

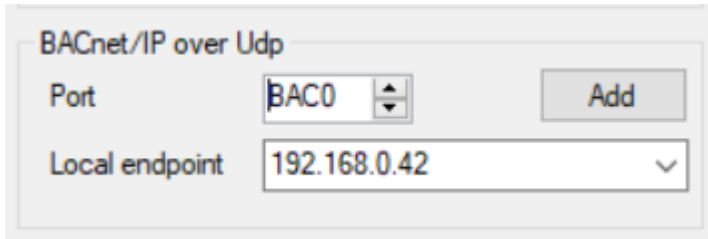
In diesem Beispiel wird das Tool YABE - Yet Another BACnet Explorer verwendet. Das Tool kann hier heruntergeladen werden: <https://sourceforge.net/projects/yetanotherbacnetexplorer/>.

✓ Starten Sie YABE.

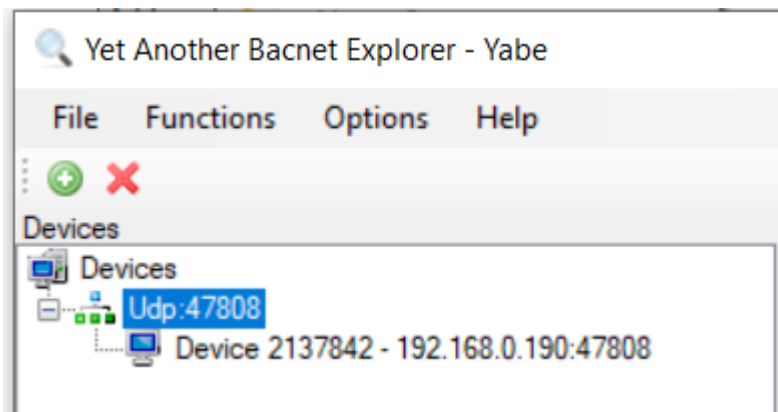
1. Wählen Sie *Add device*. Diese Funktion ist über das Menü *Functions*, die grüne Schaltfläche mit dem Pluszeichen oder durch Rechtsklick auf *Devices* erreichbar.



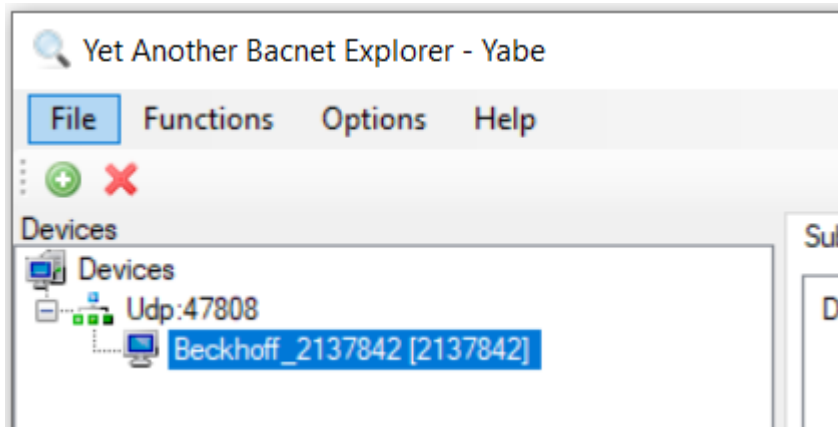
- ⇒ Ein Dialog mit der Bezeichnung *Scan* wird geöffnet.
- 2. Wählen Sie in der Kategorie BACnet/IP over UDP den Standard-BACnet-Port BAC0 (entspricht 47808 in dezimaler Schreibweise) und wählen Sie den Netzwerkadapter Ihres Engineering-PCs (*Local endpoint*).
- 3. Wählen Sie die Schaltfläche *Add* (die Schaltfläche rechts vom BACnet-Port).



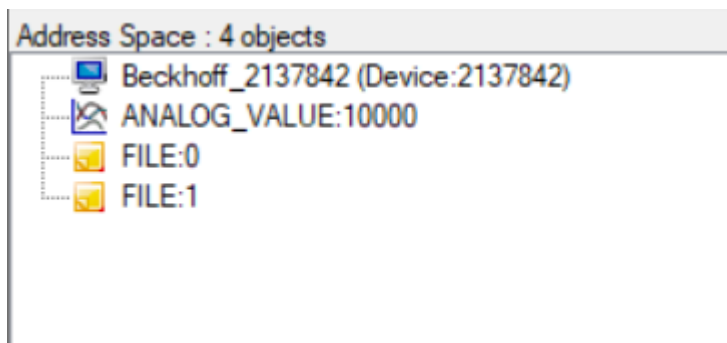
⇒ Das in TwinCAT erstellte Gerät sollte nun erscheinen.



- 4. Wählen Sie das Gerät aus. Dadurch wird der Device Object Name aufgelöst und die Auswertung der im Gerät enthaltenen BACnet-Objekte gestartet.

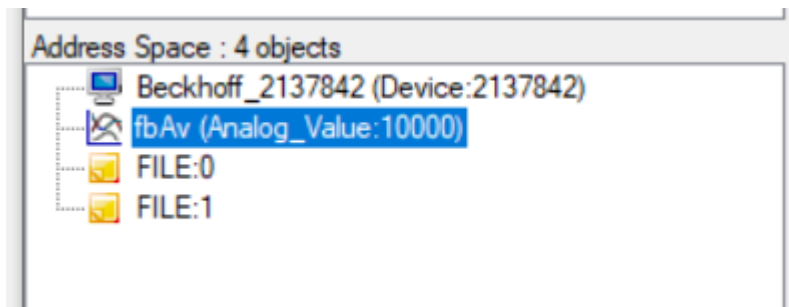


⇒ Im Gerätefenster stellen die Tree Items die im Gerät enthaltenen Objekte dar.



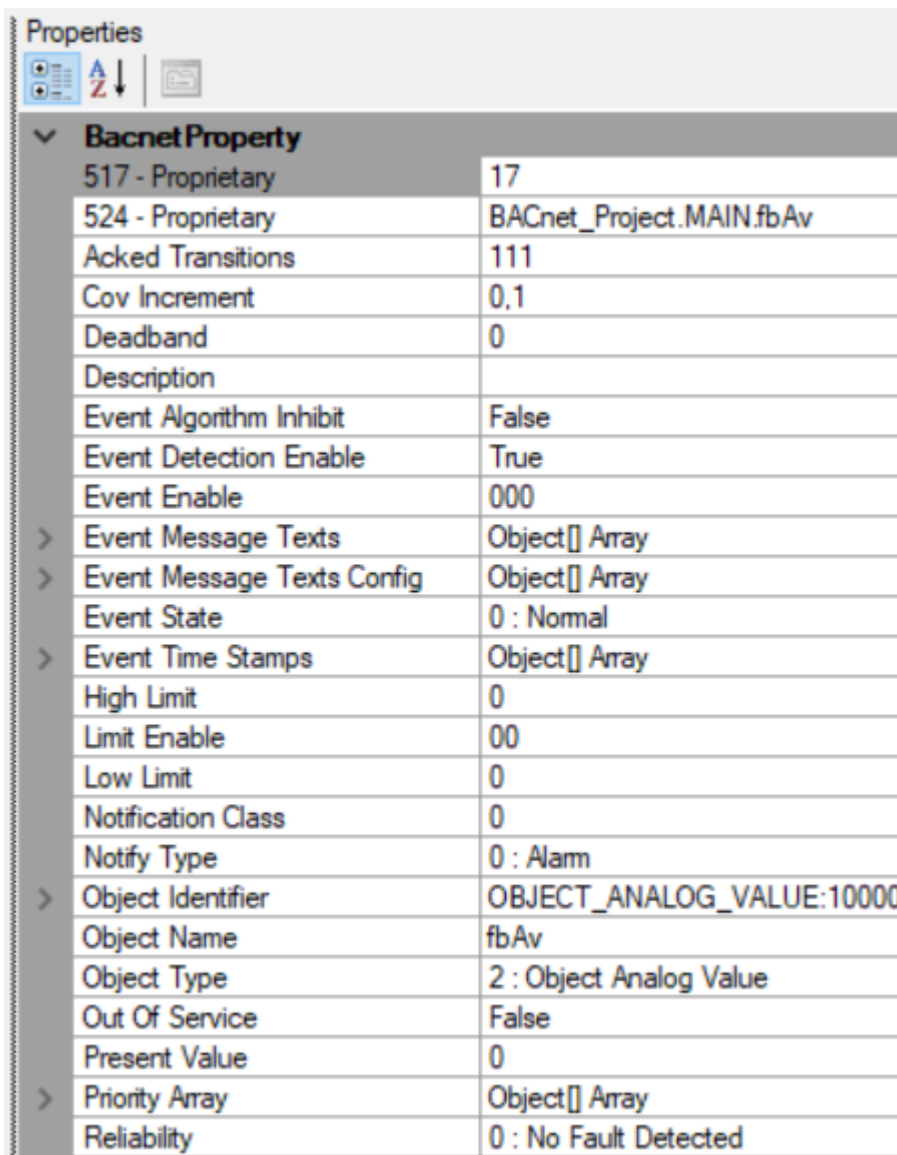
- 5. Wählen Sie *ANALOG_VALUE:10000*.

⇒ Damit beginnt die Auflösung des Objektnamens (*fbAv*).



i Wenn Sie nach der Auswahl eines Geräts die Taste CTRL-N drücken, werden alle Objektnamen aller Objekte im Gerät aufgelöst. Dies kann einige Zeit in Anspruch nehmen, insbesondere, wenn langsamere MS/TP-Geräte durchsucht werden (MS/TP = Master Slave Token Passing = serielles BACnet auf Basis von RS485).

⇒ Durch Auswahl eines Objekts in der Liste der Objekte wird die Anzeige der Objekteigenschaften im rechten Fenster *Properties* geöffnet.

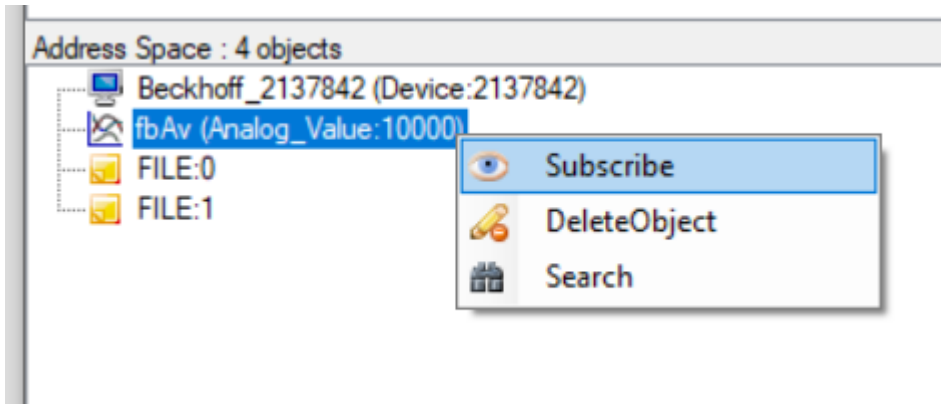


i Dieses Fenster aktualisiert die Property-Werte nicht automatisch! Um die Eigenschaften erneut zu lesen, wählen Sie ein anderes Objekt aus und kehren Sie zu dem beobachteten Objekt zurück.

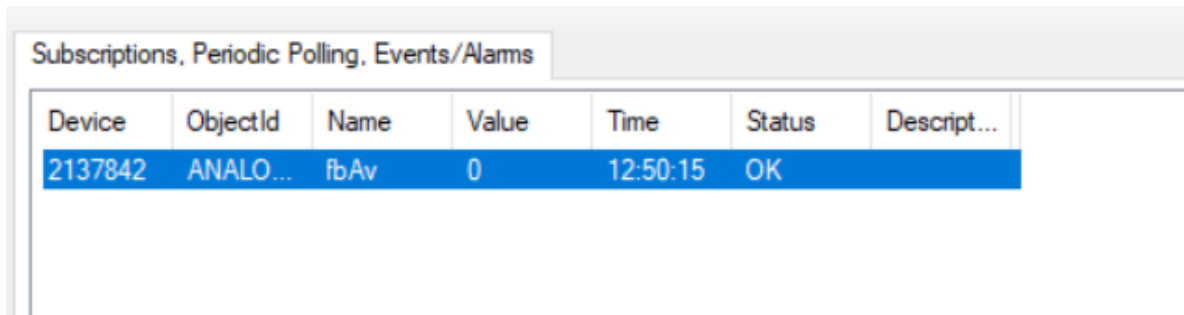
Change of Value (COV) abonnieren

COV ist eine BACnet-Prozedur, um Änderungen des Present Value und der Status Flags automatisch an BACnet-Clients zu melden, die das Objekt abonniert haben.

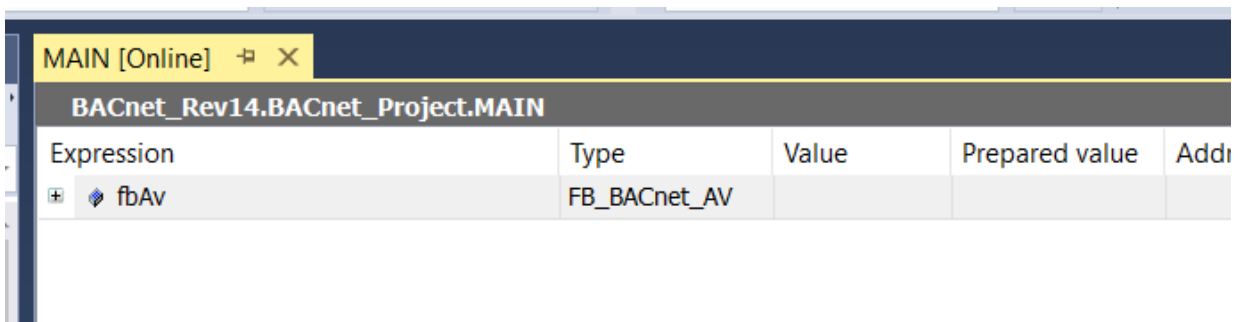
1. Um COV in YABE zu verwenden, klicken Sie mit der rechten Maustaste auf das Objekt und wählen Sie *Subscribe*. Alternativ können Sie das Objekt in das Fenster *Subscriptions* ziehen und ablegen.



⇒ Das ausgewählte Objekt wird nun im Fenster *Subscriptions* angezeigt und wird automatisch aktualisiert, wenn sich der Present Value und/oder die Status Flags des Objekts ändern. Bei Analog-Objekttypen wird die Eigenschaft COV-Increment verwendet, um kleinere Wertänderungen zu unterdrücken. Die Lebensdauer des Abonnements kann im Dialog Settings (Subscriptions Lifetime) angepasst werden.



2. Kehren Sie zurück zu TwinCAT, melden Sie sich in der SPS an und öffnen Sie die POU MAIN.
3. Erweitern Sie die Anzeige des FB *fbAv* durch einen Klick auf das Plus-Symbol.



4. Navigieren Sie zu den Variablen *bEnPgm* und *fValPgm*.

5. Fügen Sie die folgenden Werte in die Spalte *Vorbereiteter Wert* ein. Wenn bEnPgm auf TRUE gesetzt ist, wird der Wert fValPgm auf den Present Value mit der angegebenen Priorität für das Programm gesetzt (Standard: 15).

+	rAckPLCTrig	R_TRIG		
	bEnPgm	BOOL	FALSE	TRUE
	fValPgm	REAL	0	42
	fRelinquishDefault	REAL	0	
	bOutOfService	BOOL	FALSE	
	fCOVIncrement	REAL	0.1	
	eUnit	E_BA_UNIT	eOther_NoU...	

6. Wählen Sie *Werte schreiben* aus dem PLC-Menü.
 7. Wechseln Sie zur Online-Ansicht im TwinCAT System Manager oder wechseln Sie zurück zu YABE.
 ⇒ Die Eigenschaft Present Value sollte nun den Wert „42“ anzeigen.

Subscriptions, Periodic Polling, Events/Alarms						
Device	ObjectId	Name	Value	Time	Status	Descript...
2137842	ANALO...	fbAv	42	13:00:52	OK	

5 SPS-Bibliothek: Tc3_BACnetRev14

Diese Bibliothek enthält die folgenden Elemente:

DUTs = Data Unit Types

GVLs = Global Variable Lists

POUs = Program Organizational Units

Version = Globale Version

5.1 DUTs

Die Datentypen sind in drei Hauptbereiche unterteilt:

5.1.1 Aufzählungen

Dieser Abschnitt enthält die erforderlichen BACnet-Aufzählungen. Eine Ausnahme ist die Liste der aufgezählten technischen Einheiten (E_BA_Unit). Diese ist in der Bibliothek Tc3_BA2_Common zu finden (da diese von BACnet und den Gebäudeautomationsbibliotheken häufig gemeinsam verwendet wird).

5.1.2 Schnittstellen

Dieser Abschnitt enthält eine Liste von Schnittstellen, von denen die meisten in den Basisobjekttypen verwendet werden.

5.1.3 Typen

Dieser Abschnitt enthält eine Liste spezifischer Datentypen, wie ObjectIdentifier oder PropertyList.

5.2 GVLs

Die globalen Variablenlisten sind in zwei Teile aufgeteilt:

5.2.1 Version

Die Variable `stLibVersion_Tc3_BACnetRev14` vom Typ `ST_LibVersion` kann verwendet werden, um die Version der Bibliothek mit der als Projektinformation gespeicherten Version zu vergleichen.

Die TwinCAT3-Bibliotheksversion besteht aus den folgenden Elementen:

`iMajor` = Hauptversionsnummer

`iMinor` = Nebenversionsnummer

`iBuild` = Buildnummer

`iRevision` = Revisionsnummer

Darüber hinaus enthält diese Struktur die Versionsinformationen in Form einer mit Punkten getrennten Zeichenkette.

Global_Version [Online] Library Manager FB_BACnet_101_Nov...net_MSTP [Online]			
TwinCAT_Project48.Untitled1.Tc3_BACnetRev14.Global_Version			
Expression	Type	Value	Prepare
stLibVersion_Tc3_BACnetRev14	ST_LibVersion		
iMajor	UINT	4	
iMinor	UINT	0	
iBuild	UINT	20	
iRevision	UINT	11	
nFlags	DWORD	0	
sVersion	STRING(23)	'4.0.20.11'	

5.2.2 BACnet_Globals

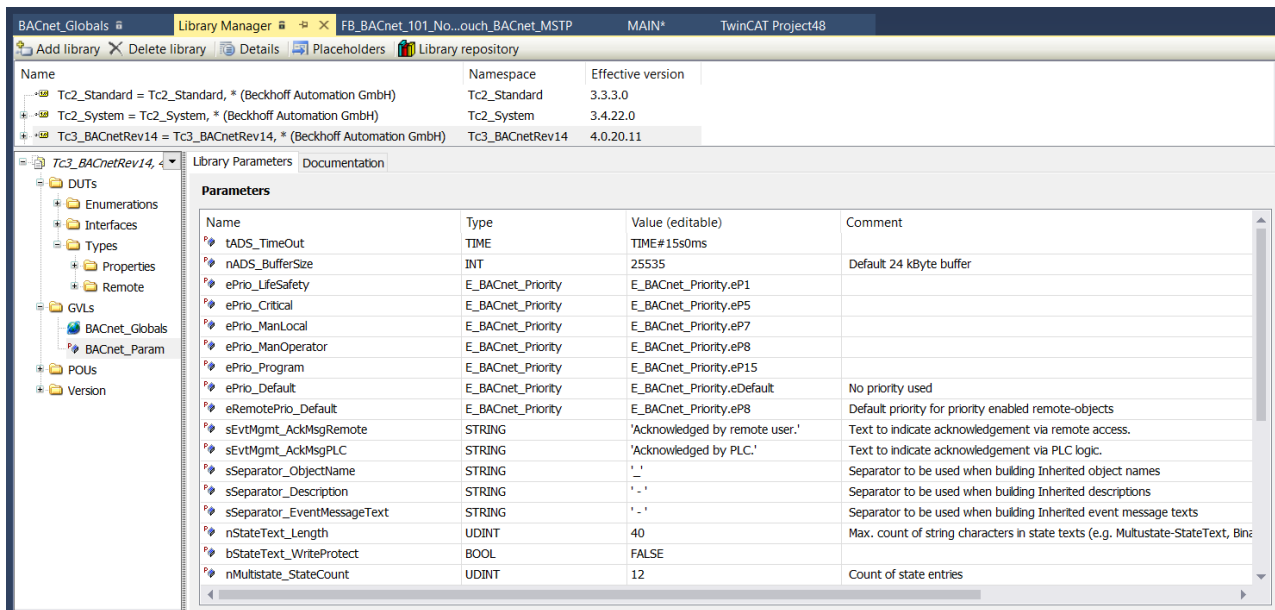
Dieser Teil der Bibliothek spezifiziert globale Einstellungen wie den Default Adapter, BACnet-spezifische Werte wie unterstützte Objekttypen und Error-, Abort- und Reject-Codes.

BACnet_Globals [Online] Library Manager FB_BACnet_101_Nov...net_MSTP [Online] MAIN [Online] TwinCAT Project48					
TwinCAT_Project48.Untitled1.Tc3_BACnetRev14.BACnet_Globals					
Expression	Type	Value	Prepared value	Address	Comment
* DefaultAdapter	FB_BACnet_Adapter				Global linkable BACnet adapter
* DefaultServer	FB_BACnet_Server				Global linkable BACnet device
sCmd_ConcatTitle	STRING(5)	'\'			Command for inheriting path in ent...
nRange_ObjectTypeFirst	INT	0			
nRange_ObjectTypeLast	INT	53			
nRange_ObjectTypeCount	INT	54			
nBACnetString_MaxLength	INT	256			
nBACnetObjId_None	T_BACnet_ObjectIdentifier	4294967295			
nBACnetInstId_Auto	T_BACnet_ObjectIdentifier	4194303			
* aSpec_ObjectTypeShortcut	ARRAY [nRange_ObjectTypeFirst..nRange_ObjectTypeLast] OF STRING(6)				
* aSpec_ErrorType	ARRAY [E_BACnet_ErrorType.First..E_BACnet_ErrorType.Last] OF STRING				
* aSpec_ErrorClass	ARRAY [E_BACnet_ErrorClass.First..E_BACnet_ErrorClass.Last] OF STRING				
* aSpec_ErrorCode	ARRAY [E_BACnet_ErrorCode.First..E_BACnet_ErrorCode.Last] OF STRING				
* aSpec_RejectReason	ARRAY [E_BACnet_RejectReason.First..E_BACnet_RejectReason.Last] OF STRING				
* aSpec_AbortReason	ARRAY [E_BACnet_AbortReason.First..E_BACnet_AbortReason.Last] OF STRING				
tInvalidAmsNetId	T_AmsNetId	'0.0.0.0.0.0'			

5.2.3 BACnet_Param

In diesem Teil der Bibliothek können globale Einstellungen und Parameter für die Bibliotheksinstanz im Projekt definiert und geändert werden. Variablen, die als `CONSTANT` deklariert sind, werden in TwinCAT als Parameter bezeichnet und können je nach den Anforderungen des spezifischen Projekts geändert und angepasst werden.

Bitte beachten Sie, dass die Standardparameter aus dem Bibliotheksrepository unverändert bleiben. Um die Standardwerte wiederherzustellen, entfernen Sie einfach die BACnet Rev14-Bibliothek und fügen Sie sie dem Projekt wieder hinzu.



Die von der Bibliothek bereitgestellten Standardwerte können über die Spalte *Value (Editable)* geändert werden.

Beispiel: Erweitern der Textlänge von Zustandstexteigenschaften.

Die folgende Abbildung zeigt eine Standard-Textlänge von 40 Zeichen für Textinformationen, die in Multistate- und Binary-Objekten bereitgestellt werden.

nStateText_Length	UDINT	40	Max. count of string characters in state texts (e.g. Multistate-StateText, Binary-In/ActiveText)
-------------------	-------	----	--

Für größere Textinformationen kann der Wert von 40 auf 60 Zeichen geändert werden, indem der Wert bearbeitet wird.

Wichtig: Diese Parameter werden (wie andere Variableninitialisierungen in TwinCAT) erst nach dem Aktivieren des Projekts oder nach einem Reset Ursprung und Neustart der SPS angewendet.

5.3 POU's

Die Funktionsbausteine im Bibliotheksbereich POU's sind für die Programmierung von BACnet-Servern und BACnet-Clients vorgesehen.

Der Abschnitt POU's ist in diese Abschnitte unterteilt:

Dynamic Objects: Der Dynamic Object Manager wird hauptsächlich für Anwendungen verwendet, die Objekte auf Basis einer Konfigurationsdatei oder zur Laufzeit erstellen oder löschen (z. B. eine Visualisierung).

Helper: Dieser Abschnitt enthält Hilfsfunktionen wie Konvertierungsfunktionen, Funktionen zur Erzeugung komplexer Datentypen usw.

Local: Dieser Abschnitt enthält Funktionsbausteine für lokale / Server-Anwendungen. Das heißt, TwinCAT stellt als Server BACnet-Objekte zur Verfügung, die Werte aus der SPS repräsentieren.

Remote: Dieser Abschnitt enthält Funktionsbausteine für Remote-/Client-Anwendungen. Das heißt, TwinCAT greift als Client auf andere Geräte zu und verarbeitet Werte von anderen Geräten in der SPS.

FB_BACnet_Adapter: Dieser Funktionsbaustein stellt einen BACnet-Adapter im Systemmanager dar. Bitte beachten Sie, dass der Default Adapter bereits in der GVL BACnet_Globals vorhanden ist und automatisch aufgerufen wird, sobald ein BACnet-Funktionsbaustein im SPS-Code aufgerufen wird.

5.3.1 Namenskonventionen

Die Namen der Funktionsbausteine für lokale / Server-Objekte beginnen mit dem Präfix "FB_BACnet_", gefolgt von der Kurzform des Objekttyps (z. B. ACC = Accumulator, AI = Analog Input, usw.).

Die Namen von Remote-/Client-Objekt-Funktionsbausteinen beginnen mit dem Präfix "FB_BACnetRM_", gefolgt von der Kurzform des Objekttyps.

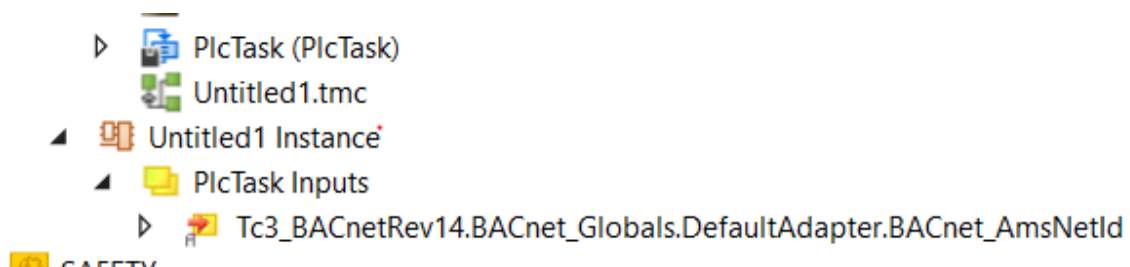
5.3.2 FB_BACnet_Adapter

Der Funktionsbaustein FB_BACnet_Adapter repräsentiert ein BACnet-Devices, das im TwinCAT System Manager unter E/A-Geräte konfiguriert ist. Um Verwechslungen mit dem BACnet-Device-Objekt zu vermeiden, wird ein BACnet-Device im Rahmen dieser Bibliothek als BACnet-Adapter bezeichnet. Bei BACnet/IP wird ein BACnet-Adapter an einen Netzwerkadapter angeschlossen, bei MS/TP wird ein BACnet-Adapter an eine EL6861-Klemme angeschlossen.

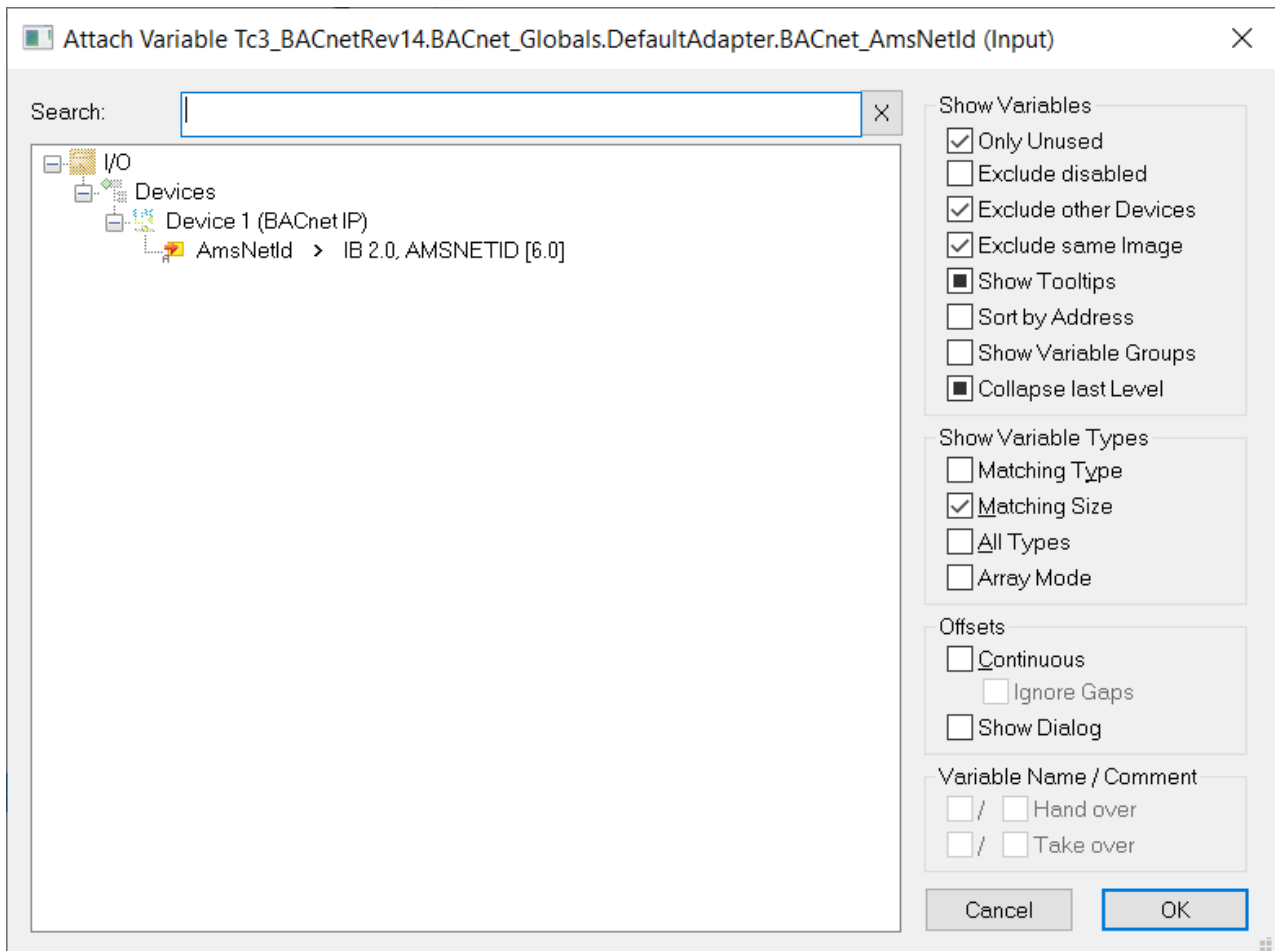
Eine Verbindung zwischen einer Instanz des Funktionsbausteins FB_BACnet_Adapter und dem zugehörigen E/A-Gerät kann durch die Verbindung der AmsNetId-Variablen hergestellt werden.

5.3.2.1 Standard-Adapter

Durch Hinzufügen der Bibliothek *Tc3_BACnetRev14* wird automatisch ein Standardadapter erzeugt. Dieser Adapter wird als globale Variable innerhalb der Bibliothek konfiguriert.



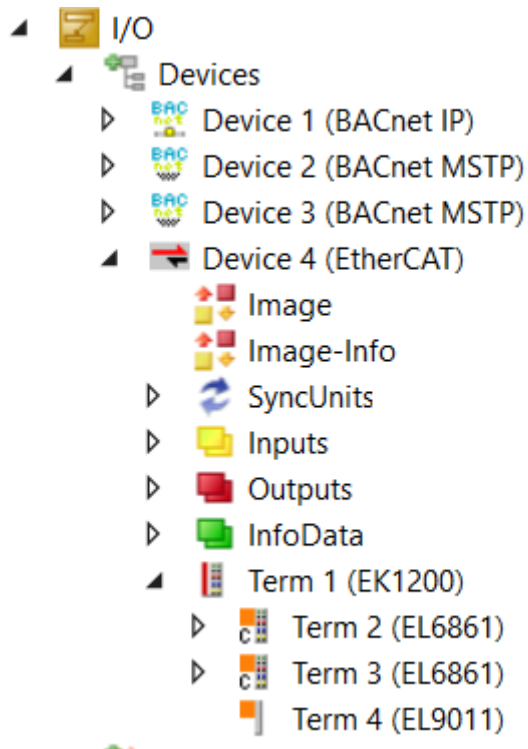
Dieser Adapter muss mit dem Gerät verbunden werden (z. B. BACnet/IP).



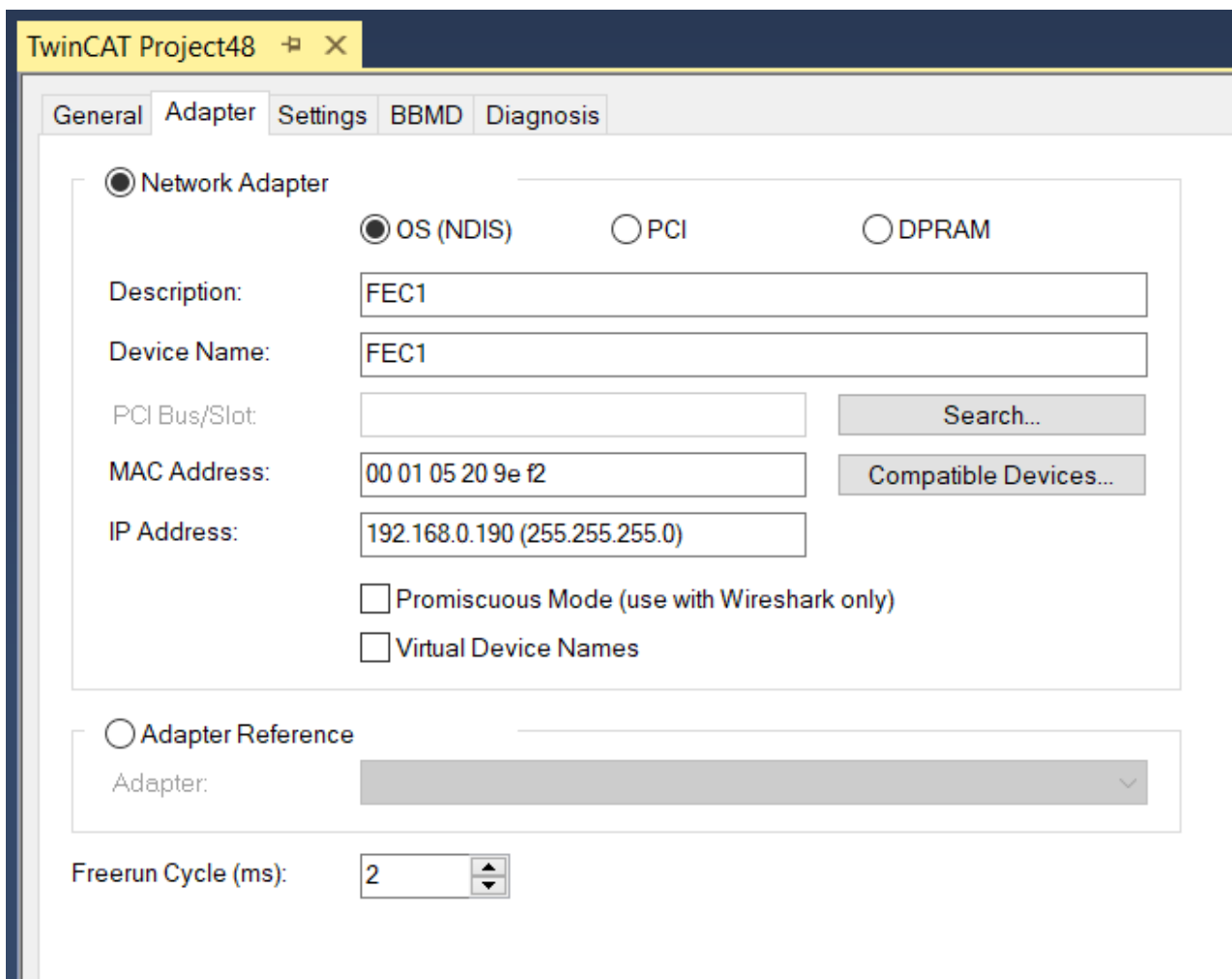
5.3.2.2 Verwendung mehrerer BACnet-Adapter

Das folgende Beispiel veranschaulicht die Verwendung mehrerer BACnet-Adapter, die mit verschiedenen Geräten verbunden sind.

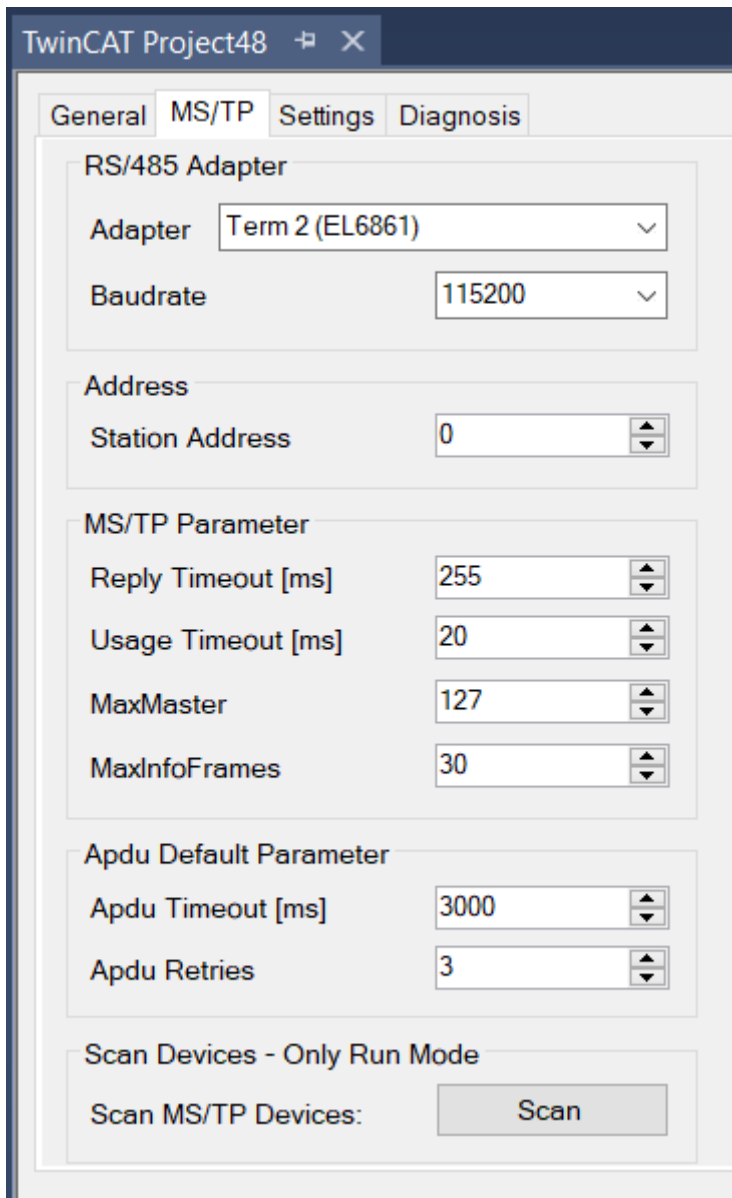
Das unten stehende Beispiel verwendet insgesamt drei Adapter, 1 x BACnet/IP und 2 x MS/TP unter Verwendung der EL6861-Klemmen. Das nächste Bild zeigt die Übersicht im TwinCAT System Manager.



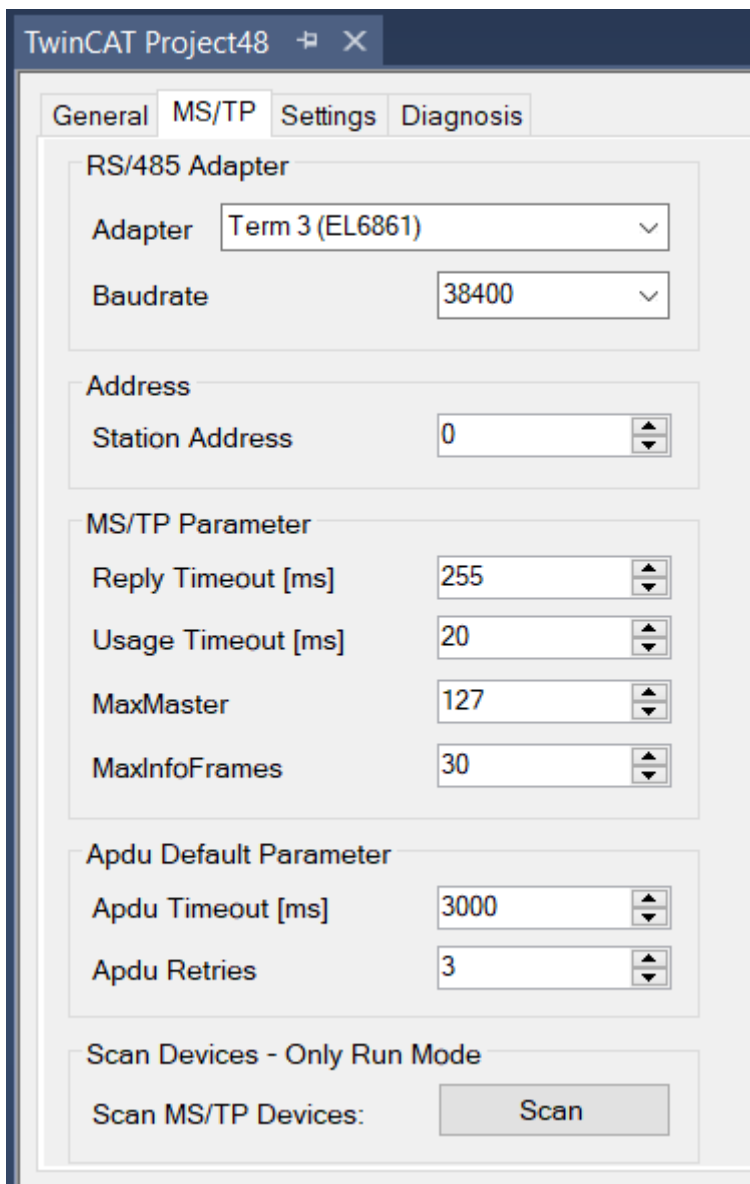
Das Gerät BACnet/IP wird an den Netzwerkadapter FEC1 des CX9020 angeschlossen.



Das erste MS/TP-Gerät *Device 2 (BACnet MSTP)* ist mit der Klemme *Term 2 (EL6861)* verbunden.



Das zweite MS/TP-Gerät *Device 3 (BACnet MSTP)* ist mit der Klemme *Term 3 (EL6861)* verbunden.



Alle drei Geräte sind nun mit der SPS verbunden.

In diesem Beispiel wurden die an das Netzwerk angeschlossenen Geräte bereits gescannt und sind als BACnet-Client-Referenzen verfügbar. Mit Hilfe der Funktion *FB Code* (siehe separates Kapitel für Details) wurde Code generiert, um auf die Objekte über Funktionsbausteine aus der BACnet-Bibliothek zuzugreifen.

Das nächste Fenster zeigt die Variablen und den Code in der POU MAIN, die aus den erstellten Client-Funktionsbausteinen kopiert wurden.

```






FB_BACnet_101_No...ouch_BACnet_MSTP  MAIN* x TwinCAT Project48
1  PROGRAM MAIN
2  VAR
3    {attribute 'ToLinkTo' := '.BACnet_AmsNetId := TIID*Device 2 (BACnet MSTP)^Inputs*AmsNetId'}
4    fbMstpDevice_2 : FB_BACnet_Adapter;
5    fbNovoCon_S_0060879 : FB_BACnet_NovoCon_S_0060879 :=(Client:=(Adapter := fbMstpDevice_2, nDeviceInstance := 60879));
6
7    {attribute 'ToLinkTo' := '.BACnet_AmsNetId := TIID*Device 3 (BACnet MSTP)^Inputs*AmsNetId'}
8    fbMstpDevice_3 : FB_BACnet_Adapter;
9    fb_101_Novos_Touch_BACnet_MSTP : FB_BACnet_101_Novos_Touch_BACnet_MSTP :=(Client:=(Adapter := fbMstpDevice_3, nDeviceInstance := 101));
10 END_VAR
11
12
13 // cyclic call first MS/TP adapter and Danfoss NovoCon valve actuator
14 fbMstpDevice_2();
15 fbNovoCon_S_0060879();
16
17 // cyclic call second MS/TP adapter and thermokon Novos room control unit
18 fbMstpDevice_3();
19 fb_101_Novos_Touch_BACnet_MSTP();
20

```

Der Code `{attribute 'TcLinkTo' := ...}` erzeugt eine dynamische Verbindung zu dem referenzierten BACnet-Device. Das Symbol für die dynamische Verbindung ist im System Manager grün markiert.

Der Standardadapter wird über eine manuelle Verbindung angeschlossen.

Die drei Verbindungen sehen wie folgt aus:

- ▲  Untitled1 Instance
 - ▲  PlcTask Inputs
 - ▶  Tc3_BACnetRev14.BACnet_Globals.DefaultAdapter.BACnet_AmsNetId
 - ▶  MAIN.fbMstpDevice_2.BACnet_AmsNetId
 - ▶  MAIN.fbMstpDevice_3.BACnet_AmsNetId

6 Programmierung eines BACnet-Servers

Ein BACnet-Server ist ein Gerät, das Objekte und Services für andere Geräte, z. B. eine MBE (Management Bedien-Einrichtung), bereitstellt. Jeder BACnet-Server benötigt ein Device-Objekt. Die Eigenschaften ObjectList und StructuredObjectList des Device-Objekts ermöglichen den Zugriff auf die Server-"Datenbank" (die im Gerät enthaltenen Objekte).

Die Funktionsbausteine, die BACnet-Objekte in einem Server darstellen, befinden sich im Ordner POU/Local/Objects.

Im Folgenden werden die Funktionsbausteine für BACnet-Objekte vorgestellt, gefolgt von typischen Anwendungsfällen, wie und welche Funktionsbausteine verwendet werden.

6.1 BACnet-Objekt POU

Dieses Kapitel beschreibt die Funktionsbausteine, die zur Programmierung eines BACnet-Servers verwendet werden.

6.1.1 Funktionsbausteine ohne Suffix

ACC	Accumulator	Dieser Objekttyp stellt akkumulierte (Impuls-)Werte dar.
AI	Analog Input	Dieser Objekttyp repräsentiert physikalische analoge Eingangsinformationen, z. B. einen Sensorwert.
AO	Analog Output	Dieser Objekttyp repräsentiert physikalische analoge Ausgangsinformationen, z. B. über einen 0-10V-Ausgang.
AV	Analog Value	Dieser Objekttyp repräsentiert eine (virtuelle) Analogwert-Information, z. B. einen Sollwert.
BI	Binary Input	Dieser Objekttyp repräsentiert eine binäre Eingangsinformation, z. B. den Zustand einer Lampe oder einer Sicherung.
BO	Binary Output	Dieser Objekttyp repräsentiert eine binäre Ausgangsinformation, z. B. über einen Schaltausgang.
BV	Binary Value	Dieser Objekttyp repräsentiert eine (virtuelle) Binärwert-Information, z. B. einen Fehlerzustand.
CAL	Calendar	Dieser Objekttyp stellt kalendarische (datumsbasierte) Informationen dar.
Device	Device	Dieser Objekttyp stellt das physische Gerät dar. Sie enthält Informationen wie die lokale Uhr, den Hersteller, den Modellnamen und mehr.
EE	EventEnrollment	Dieser Objekttyp wird verwendet, um die Ereignisüberwachung zusätzlich zum Intrinsic Reporting anzuwenden, z. B. um Warngrenzen zu implementieren.
ELOG	Eventlog	Dieser Objekttyp repräsentiert einen Eventlog-Puffer, z. B. um Alarme lokal zu speichern.
File	File	Dieser Objekttyp repräsentiert Dateien, z. B. die aktuelle Konfiguration oder persistente Daten.
Loop	Control Loop	Dieser Objekttyp repräsentiert Regelkreise, z. B. einen PI- oder PID-Kreis.
MI	Multistate Input	Dieser Objekttyp repräsentiert eine physikalische Multistate-Input-Information, z. B. einen lokalen Betriebsartenschalter.
MO	Multistate Output	Dieser Objekttyp repräsentiert eine physikalische Multistate-Output-Information, z. B. einen von der SPS gesteuerten Betriebsartenschalter.
MV	Multistate Value	Dieser Objekttyp stellt eine (virtuelle) Multistate-Wert-Information dar, z. B. einen Programmparameter.
NC	Notification Class	Dieser Objekttyp stellt eine Alarmklasse zur Benachrichtigung von Empfängern dar.
PC	Pulse Converter	Dieser Objekttyp stellt eine umgewandelte Impulsinformation dar, z. B. den Energieverbrauch in kWh.
Prog	Program	Dieser Objekttyp stellt das SPS-Programm dar.
SchedA	Schedule Analog	Dieser Objekttyp stellt einen Zeitschaltplan mit Analogwerten dar.
SchedB	Schedule Binary	Dieser Objekttyp stellt eine Liste von Binärwerten dar.
SchedM	Schedule Multistate	Dieser Objekttyp stellt einen Zeitschaltplan mit Multistate-Werten dar.
TLM	Trendlog Multiple	Dieser Objekttyp stellt ein Trendlog-Objekt dar, das mehrere Kanäle unterstützt.
Tlog	Trendlog	Dieser Objekttyp stellt ein Trendlog-Objekt dar, das einen einzelnen Kanal unterstützt.
View	Structured View	Dieser Objekttyp stellt eine benutzerorientierte Objekthierarchie dar.

6.1.2 Primitive Value-Objekttypen

Date	Single Date Value	Dieser Objekttyp stellt eine bestimmte einzelne Datumsinformation dar (Tag, Monat, Jahr-1900, Wochentag).
DateP	Date Pattern Value	Dieser Objekttyp stellt ein Datumsmuster dar. Der Musterwert 255 kann als Platzhalter verwendet werden.
DateTime	Date and Time Value	Dieser Objekttyp stellt eine Kombination aus einem bestimmten Datum und einer bestimmten Uhrzeit dar.
DateTimeP	Date and Time Pattern Value	Dieser Objekttyp stellt eine Kombination von Datums- und Zeitmustern dar.
INT	Signed Integer Value	Dieser Objekttyp stellt einen vorzeichenbehafteten Integer-Wert dar.
LAV	Large Analog Value (LREAL)	Dieser Objekttyp repräsentiert einen großen Analogwert (8 BYTE LREAL).
String	Character String Value	Dieser Objekttyp stellt eine String-Information dar.
Time	Time Value	Dieser Objekttyp steht für eine bestimmte Zeit (Stunde, Minute, Sekunde, Hundertstelsekunde).
TimeP	Time Pattern Value	Dieser Objekttyp stellt einen Zeitwert dar, der Muster unterstützt.
UINT	Unsigned Integer Value	Dieser Objekttyp stellt einen positiven Integer-Wert (UNSIGNED) dar.

Einige Funktionsbausteinnamen enthalten ein Suffix:

_IO: Diese FBs sind für den Anschluss an Hardware-Klemmenkanäle vorgesehen. Die erforderlichen AT%I*- und AT%Q*-Variablen werden in der FB-Implementierung angegeben.

Z. B. stellt eine BinaryOutput_IO diese Variablen zur Verfügung, um sich mit Klemmenkanälen zu verbinden:

```
bRawOvrrd AT %I* : BOOL; // Raw overridden (Optional)
bRawValFdbk AT %I* : BOOL; // Raw feedback Value (Optional)
bRawVal AT %Q* : BOOL; // Raw value
```

_ECAT: Diese FBs sind für den Anschluss an EtherCAT-Hardware-Klemmenkanäle vorgesehen. Der Unterschied zu den FBs mit dem Suffix _IO ist die Variable nRawState, mit der der Zustand Underrange/Overrange bestimmt wird.

_Raw: Falls die SPS den Wert für das BACnet-Objekt bereitstellt, können der Rohwert und der Rohstatus vom SPS-Programm bereitgestellt werden.

```
Beispiel: FB_BACnet_AI_Raw:
nRawState : USINT; // Raw state
// - Underrange: 0x01
// - Overrange: 0x02
// - Error: 0x04
nRawVal : INT; // Raw value
```

_Disp: Diese Funktionsbausteine beziehen sich auf Wertobjekttypen, die schreibgeschützte Werte wie die aktuelle Raumtemperatur darstellen. Present_Value ist nicht beschreibbar.

_Event: Diese Funktionsbausteine beziehen sich auf ein Wertobjekt ähnlich dem Typ _Disp (schreibgeschützt). Darüber hinaus unterstützen diese Funktionsbausteine Event Reporting. Present_Value ist nicht beschreibbar.

_Setp: Diese Funktionsbausteine beziehen sich auf Sollwerte. Sollwerte sind beschreibbare BACnet-Objekte ohne Befehlspriorisierung („last writer wins“). Present_Value ist beschreibbar, aber nicht kommandierbar.

_Buf: Diese Funktionsbausteine implementieren einen Log-Puffer in der SPS. Dies kann für lokale Visualisierungen von Trendlog- oder Eventlog-Informationen verwendet werden.

_5P: Diese Funktionsbausteine beziehen sich auf kommandierbare Ausgangs- oder Wertobjekte, die einen Satz von 5 Prioritäten für die Befehlspriorisierung bereitstellen. Die Prioritätsstufe jeder der 5 Prioritäten kann in den globalen BACnet_Param-Einstellungen festgelegt werden.

Standardeinstellungen:

Prioritätskategorie	Standardwert
Life-Safety	1
Critical Equipment Control	5
Minimale Ein-/Ausschaltzeiten	6 (durch den BACnet-Standard festgelegt, kann nicht geändert werden)
Manual Local Operator	7 (lokale Visualisierung)
Manual Operator	8 (GLT)
Programm (SPS)	15

_IO5P: Wie **_5P**, aber zum Anschluss an Hardware-Klemmen verwendet.

_Raw5P: Wie **_5P**, aber der Prozesswert wird von der SPS bereitgestellt.

_Ref: (Loop-Objekttyp) Diese Funktionsbausteine beziehen sich auf Loop-Objekte, die externe Analog-Objekte für den Sollwert, die Stellgröße und den Feedback-Wert unterstützen. Siehe Kapitel Regelkreise.

6.2 Typische BACnet-Szenarien

In diesem Kapitel werden typische Szenarien vorgestellt, die bei BACnet-Projekten häufig verwendet werden.

6.2.1 Befehlspriorisierung

Die Befehlspriorisierung bietet einen Mechanismus, um zu bestimmen, welcher Prozess oder welche Benutzerrolle Vorrang vor niedriger priorisierten Prozessen oder Benutzerrollen hat. BACnet spezifiziert 16 Prioritätsstufen, wobei 1 die höchste Priorität (Manual Life-Safety) darstellt und 16 die niedrigste und Standardpriorität.

Present_Value wird immer aus dem Wert im höchstpriorisierten Slot genommen. Ein Client kann eine Priorität an einem bestimmten Slot entfernen, indem er bei der angegebenen Priorität NULL schreibt.

Wenn keine Priorität aktiv ist (alle 16 Slots haben den Wert NULL), wird der Wert für die Eigenschaft Present_Value von der Eigenschaft Relinquish_Default übernommen.

Die Priorität wird mit einem WriteProperty- oder WritePropertyMultiple-Service angegeben. Fehlt die Priorität, so wird 16 als Standardpriorität verwendet.

Priorität	Bedeutung
1	Manual Life-Safety
2	Automatic Life-Safety
3	Verfügbar
4	Verfügbar
5	Critical Equipment Control, z. B. Abtauung
6	Minimum Ein/Aus
7	Verfügbar
8	Manual Operator, z. B. GLT
9	Verfügbar
10	Verfügbar
11	Verfügbar
12	Verfügbar
13	Verfügbar
14	Verfügbar
15	Verfügbar
16	Verfügbar (Standard, wenn keine Priorität angegeben ist)

Funktionsbausteine mit dem Suffix "_5P" implementieren insgesamt 5 der 16 möglichen Prioritäten (was normalerweise für die meisten Projekte ausreicht).

6.2.2 Event Reporting

BACnet bietet zwei verschiedene Verfahren zur Implementierung von Event Reporting:

Intrinsic Reporting und Algorithmic Change Reporting.

Die Bestimmung, ob es sich bei einem Ereignis um eine unkritische Information, z. B. eine Betriebsmeldung, oder um einen abnormalen Zustand, z. B. einen Alarm, handelt, wird durch die Eigenschaft `Notify_Type` festgelegt.

6.2.2.1 Intrinsic Reporting

Intrinsic Reporting beobachtet die Eigenschaft `Present_Value`, die Ereignisparameter werden innerhalb desselben Objekts angegeben (z. B. `High_Limit`, `TimeDelay`, `Alarm_Value`, usw.).

Ereigniserzeugende Objekte haben eine Verbindung zu einem `NotificationClass`-Objekt innerhalb desselben Geräts (referenziert durch dieselbe Instanznummer im ereigniserzeugenden Objekt und im `NotificationClass`-Objekt).

Ereignisse werden den Empfängern mitgeteilt, die in der Eigenschaft `Recipient_List` der `NotificationClass` enthalten sind und die mit den aktiven Tagen / der aktiven Zeit im Abonnement übereinstimmen.

6.2.2.2 Algorithmic Change Reporting

Algorithmic Change Reporting verwendet das Event Enrollment Objekt, um den Ereignisalgorithmus und die Ereignisparameter zu spezifizieren. Das Event Enrollment Objekt hat eine logische Verbindung zu einem `Notification Class` Objekt wie jedes andere ereigniserzeugende Objekt.

Ein typisches Szenario in Projekten implementiert ein Event Enrollment, das Warngrenzen an eine `Warning Notification Class` meldet. Intrinsic Reporting kann dann zur Erkennung von Alarmbedingungen (High- und Low-Limit) verwendet werden, die einer anderen Meldeklasse für Alarmer gemeldet werden.

6.2.3 Scheduling

BACnet Time Scheduling bietet Strategien zur Implementierung von datums- und/oder zeitbasierten Funktionen.

BACnet verwendet dafür zwei Objekttypen, `Calendar` und `Schedule`.

Der Objekttyp `Calendar` liefert Informationen, wenn HEUTE mit einem oder mehreren Einträgen in der Eigenschaft `Date_List` übereinstimmt. Dies wird z. B. verwendet, um festzustellen, ob HEUTE ein Feiertag ist oder nicht (vorausgesetzt, die Datumsliste enthält diese Information). Ein Eintrag in der `Date_List`-Eigenschaft kann ein einzelnes Datum, ein Datumsbereich oder ein `WeekNDay` (eine Kombination aus Tagen, Wochen und Monaten) sein.

Der Objekttyp Zeitschaltplan basiert hauptsächlich auf zwei Eigenschaften.

Die Eigenschaft `Weekly_Schedule` stellt ein Zeitschaltplanprogramm (Zeit/Werteliste) für jeden der Tage von Montag bis Sonntag dar.

Die Eigenschaft `Exception_Schedule` gibt eine Liste von Ausnahmen an, die Vorrang vor dem `WeeklySchedule` haben. Die Ausnahmen können als einzelnes Datum, `DateRange` oder `WeekNDay` angegeben werden oder auf einer `Calendar`-Referenz innerhalb desselben Geräts basieren. Leider ist es vom BACnet-Standard her nicht möglich, einen globalen Kalender zu spezifizieren und von anderen Geräten aus auf dieses Objekt zu verweisen. Bei Bedarf muss der Inhalt eines globalen Kalenders in lokale Kalenderobjekte in jedem BACnet-Device kopiert werden.

Zeitplanobjekte enthalten eine `Schedule_Default`-Eigenschaft, die den Rückfallwert um Mitternacht angibt, außer der Wert wird am nächsten Tag um 0.00.00:00 Uhr wiederholt.

Das Zeitplanobjekt hat keinen spezifischen Datentyp. Der tatsächliche Datentyp wird durch den Weekly- und ExceptionSchedule, Schedule_Default und die Liste der ObjectPropertyReferences bestimmt.

ObjectPropertyReferences können Eigenschaften in Objekten innerhalb desselben oder in fremden Geräten enthalten.

Die Priorität des Zeitschaltplan-Prozesses wird über die Eigenschaft Priority_For_Writing im Bereich von 1-16 angegeben.

6.2.4 Trendaufzeichnung

Trendlog-Objekte werden verwendet, um Werte aus einer einzigen Quelle zu protokollieren, Trendlog Multiple-Objekte bieten mehrere Kanäle (werden aber nur selten verwendet).

Die Protokollierung kann auf der Grundlage von zyklischen Abfragen (Polling), COV (nur Tlog) oder ausgelöster Datenerfassung (Trigger) erfolgen.

Neben den Werten der referenzierten Objekteigenschaft werden auch andere Ereignisse aufgezeichnet (z. B. Aktivierung/Deaktivierung der Aufzeichnung, Leeren des Puffers usw.).

Die Größe des Log-Puffers wird durch die Eigenschaft BUFFER_SIZE bestimmt.

Record_Count gibt Aufschluss über die aktuelle Auslastung des Log-Puffers. Das Schreiben eines Wertes von Null in Record_Count löscht den Puffer.

Trendlog-Objekte können mit der Ereigniserzeugung kombiniert werden. In diesem Fall kann ein Notification_Threshold angegeben werden, um die Clients zu benachrichtigen, damit sie die Werte abrufen können, bevor die Protokollierung endet oder ältere Werte überschrieben werden.

6.2.5 Eventaufzeichnung

Die Ereignisaufzeichnung bietet Strategien zur lokalen Speicherung von Ereignissen oder Alarmen, z. B. für den Fall, dass die Verbindung zur GLT unterbrochen wird.

In einer BACnet-Server-Implementierung müssen Event-Log-Objekte die gleiche Instanznummer wie die Notification Class-Objekte verwenden. In diesem Fall wird das Ereignisprotokollobjekt automatisch zugewiesen und ruft alle Ereignisse/Alarme ab, die über dieses Notification Class Objekt verteilt werden.

6.2.6 Regelkreise

BACnet spezifiziert einen Loop-Objekttyp, um die Parameter für Regelkreise bereitzustellen (z. B. P, I, D, Bias, Maximum_Output, usw.). Diese Werte werden als Eigenschaften des Objekttyps Loop bereitgestellt.

Falls der Sollwert, der Stellgrößenwert und der Regelgrößenwert für BACnet-Clients sichtbar sein sollen, können zusätzlich zum Loop-Objekt Analog-Objekte angegeben werden. TwinCAT bietet zwei Implementierungen für Regelkreise:

FB_BACnet_Loop: Loop-Objekt ohne externe Referenzen, der Sollwert und die Stell- und Regelwerte werden von internen Variablen übernommen.

FB_BACnet_Loop_Ref: Loop-Objekt mit externen Referenzen. Eine typische Implementierung kann ein Analog-Value-Objekt für den Sollwert, einen Analogausgang für den Stellgrößenwert und einen Analogeingang für den Regelgrößenwert umfassen.

Beispiel:

```
VAR
// control loop using internal variables
  fbLoopInternal : FB_BACnet_Loop := (
    bEn           := TRUE,
    sDescription  := 'Loop using internal control parameters',
    eOutputUnit   := E_BA_Unit.eOther_Percent,
    eAction       := E_BA_Action.eReverse,
    fProportionalConstant := 5.0,
    fIntegralConstant := 180,
    fSetpoint     := 20
  );
```

```

        fCtrlVal : REAL := 18;

// control loop using external BACnet objects
fbLoopRef_Setpt : FB_BACnet_AV_Setpt := (fValue := 20);
fbLoopRef_CtrlVar : FB_BACnet_AI := (fVal := 18);
fbLoopRef_Y : FB_BACnet_AO := ();
fbLoopRef : FB_BACnet_Loop_Ref := (
    bEn := TRUE,
    sDescription := 'Loop using reference objects',
    stControlledVariableReference :=
        F_BACnet_Reference(fbLoopRef_CtrlVar, PropPresentValue),
    stSetpointReference :=
        F_BACnet_Reference(fbLoopRef_Setpt, PropPresentValue),
    stManipulatedVariableReference :=
        F_BACnet_Reference(fbLoopRef_Y, PropPresentValue),
    eOutputUnit := E_BA_Unit.eOther_Percent,
    eAction := E_BA_Action.eReverse,
    fProportionalConstant := 5.0,
    fIntegralConstant := 180
);
END_VAR
-----
// internal control loop
fbLoopInternal.fCtrlVar := fCtrlVal;
fbLoopInternal();

// control loop using external object references
fbLoopRef_Setpt();
fbLoopRef_CtrlVar();
fbLoopRef_Y();
fbLoopRef();
    
```

6.2.7 TimeSynchronization

Die BACnet TimeSynchronization kann auf der lokalen Zeit basieren. In diesem Fall müssen sich der Zeitmaster und die Zeitempänger in der gleichen Zeitzone befinden. Wenn sich die Geräte in unterschiedlichen Zeitzonen befinden, kann stattdessen UTC-TimeSynchronization verwendet werden. Dazu muss die Eigenschaft UTC_Offset im Device-Objekt auf den Offset in Minuten zur UTC gesetzt werden. Darüber hinaus muss die Sommerzeitinformation vom Gerät bereitgestellt und zusätzlich zur UTC-Zeit berechnet werden.

Die Implementierung eines BACnet-Servers erfordert normalerweise keine besonderen Maßnahmen. Wird eine TimeSynchronization-Nachricht empfangen, wird die interne Uhr der Steuerung aktualisiert.

Wenn die Steuerung als Zeitmaster fungiert, bietet der Funktionsbaustein FB_BACnet_Adapter zwei Methoden, um Uhren in anderen BACnet-Devices zu synchronisieren.

TimeSync

TimeSyncEx

TimeSynchronization ist ein unbestätigter Service. Eine Antwort der Zeitempänger an den Zeitmaster wird nicht erwartet. Alle Geräte, die die Zeitsynchronisationsmeldung empfangen, müssen ihre lokale Uhr einstellen.

Beispiel für eine Zeitmasterfunktionalität:

```

VAR
    stDateTimeSync : ST_BA_DateTime;
    bSuccess : BOOL;
    bSync : BOOL;
END_VAR
-----
If bSync then
    bSync := FALSE;
    bSuccess := BACnet_Globals.DefaultAdapter.TimeSync(pDateTime := ADR(stDateTimeSync),
bSendBroadcast := TRUE);
END_IF
    
```

6.2.8 Abrufen von Diagnoseinformationen

Zum Abrufen von BACnet-Diagnoseinformationen bietet der FB_BACnet_Adapter diese Methode: GetDiagnosis.

GetDiagnosis ermöglicht den Zugriff auf die eingebaute Diagnose im TwinCAT System Manager, die unter dem Reiter "Diagnosis" eines BACnet-Device eingesehen werden kann. Darüber hinaus stellt jeder FB_BACnet_Client zusätzliche Diagnoseinformationen (m_stDiag) für jeden Client zur Verfügung und bietet damit zwei weitere Möglichkeiten, Client-Verbindungen zu beobachten:

Roundtrip-Messung: Es ist möglich, die Zeit einer kompletten zyklischen Anfrage zu bestimmen.

Diagnose der verschiedenen Anfragetypen, diese wird pro Clientverbindung in der Diagnose angezeigt.

Beispiel zum Abrufen von Diagnoseinformationen:

```
VAR
    fbDevice : FB_BACnet_Device;
    stDiagnosis : ST_BACnet_Diagnosis;
    bSuccess : BOOL;
    bGetDiagnosis : BOOL;
END_VAR
-----
fbDevice();
IF bGetDiagnosis THEN
    bGetDiagnosis := FALSE;
    bSuccess := BACnet_Globals.DefaultAdapter.GetDiagnosis( ADR( stDiagnosis ) );
END_IF
```

6.2.9 Scannen anderer BACnet-Devices

In manchen Fällen ist es notwendig, das BACnet-Netzwerk zu scannen (z. B. von einer Visualisierung aus), um externe BACnet-Devices zu identifizieren. Der FB_BACnet_Adapter bietet zwei Methoden, um einen Scanvorgang zu starten:

StartScan

StartScanEx

Der Aufruf einer dieser Funktionen erzeugt eine BACnet Who-Is-Anfrage an das Netzwerk. Alle Geräte, die diese Anfrage erhalten und den Anfrageparametern entsprechen, antworten mit dem I-Am-Service. Das Ergebnis dieses Scanvorgangs kann nach einer angemessenen Wartezeit mit der FB_BACnet_Adapter-Methode *GetScanResult* abgefragt werden.

GetScanResult gibt die Anzahl der im Netz gefundenen externen Geräte zurück, unabhängig von der Puffergröße zum Speichern der Ergebnisse. Im Falle eines Fehlers gibt diese Funktion -1 zurück.

Beispiel-Scanning für andere BACnet-Devices:

```
VAR
    fbDevice : FB_BACnet_Device;
    a_stScanResult : ARRAY[0..MAX_SCANRESULTS] OF ST_BACnetRM_ScanResult;
    bSuccessScan : BOOL;
    bScan : BOOL;
    fbWaitTimer : TON;
    nScanResult : DINT;
END_VAR
-----
VAR CONSTANT
    MAX_SCANRESULTS : UDINT := 200;
    tWaitTime : TIME := T#5S;
END_VAR
-----
fbDevice();
fbWaitTimer( IN := NOT fbWaitTimer.Q, PT := tWaitTime );

IF bScan THEN
    bScan := FALSE;
    nScanResult := -1;
    bSuccessScan := BACnet_Globals.DefaultAdapter.StartScan();
END_IF

IF bSuccessScan AND fbWaitTimer.Q THEN
    bSuccessScan := FALSE;
```

```
nScanResult := BACnet_Globals.DefaultAdapter.GetScanResult( ADR( a_stScanResult ),
MAX_SCANRESULTS );
END_IF
```

6.2.10 Erstellen einer strukturierten Ansicht (AKS)

Ein AKS (**Anlagen-Kennzeichnungs-System**) wird verwendet, um zu BACnet-Objekten zu navigieren, indem die Anlagenansicht und nicht die technologische Ansicht verwendet wird. Ein Facility Manager weiß, in welchem Gebäude, Stockwerk, Raum usw. sich ein BACnet-Objekt befindet.

Es ist eine gute Praxis, eine Kurzform für das AKS in die Objektnamen-Eigenschaft der BACnet-Objekte und eine (wahrscheinlich längere) Information in die Beschreibungseigenschaft zu setzen. Zusätzlich kann der Objekttyp Strukturierte Ansicht (FB_BACnet_View) verwendet werden, um die Benutzeransicht der Anlagenhierarchie bereitzustellen.

Der folgende Code veranschaulicht, wie man eine Navigation mit dem Operator 'V' (Backslash und Slash) erstellt. Dieser Operator kann den folgenden Eigenschaften (Typ Zeichenkette) zugeordnet werden:

Objektnamen

Description

Ereignismeldungstext

Zur Laufzeit wird das Trennzeichen durch die im Abschnitt Global / BACnet_Param angegebenen Zeichen ersetzt.

Codebeispiel für die Erstellung einer AKS-Struktur

```
PROGRAM MAIN
VAR
    fbDPADFirstLevel : FB_BACnet_View := (
        eNodeType := E_BACnet_NodeType.eArea,
        sObjectName := '\\A',
        sDescription := '\\Facilities');

    fbDPADSecondLevel : FB_BACnet_View := (
        iParent := fbDPADFirstLevel,
        eNodeType := E_BACnet_NodeType.eOrganizational,
        sObjectName := '\\B',
        sDescription := '\\Building');

    fbDPADThirdLevel : FB_BACnet_View := (
        iParent := fbDPADSecondLevel,
        eNodeType := E_BACnet_NodeType.eNetwork,
        sObjectName := '\\C',
        sDescription := '\\Floor');

    fbAi : FB_BACnet_AI := (
        iParent := fbDPADThirdLevel,
        sObjectName := '\\ObjectName',
        sDescription := '\\Description',
        sDeviceType := 'TemperatureSensor',
        eUnit := E_BA_Unit.eTemperature_DegreesCelsius,
        fMinPresValue := -50.0,
        fMaxPresValue := 150.0,
        fHighLimit := 100,
        fLowLimit := -25,
        bHighLimitEnable := TRUE,
        bLowLimitEnable := TRUE,
        nNotificationClass := 10,
        aEventEnable := [ TRUE, TRUE, FALSE ],
        aEventMessageTextsConfig := [ '\\Alarm', '\\Fault', '\\Normal' ]);
END_VAR

-----
fbDPADFirstLevel();
fbDPADSecondLevel();
fbDPADThirdLevel();
fbAi();
```

Dieser Code erzeugt drei Structured-View-Objekte und ein Analog-Input-Objekt und verbindet das AKS über die iParent-Elemente, die auf den aufrufenden Funktionsbaustein verweisen.

Die Auswahl, wie die Tree Items im System Manager angezeigt werden, kann in den Global Params der Bibliotheksinstanz festgelegt werden:

Name	Type	Value (editable)	Comment
ePrio_Default	E_BACnet_Priority	E_BACnet_Priority.eDefault	No priority used
eRemotePrio_Default	E_BACnet_Priority	E_BACnet_Priority.eP8	Default priority for remote user
sEvtMgmt_AckMsgRemote	STRING	'Acknowledged by remote user.'	Text to indicate acknowledgement by remote user
sEvtMgmt_AckMsgPLC	STRING	'Acknowledged by PLC.'	Text to indicate acknowledgement by PLC
sSeparator_ObjectName	STRING	'.'	Separator to be used in object names
sSeparator_Description	STRING	'.'	Separator to be used in descriptions
sSeparator_EventMessageText	STRING	'.'	Separator to be used in event message text
nStateText_Length	UDINT	40	Max. count of state text characters
bStateText_WriteProtect	BOOL	FALSE	Write protection for state text
eView_SubordinateAnnotationMode	E_BACnet_AnnotationTitle	E_BACnet_AnnotationTitle.eSymbolName	
nMultistate_StateCount	UDINT	12	Count of state text characters

eSymbolName: Der Symbolname aus der SPS wird als Tree Item Name verwendet.

eObjectName: Der BACnet-Objektname wird als Tree Item Name verwendet.

eDescription: Die BACnet-Beschreibung wird als Tree Item Name verwendet.

Der Baum des System Managers sieht wie folgt aus (im folgenden Beispiel wurde der Objektname verwendet):

- ▲ I/O
 - ▲ Devices
 - ▲ Device 1 (BACnet IP)
 - Image
 - ▶ Inputs
 - ▶ Outputs
 - ▲ Beckhoff_975635 (BACnet Server)
 - ▶ Inputs
 - ▶ Outputs
 - ▶ Beckhoff_975635 (BACnet Device Object)
 - ▶ CurrentConfig.xml (BACnet File Object)
 - ▶ BACnetOnline_1010010.bootdata (BACnet File Object)
 - ▲ A (BACnet Structured View Object)
 - ▲ B (BACnet Structured View Object)
 - ▲ C (BACnet Structured View Object)
 - ObjectName (BACnet Analog Input Object)

Online			
	Name	ID	Value
	ObjectIdentifier	75	AnalogInput:10000
*	ObjectName	77	A_B_C_ObjectName
	ObjectType	79	AnalogInput
*	Description	28	Facilities - Building - Floor - Description
	PresentValue	85	0
*	DeviceType	31	TemperatureSensor
	StatusFlags	111	⚠️ 🖱️ 🔄 🚫
	EventState	36	state_normal

Die Trennzeichen, die für die drei Eigenschaften verwendet werden, die möglicherweise ein AKS enthalten (Objektnamen, Beschreibung und Ereignismeldungstexte), können in der Einstellung Global Params der Bibliothek individuell festgelegt werden.

Name	Namespace	Effective version
Tc2_Standard = Tc2_Standard, * (Beckhoff Automation GmbH)	Tc2_Standard	3.3.3.0
Tc2_System = Tc2_System, * (Beckhoff Automation GmbH)	Tc2_System	3.4.24.0
Tc3_BA2_Common = Tc3_BA2_Common, * (Beckhoff Automation GmbH)	Tc3_BA2_Common	2.1.3.35
Tc3_BACnetRev14 = Tc3_BACnetRev14, * (Beckhoff Automation GmbH)	Tc3_BACnetRev14	4.0.22.17
Tc3_Module = Tc3_Module, * (Beckhoff Automation GmbH)	Tc3_Module	3.3.21.0

Name	Type	Value (editable)	Comment
tADS_TimeOut	TIME	TIME#15s0ms	
nADS_BufferSize	INT	25535	Default 24 kByte bu
ePrio_LifeSafety	E_BACnet_Priority	E_BACnet_Priority.eP1	
ePrio_Critical	E_BACnet_Priority	E_BACnet_Priority.eP5	
ePrio_ManLocal	E_BACnet_Priority	E_BACnet_Priority.eP7	
ePrio_ManOperator	E_BACnet_Priority	E_BACnet_Priority.eP8	
ePrio_Program	E_BACnet_Priority	E_BACnet_Priority.eP15	
ePrio_Default	E_BACnet_Priority	E_BACnet_Priority.eDefault	No priority used
eRemotePrio_Default	E_BACnet_Priority	E_BACnet_Priority.eP8	Default priority for p
bEvtMgmt_AckResetRequired	BOOL	TRUE	Reset is required to
sEvtMgmt_AckMsgRemote	STRING	'Acknowledged by remote user.'	Text to indicate ack
sEvtMgmt_AckMsgPLC	STRING	'Acknowledged by PLC.'	Text to indicate ack
sSeparator_ObjectName	STRING	'.'	Separator to be use
sSeparator_Description	STRING	'.'	Separator to be use
sSeparator_EventMessageText	STRING	'.'	Separator to be use
nStateText_Length	UDINT	40	Max. count of string
bStateText_WriteProtect	BOOL	FALSE	

6.2.11 Verknüpfung von Hardware über Attribute

Mit dem Attribut "TcLinkTo" können Hardware-Klemmen mit Rohwert- und Rohzustandsvariablen verbunden werden. Um den TIID-Pfad zu ermitteln, navigieren Sie zur Klemme und kopieren den Pfad.

Beispiel:

```
{attribute 'TcLinkTo' :=
  '.nRawVal := TIID^Device 2 (EtherCAT)^Term 1 (EK1200)^Term 4 (EL3214)^RTD Inputs Channel
1^Value;
  .nRawState := TIID^Device 2 (EtherCAT)^Term 1 (EK1200)^Term 4 (EL3214)^RTD Inputs Channel
1^Status' }
fbAi_IO : FB_BACnet_AI_IO := (
  fMinPresValue := -150.0,
  fMaxPresValue := 150.0
);
```

Die auf diese Weise hergestellten Verbindungen werden im System Manager als grüne Pfeilsymbole angezeigt.

6.3 Wichtige Hinweise zur Benutzung der Bibliothek

Dieses Kapitel enthält Hinweise und Empfehlungen für die Verwendung der Bibliothek bei der Programmierung eines BACnet-Servers.

6.3.1 Deklaration von Eigenschaften beim Start

Bei normaler SPS-Programmierung werden Sie wahrscheinlich Aufrufe wie diesen finden:

```
PROGRAM test
VAR
    fbBi      :      FB_BACnet_BI;
END_VAR

fbBi(
    Server:= ,
    nObjectInstance:= ,
    sObjectName:= 'Name',
    sDescription:= 'Description',
    bEventDetectionEnable:= TRUE,
    bEventAlgorithmInhibit:= ,
    bReliabilityEvaluationInhibit:= ,
    nNotificationClass:= ,
    eNotifyType:= ,
    aEventEnable:= ,
    aEventMessageTextsConfig:= ,
    nTimeDelay:= ,
    nTimeDelayNormal:= ,
    bAcknowledgeRm:= ,
    bEvent=> ,
    eRlbtty=> ,
    bOutOfService:= ,
    sInactiveText:= ,
    sActiveText:= ,
    sDeviceType:= ,
    bAlarmValue:= ,
    ePolarity:= ,
    nChangeOfStateCount:= ,
    nElapsedActiveTime:= ,
    bPresVal=> ,
    bVal:= ,
    eRlbttyPgm:= );
```

Wenn die Werte ständig von der SPS geschrieben werden, hätte das Schreiben aus dem BACnet keine Wirkung, die Werte würden sofort überschrieben werden. Daher sollten vor allem die Eigenschaften, die sich nicht oft oder gar nicht ändern, im Variablenbereich als Anfangsparameter deklariert werden. Der folgende Code zeigt ein Beispiel für die Initialisierung von Eigenschaften zum Zeitpunkt des ersten SPS-Starts.

```
PROGRAM test
VAR
    fbBI      :      FB_BACnet_BI      := (
        sObjectName := 'Name',
        sDescription := 'Description',
        bEventDetectionEnable := TRUE
    );
    bDescriptionChanged      :      BOOL;
END_VAR

-----

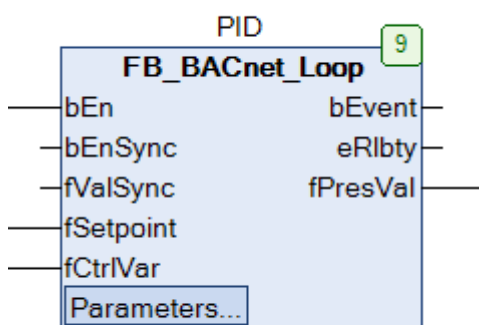
fbBI ();
```

```
IF bDescriptionChanged THEN
    bDescriptionChanged := FALSE;
    fbBI.sDescription := 'New Description';
END_IF
```

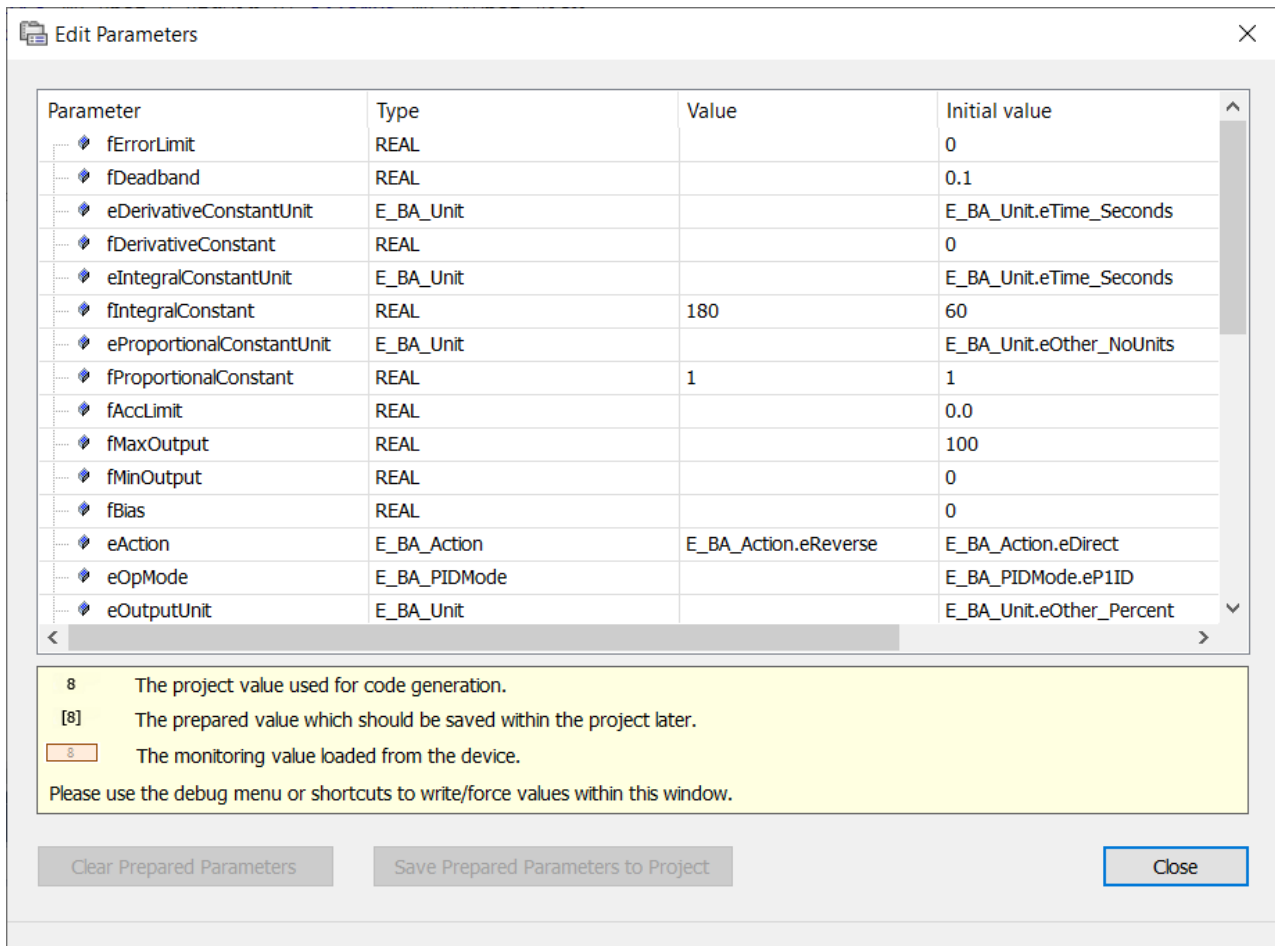
Im obigen Beispiel werden die Property-Werte (nur die, die eine Konfiguration erfordern) aus der Deklaration zum Zeitpunkt des ersten SPS-Starts übernommen. Wenn eine Konfigurationseigenschaft geändert werden muss, empfehlen wir ein Write-on-Change-Verfahren, wie hier gezeigt, und nicht zyklisches Schreiben. Außerdem würde die Implementierung eines zyklischen Schreibens dazu führen, dass keine von BACnet geschriebenen Werte mehr angenommen werden.

6.4 Parameter-Dialoge

Bei Verwendung von CFC (Continuous Function Chart) ist der Zugriff auf die Konfigurationsparameter über Parameterdialoge möglich. Öffnen Sie den Parameterdialog über die Schaltfläche in der linken Ecke.



Die in diesem Dialog angezeigten Werte können in der Spalte *Wert* geändert und mit der Funktion *Werte schreiben* im TwinCAT PLC Menü in die SPS geschrieben werden. Die Werte werden auch mit Hilfe der SPS-Persistenz persistent gespeichert.



6.4.1 BACnet-Funktionsbausteine immer zyklisch aufrufen

Es ist sehr wichtig, dass alle BACnet-Funktionsbausteine einmal pro Zyklus aufgerufen werden und nur einmal!

Rufen Sie die Funktionsbausteine **nicht** auf diese Weise auf:

```
If bCondition then
  fbAnalogInput();
END_IF
```

Dies würde ein BACnet-Objekt erzeugen, das nicht mehr aktualisiert wird, wenn bCondition FALSE ist. Das Objekt sieht für den Client, der versucht, auf das Objekt zuzugreifen, defekt aus.

Es ist jedoch möglich, ein BACnet-Objekt als Variable zu deklarieren und es gar nicht aufzurufen. In diesem Fall wird das BACnet-Objekt nicht generiert und existiert somit nicht im BACnet-Server.

6.4.2 Aufruf von BACnet-Funktionsbausteinen mit der gleichen Zykluszeit

Der Aufruf verschiedener BACnet-FBs mit unterschiedlichen Zykluszeiten führt zu Laufzeitfehlern, die durch verzögerte Bibliotheksaufrufe verursacht werden. Das BACnet-Supplement und die SPS-Bibliothek erfordern eine Synchronisierung beim Start. Es ist nicht möglich, BACnet-FBs in irgendeiner Weise verzögert aufzurufen. Wenn Objekte zur Laufzeit erstellt oder gelöscht werden müssen, lesen Sie bitte den Abschnitt Dynamic Object Manager in diesem Handbuch.

6.5 Berechnung des Routerspeichers

Der Speicher für die BACnet-Objektdarstellung in der SPS wird aus dem Routerspeicher entnommen. Die Standardeinstellung in TwinCAT ist 32 MB und muss bei einer größeren BACnet-Objektdatenbank eventuell erhöht werden. Der Log-Puffer von Trendlog- oder Eventlog-Objekten wird ebenfalls aus dem Routerspeicher entnommen.

Im Durchschnitt benötigen die Eigenschaften von BACnet-Objekten jeweils 20 KB.

Um etwas Speicherplatz für andere Zwecke zu reservieren, löst die Bibliothek einen Fehlerzustand aus, wenn mehr als 60 % des verfügbaren Routerspeichers von der BACnet-Bibliothek angefordert wird. In diesem Fall werden keine weiteren BACnet-Objekte mehr angelegt, so dass die Projektdatenbank unvollständig sein kann.

Auf Windows Embedded Compact (Windows CE) Systemen ist der maximale Routerspeicher auf 200 MB begrenzt. Unter Big-Windows (Windows 10) oder BSD hängt der Routerspeicher vom non-paged Pool ab. Wenn z. B. der gesamte non-paged Pool 1 GB beträgt, kann der Routerspeicher auf 600-700 MB erhöht werden. Bitte beachten Sie, dass die Erweiterung des Routerspeichers einen Neustart der SPS erfordern kann.

6.5.1 Beispiel für die Berechnung des erforderlichen Routerspeichers

Das folgende Beispiel zeigt eine Berechnung des erforderlichen Routerspeichers.

Anforderungsbeispiel:

280 Digital- und Analog-Objekte = $280 \times 20 \text{ KB} = 5.600 \text{ KB}$

280 Trendlog = $280 \times 20 \text{ KB} = 5.600 \text{ KB}$ (nur Eigenschaften)

Trendlog-Puffergröße (7 Tage / 10 Min. Intervall) = $1440 / 10 \times 7 = 1008$ Datensätze pro Trendlog

Speicherbedarf für Trendlog-Puffer: $280 \times 1008 \times 56 = 15.805.440$ (56 Bytes pro Datensatz im Logspeicher)

Erforderlicher Gesamtspeicher des Routers

5.600.000 Bytes Normale Objekte

5.600.000 Bytes Trendlog-Objekte

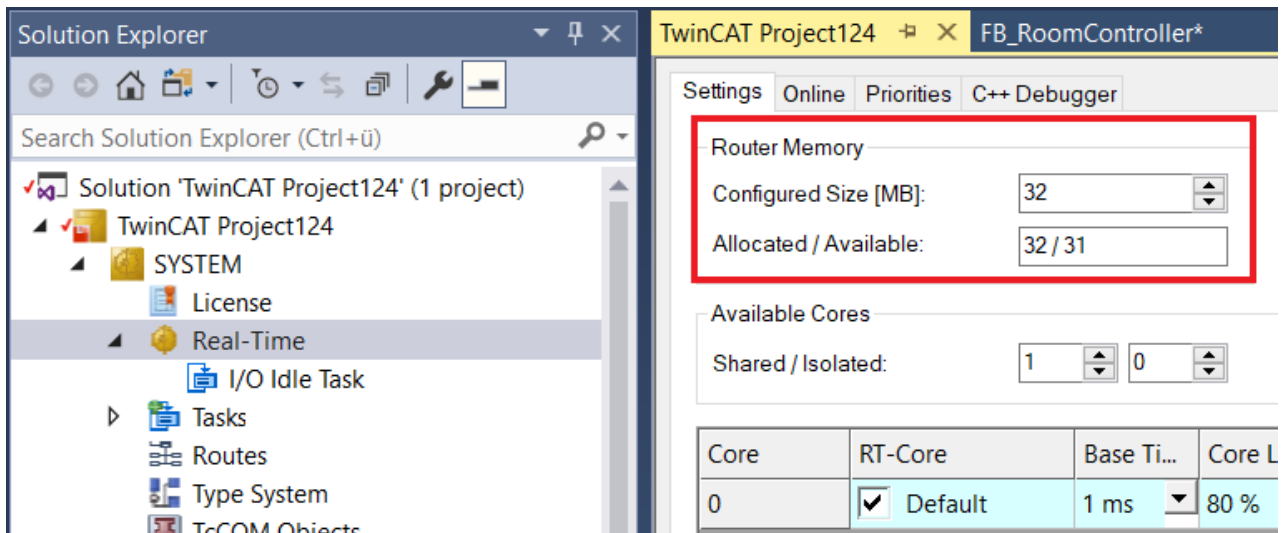
15.805.440 Bytes Trendlog-Puffer

Gesamter Routerspeicher für BACnet

27.005.440 Bytes

Bitte beachten Sie, dass der gesamte BACnet-Router-Speicher nur 60 % des gesamten benötigten Routerspeichers ausmacht, so dass der Standardwert von 32 MB für dieses Beispiel nicht ausreichen würde und erhöht werden muss.

Die Größe des Routerspeichers kann in den Echtzeit-Einstellungen von TwinCAT festgelegt werden. Bitte beachten Sie, dass der Routerspeicher erst nach einem Neustart der Steuerung zugewiesen wird.

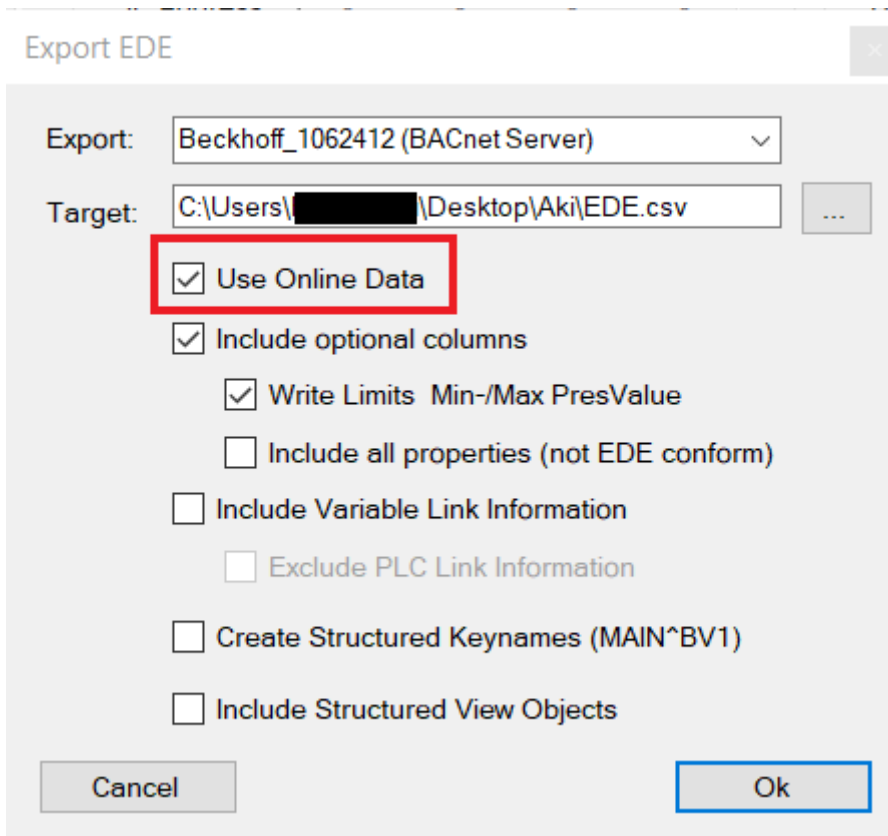


6.6 Spezifische Funktionen

Dieses Kapitel beschreibt spezifische Funktionen, die das BACnet-Supplement oder die Bibliothek verwenden.

6.6.1 Generierung einer EDE-Datei

Beim Exportieren einer EDE-Datei ist es wichtig, das Kontrollkästchen "Use Online Data" zu aktivieren (siehe unten). Aufgrund der dynamischen Erzeugung durch die BACnet Rev 14 Bibliothek müssen die Objekte zum Zeitpunkt der Erstellung der EDE-Datei sichtbar sein.



6.6.2 Zykluszeit-Überschreitungs-zähler

Zum Zeitpunkt des SPS-Starts können Überschreitungen der Zykluszeit auftreten und sind normal. Die Synchronisation zwischen dem BACnet-Supplement und der SPS kann länger dauern als die Zykluszeit. Überschreitungen sollten jedoch nicht nach der Initialisierungsphase auftreten.

6.6.3 Aktivieren / Deaktivieren von Eigenschaften

Um Eigenschaften, die für eine bestimmte Objektinstanz nicht benötigt werden, zu deaktivieren, kann das Array *"aDisabled"* der Struktur *"stSettings"* verwendet werden.

Die Struktur *"stSettings"* ist auch in den Global Params verfügbar. Die Einstellungen in den Global Params wirken sich auf alle Objekte aus.

Bitte beachten Sie, dass zusammengehörige Eigenschaften (mit der gleichen Nummer in den Fußnoten des BACnet-Standards) alle deaktiviert oder aktiviert werden müssen!

6.6.4 Schreibgeschützte Eigenschaften

Um einen Schreibschutz auf bestimmte Eigenschaften anzuwenden, kann das Array *"aWriteProtected"* der Struktur *"stSettings"* verwendet werden.

Die Struktur *"stSettings"* ist auch in den Global Params verfügbar. Die Einstellungen in den Global Params wirken sich auf alle Objekte aus.

Im Falle einer WriteProperty oder WritePropertyMultiple Anfrage zu diesen Properties wird eine Fehlermeldung *"Write_Access_Denied"* an den BACnet-Client zurückgegeben.

6.6.4.1 Beispiel Disable/Writeprotect

Das folgende Beispiel zeigt, wie mit *stSettings* Eigenschaften deaktiviert und schreibgeschützt werden.

```
fbBi : FB_BACnet_BI := (
  sObjectName := 'Example Binary Input Object',
  sDescription := 'Objectname and Description properties are read-only',
  stSettings := (
    aDisabled := [
      E_BACnetPropIdentifier.PropChangeOfStateCount,
      E_BACnetPropIdentifier.PropChangeOfStateTime,
      E_BACnetPropIdentifier.PropTimeOfStateCountReset
    ],
    aWriteProtected := [
      E_BACnetPropIdentifier.PropObjectName,
      E_BACnetPropIdentifier.PropDescription
    ]
  )
);
```

6.6.5 Freigabe des Schreibzugriffs durch die SPS

Objekte, die eine Eingangsinformation wie ACC, AI, BI usw. darstellen, liefern eine Eingangsvariable (nVal, fVal oder bVal), deren Wert ohne weitere Verarbeitung in die Eigenschaft *Present_Value* übertragen wird.

Objekte, die eine Ausgangsinformation wie AO, AV usw. darstellen, liefern eine Eingangsvariable, die den SPS-Wert (nVal, fVal, bVal) repräsentiert. Um den Wert bei der angegebenen Programmpriorität zu aktivieren, muss die Variable *bEnPgm* außerdem auf TRUE gesetzt werden. Wenn der Wert von *bEnPgm* auf FALSE gesetzt ist, wird der Wert bei der angegebenen Programmpriorität auf NULL gesetzt.

Im Falle von *_5P*-Funktionsbausteinen ist für jede der Prioritäten (mit Ausnahme von Manual Operator, der wahrscheinlich von der GLT bereitgestellt wird) ein separates Flag vorgesehen.

Die derzeit aktive Priorität kann anhand der Variablen *eActPrio* (1-16, 17=Relinquish_Default) ermittelt werden.

6.6.6 Hinzufügen eines Empfängers zu einer Empfängerliste

Die Einträge in der Eigenschaft Empfängerliste von Notification Class Objekten sind komplexe Datentypen, die zwei Optionen unterstützen. Zur Identifizierung des Empfängers kann entweder die Geräteinstanznummer oder alternativ die BACnet MAC-Adresse (physikalische Adresse) angegeben werden.

Das folgende Beispiel zeigt beide Optionen für die Variable aRecipientList:

```
aRecipientList := [
(
  stValidDays :=
    (bMonday:=TRUE, bTuesday:=TRUE, bWednesday:=TRUE, bThursday:=TRUE, bFriday:=TRUE),
  stFromTime := F_BA_ToSTime(T#0H),
  stToTime := F_BA_ToSTime(T#23H59M59S),
  stRecipient :=
    F_BACnet_DeviceRecipient(nDeviceInstance:=42),
    nProcessId := 10000,
    bIssueConfirmed := FALSE,
    stEventTransitions :=
      (bToOffNormal:=TRUE, bToFault:=TRUE, bToNormal:=TRUE)
),
(
  stValidDays := (bSunday:=TRUE, bSaturday:=TRUE),
  stFromTime := F_BA_ToSTime(T#7H),
  stToTime := F_BA_ToSTime(T#15H30M),
  stRecipient :=
    F_BACnet_EthernetRecipient(nIPAddress1:=192,168,10,200, nPort:=47808, nNetworkNr:=444),
    nProcessId := 30100,
    bIssueConfirmed := TRUE,
    stEventTransitions := (bToOffNormal:=TRUE)
)
]
```

6.6.7 Verwendung von UTF-8-Zeichen

UTF-8 ist der Standard-Zeichensatz, der in dem BACnet-Supplement verwendet wird. Der Zeichensatz UTF-8 umfasst alle Unicode-Zeichen und benötigt je nach Sprache bis zu 4 Bytes pro Zeichen.

Reguläre Zeichenketten ohne Sonderzeichen bedürfen keiner besonderen Behandlung. Latin-1-Zeichen wie äöüÄÖÜß©ÒØ usw. erfordern ebenfalls keine besondere Behandlung.

Bei kyrillischen, afrikanischen oder asiatischen Zeichen ist eine Konvertierung erforderlich. In diesem Fall muss das Attribut "TcEncoding" auf "UTF-8" gesetzt werden. Wie jedes Attribut gilt dies für die Variable unterhalb des Attributs und muss für jede Variable, die UTF-8-Zeichen enthält, wiederholt werden.

Beispiel mit UTF-8-Zeichen:

```
{attribute 'TcEncoding' := 'UTF-8'}
sMyUTF8Text : STRING := wsLiteral_TO_UTF8( "äöüßéèêµ€ Αθήναι İstanbul Київ" );
```



Bitte beachten Sie, dass Sie bei WSTRING-Variablen doppelte Anführungszeichen verwenden!

6.7 FB_BACnet_Server

Dieser Funktionsbaustein stellt den BACnet-Server in der SPS dar. Neben Informationen über die Speicherauslastung und den Speicherzustand gibt es verschiedene Möglichkeiten, z. B. Alarme zu quittieren, persistente Daten zu schreiben, den Fehlerzustand des Servers zurückzusetzen oder über die Objektdatenbank zu iterieren.



Dieser Funktionsbaustein wird automatisch aufgerufen und muss im Normalfall nicht innerhalb des SPS-Programms aufgerufen werden.

Sollten Sie auf die Eigenschaften des Servers zugreifen wollen, nutzen Sie dazu die bereits existierende Instanz aus den BACnet Globals.

6.8 FB_BACnet_Device

Dieser Funktionsbaustein kann verwendet werden, um Einstellungen des Device-Objekts zur Laufzeit zu ändern.

Beispiel:

```

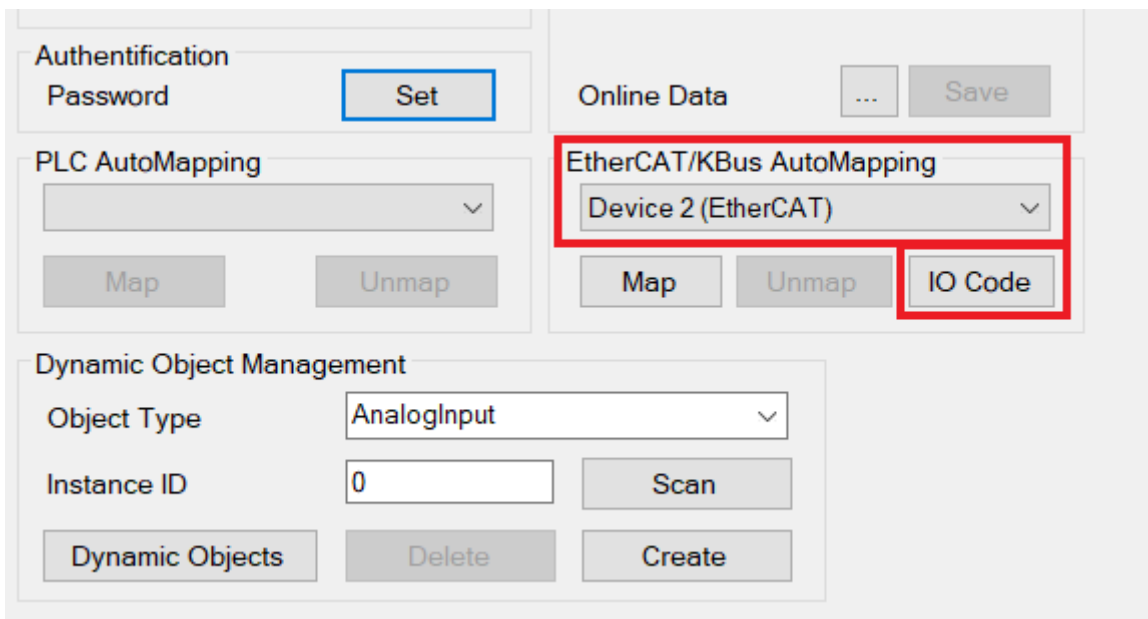
VAR
    fbDevice : FB_BACnet_Device;
    bChangeDeviceObj : BOOL;
END_VAR

-----
fbDevice();
IF bChangeDeviceObj THEN
    bChangeDeviceObj := FALSE;
    fbDevice.sObjectName := 'TEST_ObjName';
    fbDevice.sDescription := 'TEST_Description';
    fbDevice.sLocation := 'TEST_Location';
END_IF
    
```

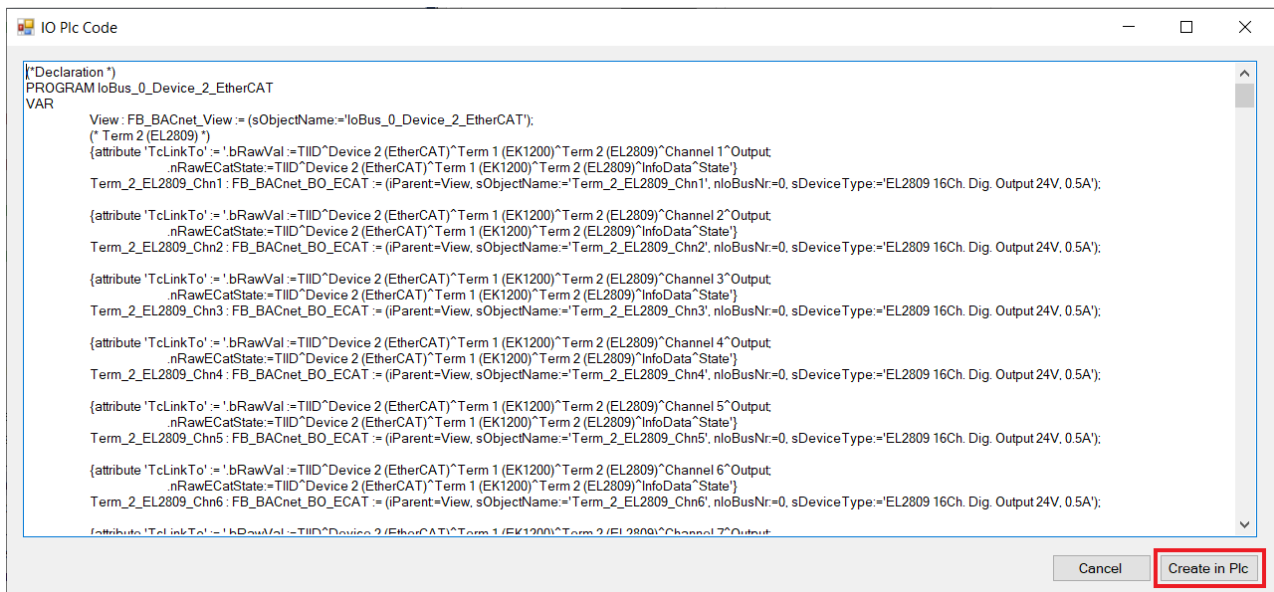
6.9 IO Code

Diese Funktion kann verwendet werden, um automatisch den notwendigen Programmcode für die Referenzierung von Hardware-Klemmen auf die SPS und BACnet-Objekte zu erzeugen. Der generierte Code wird als Programm in der SPS erstellt.

1. Um die IO-Code-Generierung zu starten, navigieren Sie zum BACnet-Server und wählen den Dialog Settings.



2. Wählen Sie die Hardware aus und verwenden Sie die Schaltfläche IO Code.
 - ⇒ Digitale und analoge Eingangs- und Ausgangsklemmen und -kanäle werden erkannt und der Programmcode wird generiert.
 - Digitale Eingangskanäle werden auf BACnet-Binary-Input-Objekte abgebildet.
 - Digitale Ausgangskanäle werden auf BACnet-Binary-Output-Objekte abgebildet.
 - Analoge Eingangskanäle werden auf BACnet-Analog-Input-Objekte abgebildet.
 - Analoge Ausgangskanäle werden auf BACnet-Analog-Output-Objekte abgebildet.



3. Mit der Schaltfläche Create in Plc wird der Code als Programm im Ordner POU/BACnet_IoBus erzeugt.
4. Rufen Sie dieses Programm zyklisch auf, z. B. aus der POU MAIN.

```
IoBus_0_Device_2_EtherCAT();
```

6.10 Empfohlener Arbeitsablauf / BACnet-Persistenz

In Projekten der Gebäudeautomation kommt es bei der Programmierung und Konfiguration der Anwendung sehr häufig zu Änderungen der Parametereinstellungen.

Es wird empfohlen, Parameter wie P, I, D oder Hi-Limit / Lo-Limit, und selbst Beschreibungstexte, etc. so lange wie möglich ohne die BACnet-Persistenz einzustellen. Als letzter Schritt vor der Übergabe der Gebäudeautomationsanwendung sollte die BACnet-Persistenz aktiviert werden.

Normalerweise werden die Werte beim Start von dem BACnet-Supplement zur SPS synchronisiert. Wenn die BACnet-Persistenz aktiviert ist, überschreiben die zuvor von BACnet geschriebenen Werte die Werte in der SPS.

Der FB_BACnet_Server stellt eine Variable elnitMode zur Verfügung, um die Richtung dieser Synchronisation zu definieren:

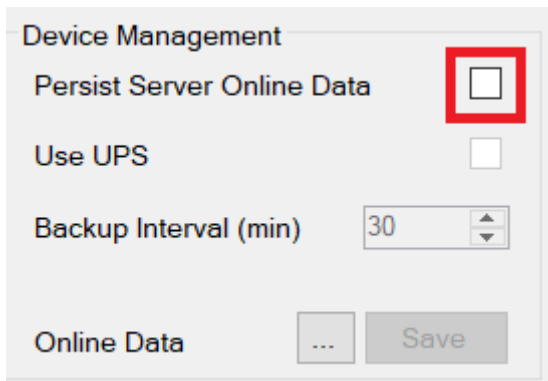
- elnitReset: Dieser Wert wird gesetzt, nachdem ein Reset to Origin durchgeführt wurde. Die BACnet-Objekte im Stack werden entfernt und der elnitMode wird auf elnitToPlc gesetzt.
- elnitToPlc (Voreinstellung): Die Werte werden von BACnet zur SPS synchronisiert.
- elnitForceFromPlc: Werte werden von der SPS auf BACnet synchronisiert.

6.11 Persistenz

Die zur Laufzeit geänderten Propertywerte werden in der Datei *BACnetOnline_1010010.bootdata* im TwinCAT-Bootordner gespeichert. Zusätzlich existiert noch eine Backupdatei mit dem Namen *BACnetOnline_1010010.bootdata-old*.

BACnet Persistenz einschalten und konfigurieren

In den Einstellungen des BACnet-Servers kann im Dialogfenster **Device Management** die Persistenz ein- und ausgeschaltet werden.



Ohne Verwendung einer unterbrechungsfreien Stromversorgung (USV) wird die Persistierung in einem konfigurierbaren Intervall durchgeführt. Dabei sollte der eingestellte Standardwert von 30 Minuten nicht unterschritten werden, da es sonst zu einer höheren Abnutzung des Flashmediums kommen könnte.

Wird eine USV verwendet, kann die Eigenschaft *Use UPS* aktiviert werden. In diesem Fall ist seitens des SPS-Programms sicherzustellen, dass die Funktionsbausteine der USV verwendet werden, um einen Ausfall der Stromversorgung zu erkennen. In diesem Fall muss seitens der SPS die Funktion zum Persistieren der Daten aufgerufen werden.

Persistierung aus dem SPS-Programm aufrufen

Zum Auslösen der Persistierung stehen zwei verschiedene Methoden zur Verfügung.

1. Setzen der Variable *bWritePersistent*:

```
BACnet_Globals.DefaultServer.bWritePersistent := TRUE;
```

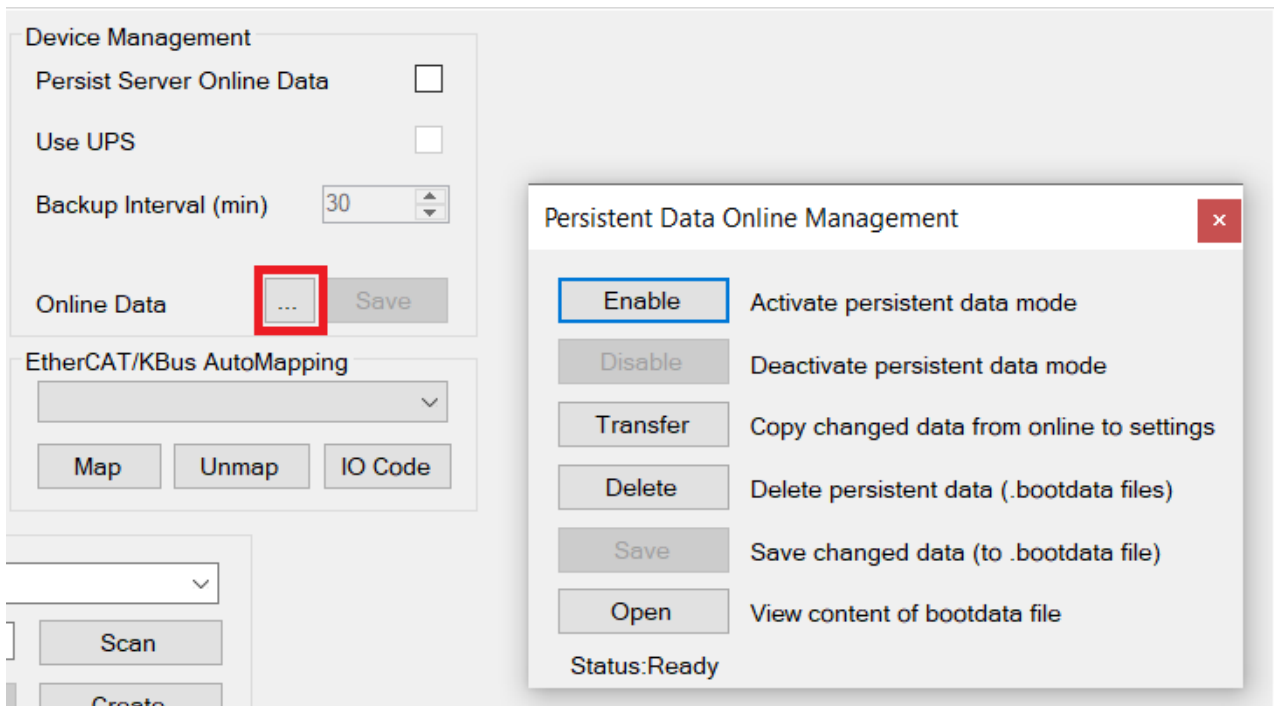
Das Rücksetzen der Variable muss ebenfalls aus dem SPS-Programm erfolgen.
2. Aufruf der Methode *SavePersistentStackData*:

```
bSuccess := BACnet_Globals.DefaultServer.SavePersistentStackData();
```

Anhand des Rückgabewertes kann erkannt werden, ob der Aufruf erfolgreich durchgeführt wurde.

Persistenzdaten anzeigen und bearbeiten

- ✓ Sie haben das Dialogfenster Device Management geöffnet.
- 1. Wählen Sie die Option **Online Data > ...** aus.
- ⇒ Es erscheint der Dialog zur Bearbeitung der Persistenzdaten, dessen Inhalte im Folgenden erläutert werden.

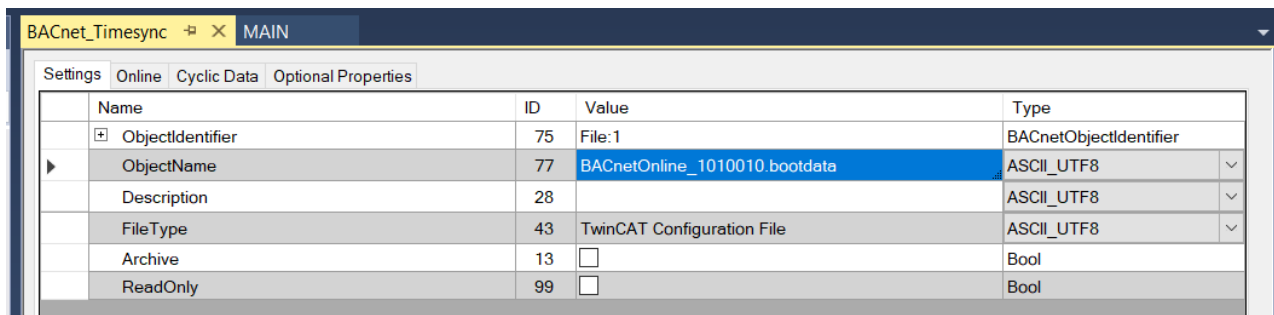


Option	Beschreibung
Enable	Persistierung einschalten.
Disable	Persistierung ausschalten.
Transfer	Ermöglicht das Kopieren der persistenten Daten in die Standardeinstellungen der Objekte (Settings).
Delete	Löscht die Dateien, welche die Persistenzdaten enthalten.
Save	Speichert geänderte Daten in der Persistenzdatei.
Open	Kann eine Persistenzdatei laden und anzeigen.

Auf die Persistenzdatei mit dem FILE-Objekt zugreifen

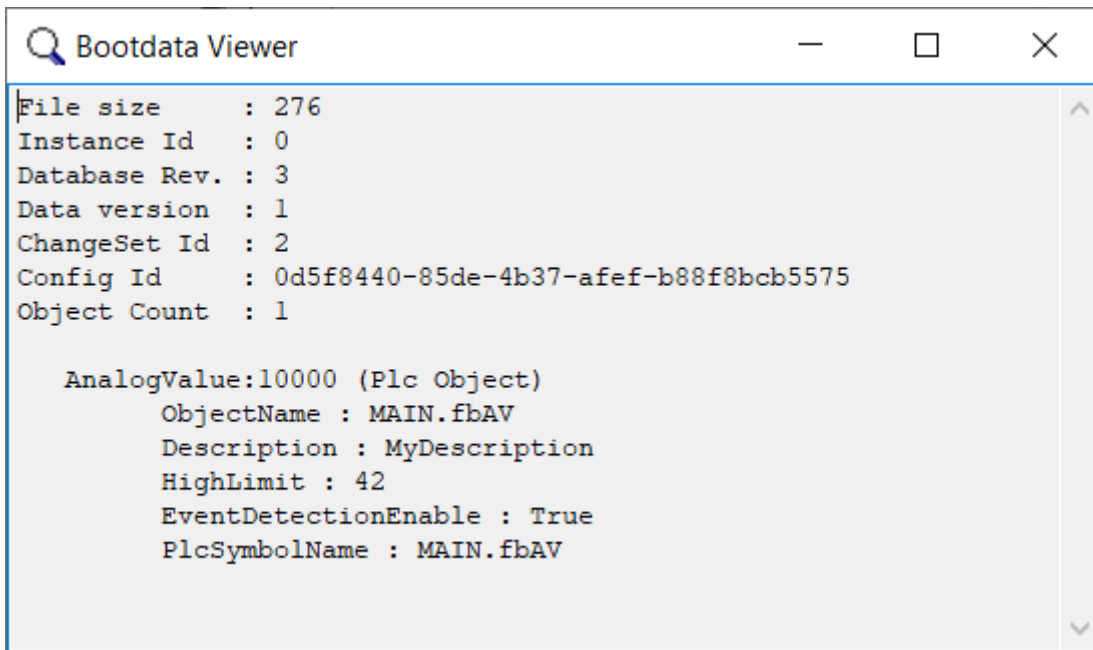
Die Persistenzdatei wird über BACnet innerhalb des Servers durch das File-Objekt mit der Instanznummer 1 repräsentiert.

1. Klicken Sie mit einem Doppelklick auf das Feld *Value* des ObjectName, um den aktuell vorhandenen Inhalt der Persistenzdatei ansehen zu können.
- ⇒ Sofern die Persistenzdatei Daten enthält, wird anschließend der Bootdata Viewer aufgerufen und die geänderten Properties werden angezeigt.



Bootdata Viewer

Im Fenster **Bootdata Viewer** werden die geänderten Properties aus der Persistenzdatei angezeigt und können von dort auch für die Weiterverarbeitung markiert und kopiert werden.



```
Bootdata Viewer
File size      : 276
Instance Id    : 0
Database Rev.  : 3
Data version   : 1
ChangeSet Id   : 2
Config Id      : 0d5f8440-85de-4b37-afef-b88f8bcb5575
Object Count   : 1

AnalogValue:10000 (Plc Object)
  ObjectName   : MAIN.fbAV
  Description  : MyDescription
  HighLimit    : 42
  EventDetectionEnable : True
  PlcSymbolName : MAIN.fbAV
```

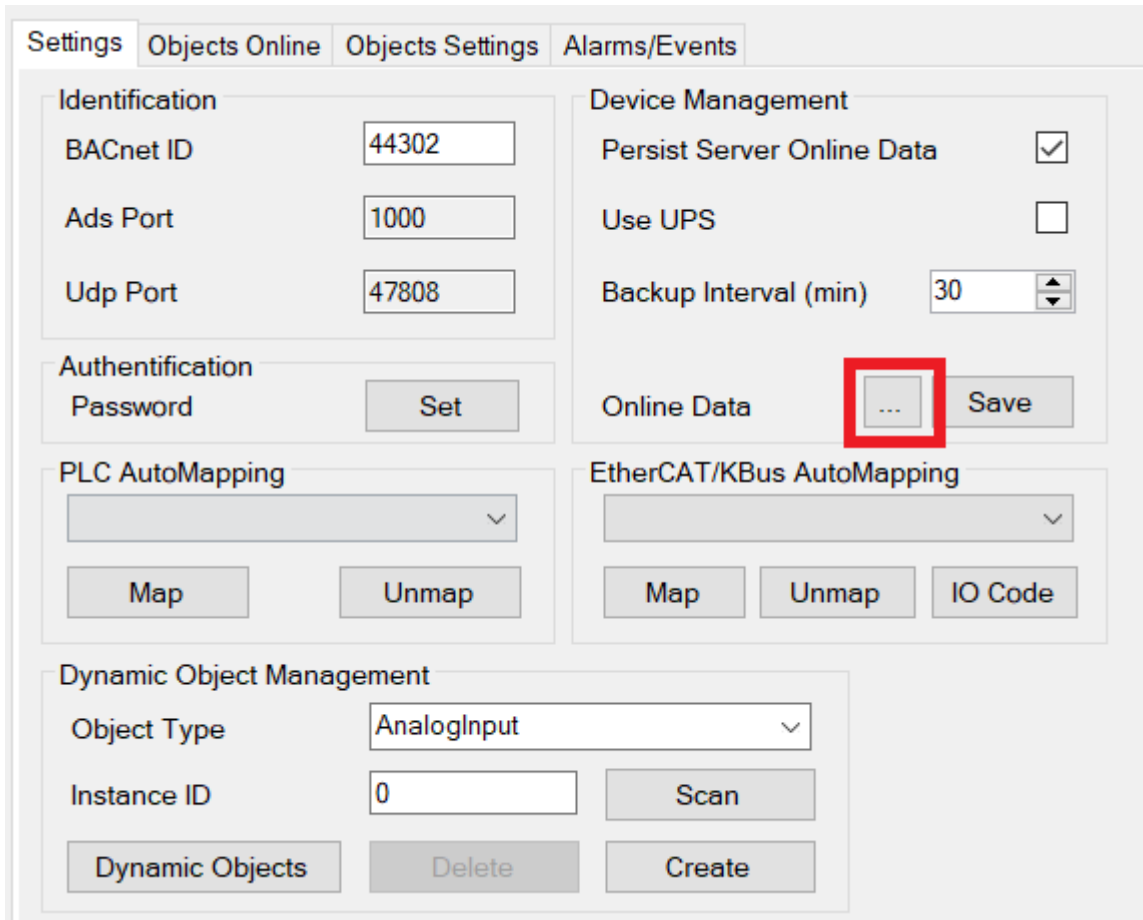
In diesem Beispiel wurden im Objekt Analog Value mit der Instanznummer 10000 die Properties `ObjectName`, `Description`, `HighLimit`, `EventDetectionEnable` und `PlcSymbolName` geändert.

Persistenzdaten löschen

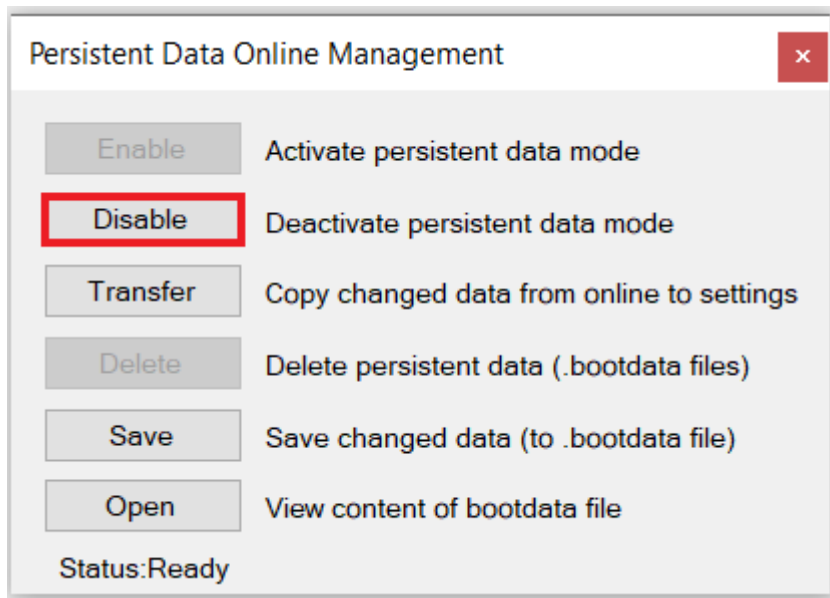
Dialog Online Data aufrufen

1. Öffnen Sie im BACnet-Server die Registerkarte **Settings**.

2. Klicken Sie den Menüpunkt **Online Data > ...**

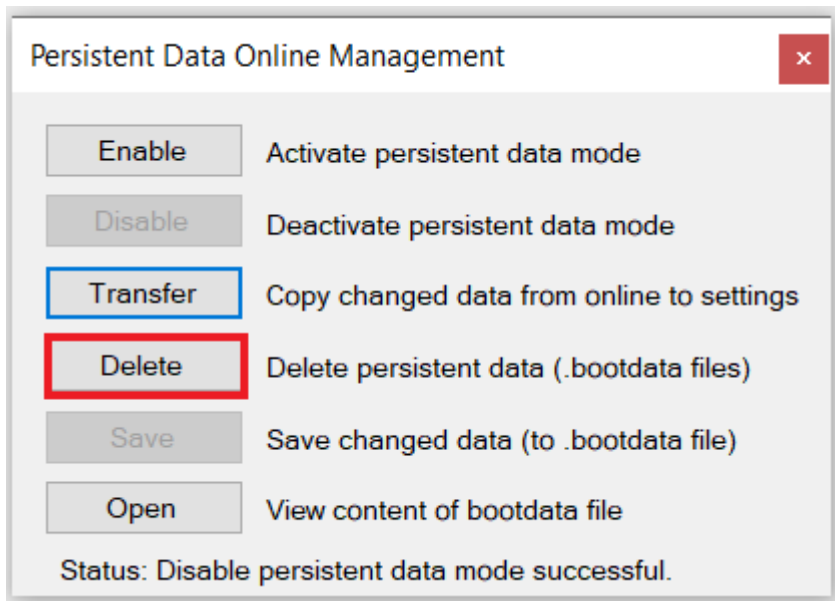


⇒ Es öffnet sich das Dialogfenster **Persistent Data Online Management**.



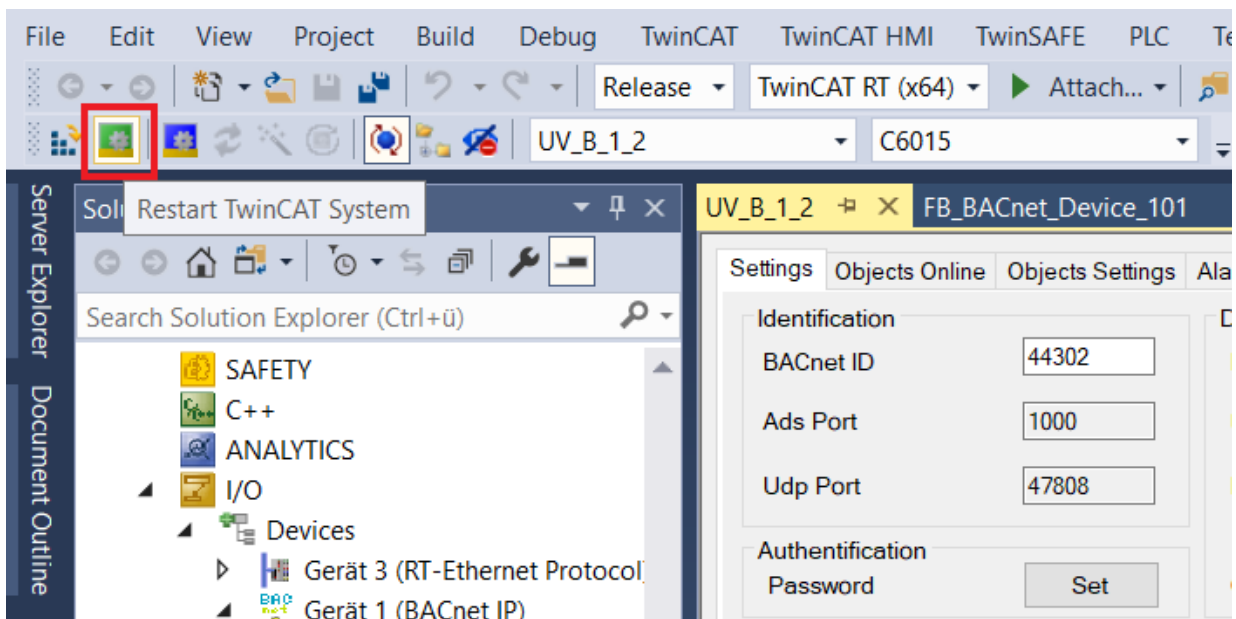
3. Klicken Sie den Button **Disable**.

4. Klicken Sie anschließend im selben Menü den Button **Delete**.



5. Beachten Sie die Statusanzeige.
Diese gibt Auskunft, ob die Funktion erfolgreich ausgeführt werden konnte.

6. Machen Sie zum Abschluss einen Restart über das TwinCAT-Hauptmenü, in dem Sie den Button Runtime-Mode anklicken.



6.12 Uhrzeitsynchronisation

In dieser Beschreibung steht der Begriff *Uhrzeit* für die Kombination aus Datum und Uhrzeit (Date und Time).

BACnet Uhrzeitsynchronisation

Die Uhrzeitsynchronisation erfolgt bei BACnet über zwei Dienste:

TimeSynchronization synchronisiert die Uhrzeit auf Basis der lokalen Uhrzeit, d. h. Zeitgeber und Zeitempfänger müssen sich in derselben Zeitzone befinden.

Bei der Verwendung von *UTCTimeSynchronization* (UTC=Universal Time Coordinated) wird die Uhrzeit als GMT (Greenwich Mean Time = Ortszeit London) übertragen. Der Zeitgeber rechnet vor dem Senden die Uhrzeit in GMT um. Anhand der lokalen Einstellungen *UTC_Offset* und *DaylightSavingsStatus* (im Device Objekt des Servers) wird dann vom Zeitempfänger wieder auf die lokale Zeit umgerechnet.

UTC_Offset gibt den Wert in Minuten relativ zu GMT an. Ein positiver Wert ist westwärts von GMT, ein negativer Wert ostwärts von GMT zu verwenden.

Westeuropa hat einen Offset von -60 Minuten zu GMT. Dies ist als Defaultwert im TwinCAT BACnet-Server festgelegt und muss ggf. bei Einsatz in anderen Zeitzonen mit Verwendung von UTC angepasst werden.



- Die Uhrzeitsynchronisation hat eine direkte Auswirkung auf die Aufzeichnung von BACnet Logging Objekten (Trendlog, Trendlog-Multiple und Eventlog). Wenn sich die Uhrzeit ändert, wird dies als Eintrag in den Logspeicher übernommen. Bei kurzen Uhrzeitsynchronisationsintervallen kann es daher zu vielen (unnötigen) Einträgen im Logspeicher der Logging Objekte kommen.
 - In einem BACnet-Netzwerk sollte nur ein einziger Uhrzeitgeber existieren. Dies ist seitens des Betreibers bzw. Fachplaners sicherzustellen. Auch wenn es nach dem BACnet-Standard grundsätzlich möglich ist, die Uhrzeitsynchronisation nur an eine Gruppe von Teilnehmern (Multicast) oder an nur einen einzelnen Teilnehmer (Unicast) zu senden, wird die Uhrzeit in der Praxis häufig als Broadcast an *alle* Teilnehmer des Netzwerks gesendet.
-

TwinCAT als BACnet Zeitempfänger

Empfängt ein TwinCAT Controller eine BACnet Uhrzeitsynchronisation, wird die interne Uhrzeit des Controllers mit dieser Uhrzeit gestellt.

TwinCAT als BACnet Zeitgeber

Wird ein TwinCAT Controller als BACnet Zeitgeber verwendet, muss dieser die exakte Uhrzeit von einer externen Stelle ermitteln, z. B. durch Verwendung des NTP-Dienstes auf einen externen NTP-Server (NTP = Network Time Protocol). Als öffentlich verfügbarer Zeitserver steht z. B. time.windows.com zur Verfügung.

Für die Übernahme der Zeit und ggf. Aussendung im Netzwerk steht die Methode `TimeSyncEx` im `FB_BACnet_Adapter` zur Verfügung.

Als Parameter werden folgende Werte an den Aufruf übergeben:

Name	Typ	Beschreibung
pDateTime	ST_BA_DateTime	Pointer auf eine Struktur vom Typ ST_BA_DateTime (siehe Bibliothek Tc3_BA2_Common). In dieser Struktur wird das zu setzende Datum und die Uhrzeit übergeben bzw. im Falle von eSyncMode = 3 zurück geliefert.
eSyncMode	E_BACnet_TimeSyncMode	<p>Enumeration Der Wert bestimmt die Art der Synchronisation:</p> <p>eSyncAndSendLocalTime := 0 Setzt pDateTime in den BACnet Stack (als Local Time) und sendet die Zeit als Broadcast (TimeSync request) Anwendungsfall: Der Controller erhält per NTP die Uhrzeit und sendet die lokale BACnet Zeit an andere Controller als Broadcast.</p> <p>eSyncOnly := 1 Setzt pDateTime in den BACnet Stack (als Local Time) Anwendungsfall: Alle Controller erhalten per NTP die Uhrzeit und synchronisieren ihre eigene BACnet Uhr.</p> <p>eSyncAndSendUtcTime := 2 Setzt pDateTime in den BACnet Stack (als UTC Time) und sendet die Zeit als Broadcast (UTC Time Sync request) Anwendungsfall: Der Controller erhält per NTP die Uhrzeit und sendet die BACnet Zeit als UTC an andere Controller als Broadcast.</p> <p>eSyncGetTime := 3 Gibt in pDateTime die aktuelle Zeit im BACnet Stack zurück. Anwendungsfall: Wird eher sehr selten verwendet. Ein möglicher Anwendungsfall wäre z. B. ein Vergleich der BACnet Uhr mit der Trendaufzeichnung.</p>

6.13 Empfohlene Zykluszeit der PLC-Task

Der Standardwert für die SPS-Zykluszeit (PLC-Task) beträgt 10 ms. Dies ist in vielen Fällen für die Verarbeitung der BACnet-Daten nicht notwendig.



Es wird empfohlen, eine Zykluszeit von 40-50 ms für die PLC-Task zu verwenden.

Noch höhere Werte (d. h. eine langsamere Zykluszeit) sind nicht empfehlenswert, da ansonsten die Kommunikation über BACnet, z. B. mit einer MBE (Management Bedieneinrichtung, früher GLT=Gebäudeleittechnik genannt), verlangsamt wird.

7 Programmierung eines BACnet-Clients

Ein BACnet-Client ist ein Verweis auf ein externes Peer-BACnet-Device. Als Client ermöglicht das TwinCAT BACnet Supplement den Zugriff auf Objekte und Services, die von einem externen Gerät bereitgestellt werden.

Für jedes externe Gerät muss ein Bezug hergestellt werden.

Die Funktionsbausteine, um als Client auf BACnet-Objekte zuzugreifen, befinden sich im Abschnitt POU's/Remote/Objects.

Unterschied zwischen Server- und Client-FBs

Client-FBs sind für den zyklischen Datenaustausch mit einem externen Gerät vorgesehen, wobei hauptsächlich der Present Value und die Status Flags übertragen werden. Andere Eigenschaften wie Beschreibung, Grenzen, Zustandstexte, Minimal-/Maximalwert des Present Value usw. sind nicht Teil der Standard-FB-Implementierung. Dennoch können die Funktionsbausteine FB_BACnet_ReadProperty und FB_BACnet_WriteProperty verwendet werden, um auf andere Eigenschaften als die in der FB-Implementierung vorgesehenen Standard-Eigenschaften zuzugreifen. Diese Funktionsbausteine bieten azyklischen Zugriff auf BACnet-Property-Daten.

7.1 Schreibmodus

Das Schreiben auf das Fremdgerät ist standardmäßig als `WriteOnChange` implementiert. Das bedeutet, dass Werte nur geschrieben werden, wenn sich der Wert geändert hat. In den Client-FBs kann ein periodisches Schreiben eingestellt werden.

Der aufgezählte Typ `E_BACnet_WriteMode` spezifiziert den Schreibzugriff auf Client-Eigenschaften:

`eAuto` = Automatisch, bei `TRUE` wird der Modus aus den Eigenschaften des Clientgerätes bestimmt

`eCyclic` = Zyklisches Schreiben, Zykluszeit wird aus `tWriteCycleTime` entnommen

`eOnChange` = Schreiben wird nur ausgelöst, wenn sich der Wert geändert hat

7.2 Lesemodus

Die Verwendung der Remote-Funktionsbausteine als Client erfordert die Angabe des Zugriffsmodus auf die Peer-Objekte. Die Variable `eReadMode` vom Typ `E_BACnet_CommMode` gibt die Zugriffsmethode an.

`eAuto` = Automatisch

`eCov` = Unconfirmed COV

`eCovC` = Confirmed COV

`eCovU` = Unsubscribed COV

`eRp` = Read Property

`eRpm` = RPM=Read Property Multiple

7.2.1 Automatikbetrieb

Die folgende Beschreibung erläutert die Implementierung des automatischen Lesemodus. Während des Starts im automatischen Modus werden wichtige Eigenschaften wie `ProtocolServicesSupported`, `ApduSize` aus dem Peer-Gerät gelesen. Anhand dieser Property-Daten wird der Lesemodus nach den folgenden Regeln berechnet. Es wird auch berechnet, wie viele Eigenschaften in einer `Rpm`-Anfrage optimal sind, um eine Segmentierung zu vermeiden.

7.2.1.1 Fall 1: Geräte mit kleiner APDU-Größe

MS/TP oder BACnet over LonTalk Geräte unterstützen nur kleine APDU (Application Protocol Data Unit) Größen. MS/TP unterstützt maximal 480 Bytes, BACnet over LonTalk nur bis zu 206 Bytes.

Die Höchstzahl der parallelen Anfragen ist auf 1 festgelegt, d. h. es wird nur 1 Anfrage gesendet und die Antwort abgewartet, bevor die nächste Anfrage gestellt wird.

Wenn die APDU-Größe kleiner oder gleich 480 Bytes ist (z. B. MS/TP-Geräte), dann wird der ReadPropertyMultiple-Schwellenwert auf 5 gesetzt. Change-of-Value wird in diesem Fall nicht verwendet.

Die Anzahl der Eigenschaften pro RPM-Anfrage ist auf 20 begrenzt.

Wenn die APDU-Größe kleiner oder gleich 206 Bytes ist (BACnet over LonTalk-Geräte), ist die Anzahl der Eigenschaften pro RPM-Anfrage auf 10 Eigenschaften begrenzt.

7.2.1.2 Fall 2: BACnet/IP-Geräte

BACnet/IP-Geräte unterstützen eine APDU-Größe von bis zu 1476 Bytes.

Wenn die APDU-Größe größer als 480 Bytes ist (z. B. 1476 für BACnet/IP), dann wird die Anzahl der parallelen Anfragen auf 50 gesetzt, d.h. es können bis zu 50 Anfragen auf die Antworten warten.

Der ReadPropertyMultiple-Schwellenwert wird auf 300 gesetzt, d.h. wenn mehr als 300 Eigenschaften zyklisch angefragt werden, wird RPM anstelle von COV verwendet.

7.2.1.3 ReadPropertyMultiple-Schwellenwert und ausgewählter Lese-Service

Wenn die Gesamtzahl der in einem Zyklus zu lesenden Eigenschaften den ReadPropertyMultiple-Schwellenwert überschreitet und ReadPropertyMultiple vom Peer-Gerät unterstützt wird, wird der Lesemodus auf ReadPropertyMultiple gesetzt.

Wenn die Gesamtzahl der in einem Zyklus zu lesenden Eigenschaften kleiner oder gleich dem ReadPropertyMultiple-Schwellenwert ist und ChangeOfValue vom Peer-Gerät unterstützt wird, wird der Lesemodus auf ChangeOfValue gesetzt.

Wenn die Gesamtzahl der in einem Zyklus zu lesenden Eigenschaften kleiner oder gleich dem ReadPropertyMultiple-Schwellenwert ist und ChangeOfValue vom Peer-Gerät nicht unterstützt wird, wird der Lesemodus auf ReadPropertyMultiple gesetzt.

Wenn die Gesamtzahl der in einem Zyklus zu lesenden Eigenschaften kleiner oder gleich dem ReadPropertyMultiple-Schwellenwert ist und ReadPropertyMultiple vom Peer-Gerät nicht unterstützt wird, wird der Lesemodus auf ReadPropertyMultiple gesetzt.

7.2.2 Anwenden des Lesemodus auf das gesamte Peer-Gerät

Der Zugriffsmodus kann für die gesamte Clientverbindung angegeben werden (d. h. diese Einstellung wird auf alle Objekte des Peer-Geräts angewendet).

```
Client : FB_BACnet_Client :=
(eReadMode:=E_BACnet_CommMode.eCovU,bSuppCov:=TRUE,bSuppCovP:=TRUE,bSuppRpm:=TRUE,tReadCycleTime:=T#
2S550ms,tWriteCycleTime:=T#2S550ms);
```

7.2.3 Anwenden des Lesemodus auf einzelne Objekte

Geräte verschiedener Hersteller unterstützen zwar Change-of-Value, aber nicht für alle Objekte in einem Gerät. In diesem Fall schlägt der Versuch, COV auf alle Objekte anzuwenden, fehl. Wenn man weiß, welche Objekte COV unterstützen (aus dem Gerätehandbuch oder aus anderen Quellen wie dem PICS-Dokument (Protocol Implementation Conformance Statement)), kann man einen COV-Zugang zu den Objekten einrichten, die diese Funktion unterstützen.

Beispiel: Das Objekt AV:2 wird im Lesemodus COVC (Confirmed COV) angefordert.

```
Actual_Flow_Rate_feedback : FB_BACnetRM_AV :=
(Client:=Client,nObjectInstance:=2,eReadMode:=E_BACnet_CommMode.eCovC );
```




Für den Fall, dass dem Peer-Gerät der Speicher ausgeht und es keine weiteren COV-Abonnements annehmen kann, ist kein Rückfallverfahren vorgesehen. In diesem Fall geht der FB in einen Fehlerzustand über. Falls Sie unsicher sind, verwenden Sie die RP/RPM-Abfrage anstelle von COV.

7.2.4 COV-Reporting

COV (Change-of-Value) Reporting bietet die Möglichkeit, automatisch Benachrichtigungen bei Wertänderungen zu erhalten. Normalerweise werden nur die Eigenschaften `Present_Value` und `Status_Flags` gemeldet. Um als Kunde COV-Benachrichtigungen zu erhalten, muss er für einen bestimmten Zeitraum ein COV-Abonnement abschließen (auch wenn unendlich viele Abonnements möglich sind, ist es sinnvoll, nur begrenzte Laufzeiten zu verwenden). Wenn das Abonnement erfolgreich war, wird vom Server eine einfache Bestätigung zurückgegeben. Der aktuelle Wert wird als Anfangswert gemeldet, sobald das Gerät ihn liefern kann (normalerweise innerhalb weniger Sekunden).

Die Liste der aktiven COV-Abonnements kann über die Eigenschaft `Active_COV_Subscriptions` des Device-Objekts abgerufen werden.

Wenn das Abonnement nach Ablauf der Lebensdauer des Abonnements nicht wiederholt wird, wird es stillschweigend entfernt und es werden keine COV-Meldungen mehr für dieses Abonnement gemeldet.

Nach der Erneuerung des Abonnements wird der aktuelle Wert wieder als Anfangswert zur Verfügung gestellt, was insbesondere bei Binary-Objekten, die sich nicht sehr oft ändern, als Heartbeat verwendet werden kann.

Die Anzahl der Abonnements ist in der BACnet-Norm nicht eindeutig festgelegt. Außerdem gibt es Geräte auf dem Markt die COV unterstützen, aber nicht für alle Objekte. Oder die Anzahl der Abonnements kann auf einige wenige pro Objekt oder pro Gerät beschränkt sein. Andere Client-Prozesse wie die GLT können ebenfalls COV verwenden, was dazu führen kann, dass ein BACnet-Device keinen Platz mehr für weitere Abonnements hat.

Im Falle von BACnet-Fehlermeldungen, insbesondere bei kleineren MS/TP-Geräten, empfehlen wir die Verwendung einer `ReadProperty` / `ReadProperty Multiple` Abfrage anstelle von COV.

7.3 Client POU's

Remote-FBs zur Programmierung eines BACnet-Clients tragen den Namen "FB_BACnetRM_", gefolgt von dem in der folgenden Tabelle angegebenen Kürzel, z. B. `FB_BACnetRM_AI` repräsentiert einen Remote-FB für den Zugriff auf ein analoges Eingangsobjekt in einem Peer-Gerät.

ACC	Accumulator	Dieser Objekttyp stellt akkumulierte (Impuls-)Werte dar.
AI	Analog Input	Dieser Objekttyp repräsentiert physikalische analoge Eingangsinformationen, z. B. einen Sensorwert.
AO	Analog Output	Dieser Objekttyp repräsentiert physikalische analoge Ausgangsinformationen, z. B. über einen 0-10V-Ausgang.
AV	Analog Value	Dieser Objekttyp repräsentiert eine (virtuelle) Analogwert-Information, z. B. einen Sollwert.
AV_Displ	Analog Value Display	Dieser Objekttyp stellt eine schreibgeschützte Analogwert-Information dar.
AV_Setp	Analog Value Setpoint	Dieser Objekttyp stellt eine beschreibbare, aber nicht kommandierbare Analogwert-Information dar, Priority-Array und Relinquish-Default sind nicht vorgesehen.
AVG	Averaging	Dieser Objekttyp stellt ein Averaging-Objekt dar, das statistische Informationen liefert.
BI	Binary Input	Dieser Objekttyp repräsentiert eine binäre Eingangsinformation, z. B. den Zustand einer Lampe oder einer Sicherung.
BO	Binary Output	Dieser Objekttyp repräsentiert eine binäre Ausgangsinformation, z. B. über einen Schalter.
BV	Binary Value	Dieser Objekttyp repräsentiert eine (virtuelle) Binärwert-Information, z. B. einen Fehlerzustand.
BV_Displ	Binary Value Display	Dieser Objekttyp stellt eine schreibgeschützte Binärwert-Information dar.
BV_Setp	Binary Value Setpoint	Dieser Objekttyp stellt eine beschreibbare, aber nicht kommandierbare Binärwert-Information dar, Priority-Array und Relinquish-Default sind nicht vorgesehen.
CAL	Calendar	Dieser Objekttyp stellt kalendarische (datumsbasierte) Informationen dar.
CMD	Command	Dieser Objekttyp stellt Befehlsinformationen (Szeneninformationen) dar.
Device	Device	Dieser Objekttyp stellt das physische Gerät dar. Sie enthält Informationen wie die lokale Uhr, den Hersteller, den Modellnamen und mehr.
EE	EventEnrollment	Dieser Objekttyp wird verwendet, um die Ereignisüberwachung zusätzlich zum Intrinsic Reporting anzuwenden, z. B. um Warngrenzen zu implementieren.
ELOG	Eventlog	Dieser Objekttyp repräsentiert einen Eventlog-Puffer, z. B. um Alarme lokal zu speichern.
File	File	Dieser Objekttyp repräsentiert Dateien, z. B. die aktuelle Konfiguration oder persistente Daten.
Group	Group	Dieser Objekttyp stellt eine Gruppe von Objekten dar.
Loop	Control Loop	Dieser Objekttyp repräsentiert Regelkreise, z. B. einen PI- oder PID-Kreis.
Loop_Cnv	Control Loop	Wie Loop-Objekt, zusätzlich sind P, I und D sowie die jeweiligen Einheiten vorgesehen.
MI	Multistate Input	Dieser Objekttyp repräsentiert eine physikalische Multistate-Input-Information, z. B. einen lokalen Betriebsartenschalter.
MO	Multistate Output	Dieser Objekttyp repräsentiert eine physikalische Multistate-Output-Information, z. B. einen von der SPS gesteuerten Betriebsartenschalter.
MV	Multistate Value	Dieser Objekttyp stellt eine (virtuelle) Multistate-Wert-Information dar, z. B. einen Programmparameter.
MV_Displ	Multistate Value Display	Dieser Objekttyp stellt eine schreibgeschützte Multistate-Wert-Information dar.
MV_Setp	Multistate Value Setpoint	Dieser Objekttyp stellt eine beschreibbare, aber nicht kommandierbare Multistate-Wert-Information dar, Priority-Array und Relinquish-Default sind nicht vorgesehen.
NC	Notification Class	Dieser Objekttyp stellt eine Alarmklasse zur Benachrichtigung von Empfängern dar.
PC	Pulse Converter	Dieser Objekttyp stellt eine umgewandelte Impulsinformation dar, z. B. den Energieverbrauch in kWh.

Prog	Program	Dieser Objekttyp stellt das SPS-Programm dar.
SchedA	Schedule Analog	Dieser Objekttyp stellt einen Zeitschaltplan mit Analogwerten dar.
SchedB	Schedule Binary	Dieser Objekttyp stellt eine Liste von Binärwerten dar.
SchedM	Schedule Multistate	Dieser Objekttyp stellt einen Zeitschaltplan mit Multistate-Werten dar.
TLM	Trendlog Multiple	Dieser Objekttyp stellt ein Trendlog-Objekt dar, das mehrere Kanäle unterstützt.
Tlog	Trendlog	Dieser Objekttyp stellt ein Trendlog-Objekt dar, das einen einzelnen Kanal unterstützt.

7.4 FB Code

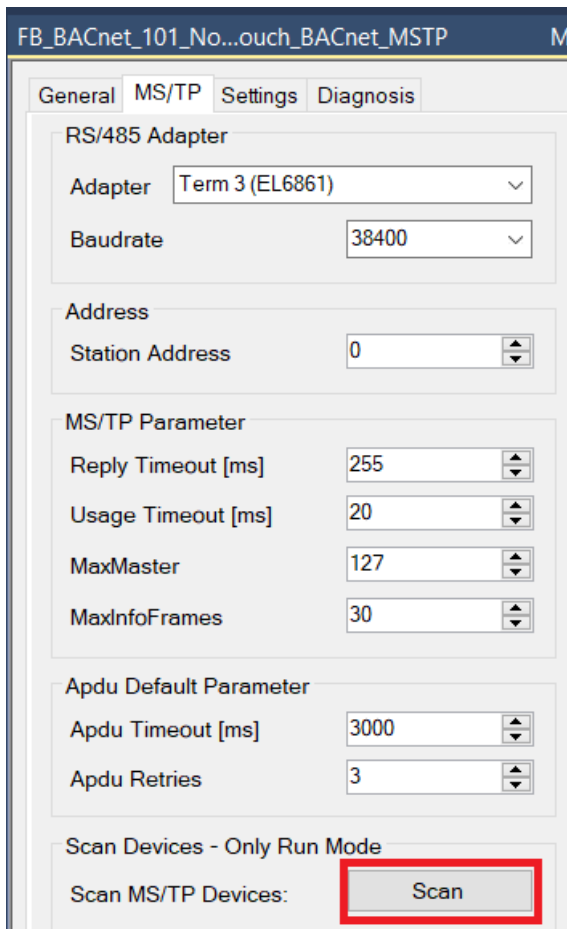
Diese Funktion kann verwendet werden, um automatisch den notwendigen Funktionscode für die Referenzierung von BACnet-Clients in der SPS zu erzeugen. Die Remote-Funktionsbausteine, die sich auf die BACnet-Remote-Objekte beziehen, werden bei der Codegenerierung automatisch referenziert.

7.5 Aufrufen der Funktion FB Code

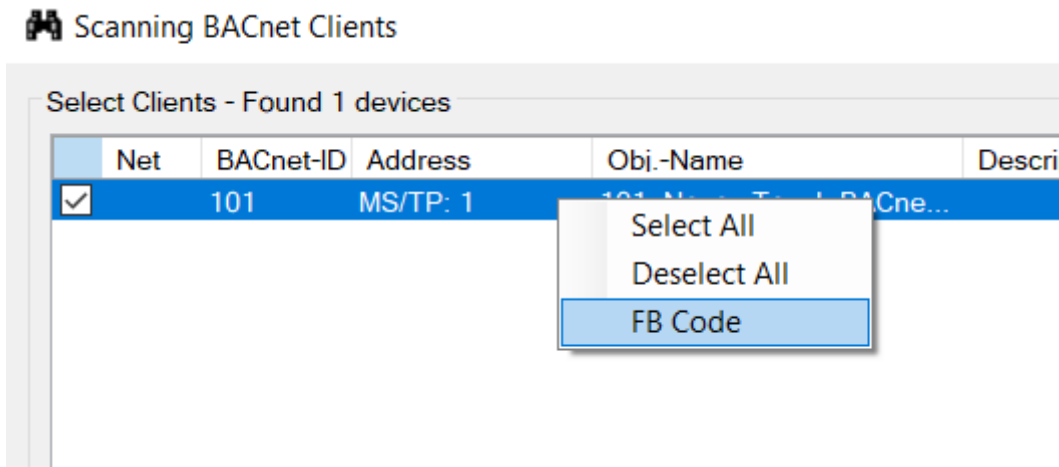
FB Code kann über verschiedene Menüpunkte im System Manager aufgerufen werden.

7.5.1 Aufruf von FB Code aus dem Dialog Scan

Mit der Funktion *Scan* werden BACnet-Devices automatisch über die BACnet-Services Who-Is / I-Am erkannt.



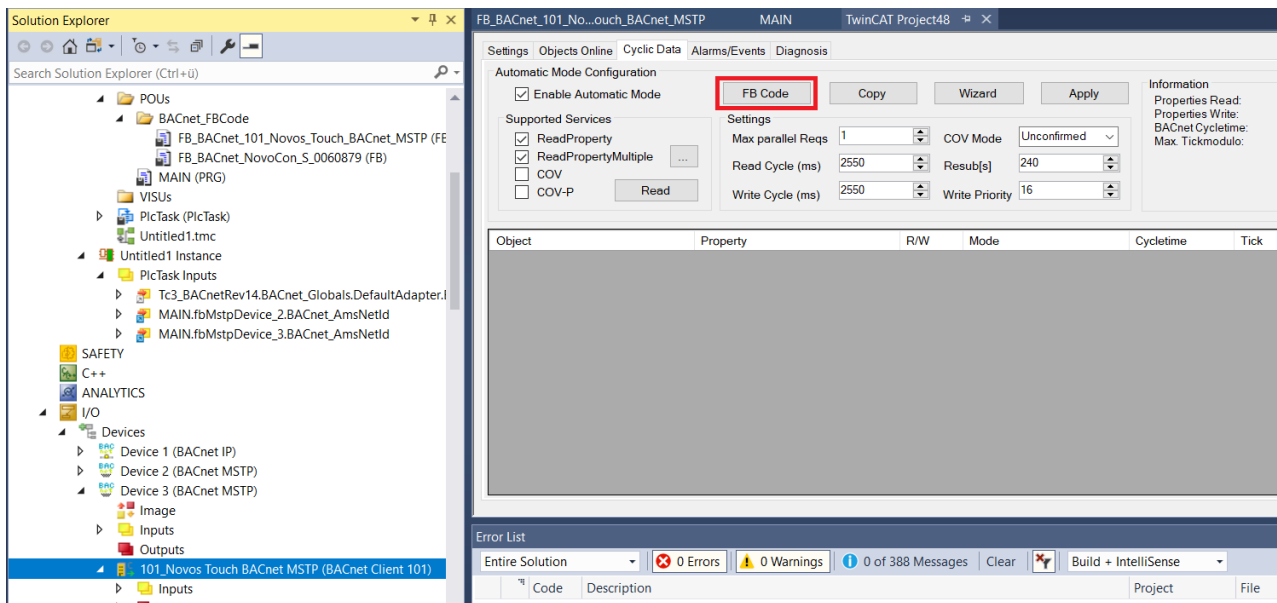
Nach der Suche nach Geräten zeigt der Dialog alle im Netzwerk gefundenen verfügbaren Geräte an. Klicken Sie mit der rechten Maustaste auf ein Gerät und wählen Sie **FB Code**, um die Codegenerierung zu starten. Vergewissern Sie sich, dass bereits ein SPS-Projekt existiert.



7.5.2 Aufruf von FB Code aus dem Dialog Cyclic Data

Wurde das Gerät gescannt und als BACnet-Client-Referenz zum System Manager hinzugefügt, kann der FB Code über den Dialog *Cyclic Data* aufgerufen werden.

Wählen Sie das Client-Gerät im Baum des System Managers aus und wählen Sie den Dialog *Cyclic Data*.



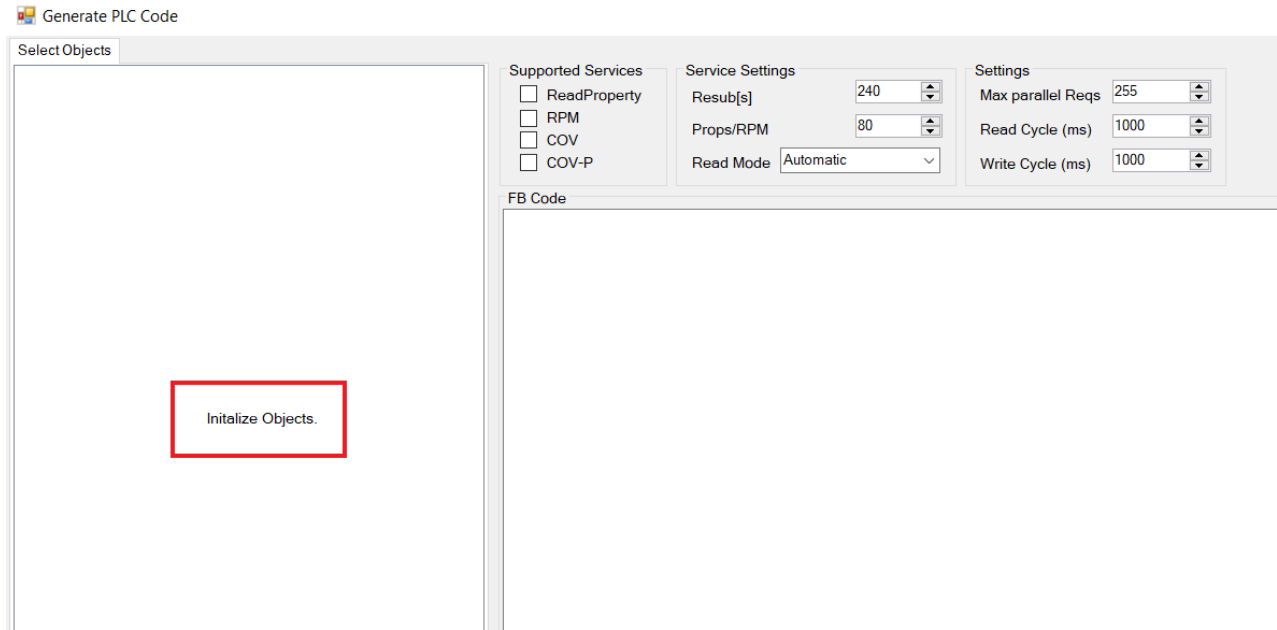
7.6 Der Dialog Client FB Code

Nach dem Aufruf der Funktion FB Code öffnet der unten abgebildete Dialog das Fenster zur Codegenerierung.

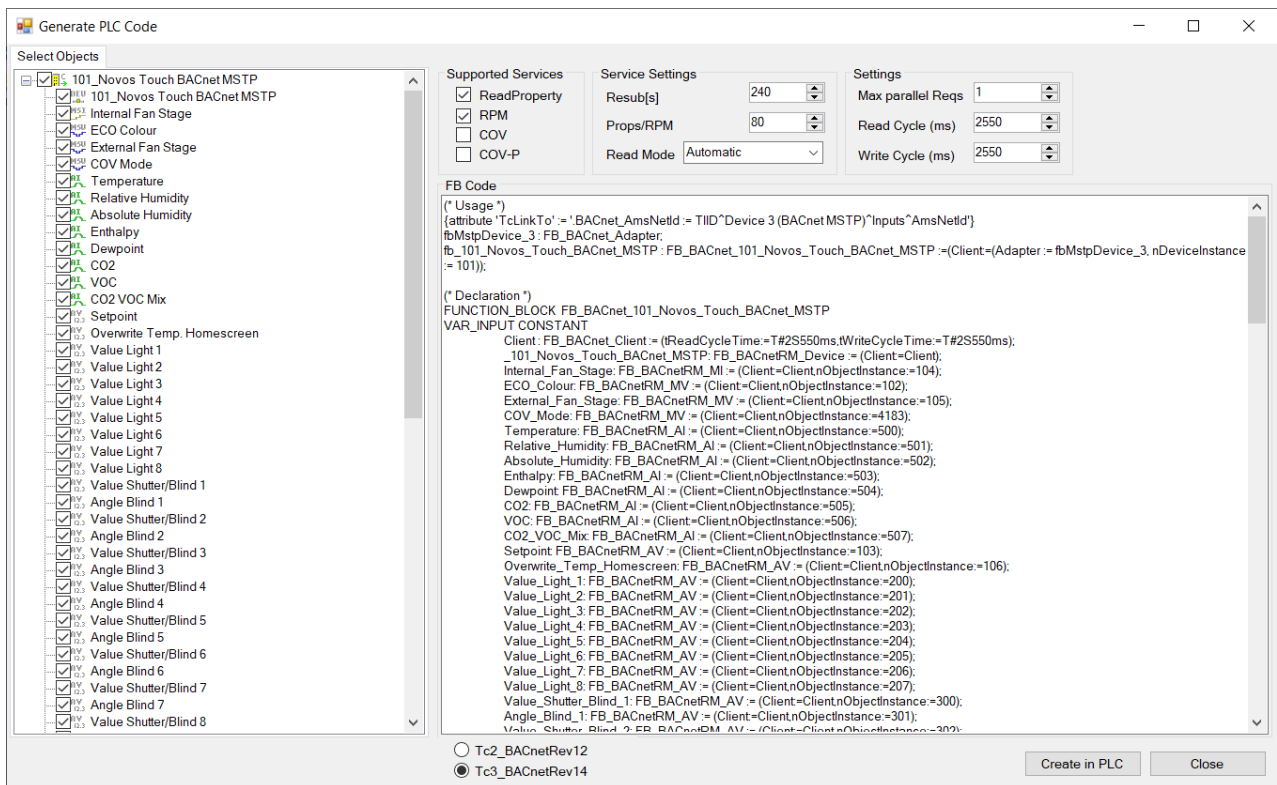
Die im ausgewählten Gerät enthaltenen BACnet-Objekte werden in der Strukturansicht im linken Fenster angezeigt.



Wenn Sie FB Code aus dem Dialog **Scan** verwenden, kann die Suche nach allen Objekten in einem Gerät einige Zeit in Anspruch nehmen. Dies wird durch animierte Punkte angezeigt. Bitte haben Sie etwas Geduld, um das gesamte Gerät zu lesen, besonders, wenn Sie langsamere MS/TP-Geräte scannen!



Nachdem das gesamte Gerät gescannt wurde, zeigt der Dialog alle im Gerät enthaltenen Objekte an.



7.6.1 Das Fenster Select Objects

In diesem Fenster werden die im Gerät enthaltenen Objekte angezeigt. Objekte, die nicht in die Codegenerierung einbezogen werden sollen, können mit Hilfe der Kontrollkästchen deaktiviert werden.

7.6.2 Unterstützte Services

In diesem Fenster werden die unterstützten Services des Peer-Geräts angezeigt.

ReadProperty

Zur Ermittlung der Property-Werte wird eine zyklische Abfrage durchgeführt. Read Property liest nur einzelne Eigenschaften. Read Property Polling ist daher prinzipbedingt sehr langsam.

RPM = Read Property Multiple

RPM ist eine zyklische Abfrage, die mehrere Werte auf einmal abfragt, um die Property-Werte zu ermitteln. Dieser Prozess ist schneller als das Abfragen mit Read Property. In einigen Fällen muss der Wert *PropsRPM* auf einen niedrigeren Wert gesenkt werden, insbesondere bei Geräten mit begrenzten Ressourcen.

COV = Change of Value

Diese Einstellung verwendet COV, um die Property-Werte (Present Value und Status Flags) abzufragen.

Hinweis: Einige Geräte, insbesondere MS/TP-Geräte, unterstützen COV möglicherweise nur für einige, nicht für alle im Gerät enthaltenen Objekte. Heben Sie in diesem Fall die Auswahl von COV aus den unterstützten Services auf und fügen Sie, falls erforderlich, die COV-Unterstützung im SPS-Code nur den Objekten hinzu, die COV unterstützen. Im Datenblatt des BACnet-Device finden Sie die entsprechenden Objekte. Wenn Sie unsicher sind, deaktivieren Sie COV und verwenden Sie stattdessen die RP/RPM-Abfrage.

COV-P = Change of Value Property

Bei diesem Service abonniert TwinCAT als Client einzelne Eigenschaften (Properties). Hinweis: Nicht alle BACnet-Devices unterstützen diesen Service!

7.6.3 Service-Einstellungen

Resub (s)

Mit dieser Einstellung wird das Intervall für das erneute Abonnieren von Change-of-Value-Benachrichtigungen festgelegt. Dieser Wert gibt auch den Heartbeat an (jedes Abonnement führt zu einer Wertaktualisierung).

Props/RPM

Diese Einstellung gibt die Anzahl der Eigenschaften pro ReadPropertyMultiple-Anfrage an. Bitte beachten Sie, dass einige BACnet-Devices möglicherweise nicht in der Lage sind, eine größere Anzahl von Eigenschaften pro Anfrage zu verarbeiten. In diesem Fall kann dieser Wert für eine ordnungsgemäße Kommunikation verringert werden.

Read Mode

Automatisch

Unconfirmed COV

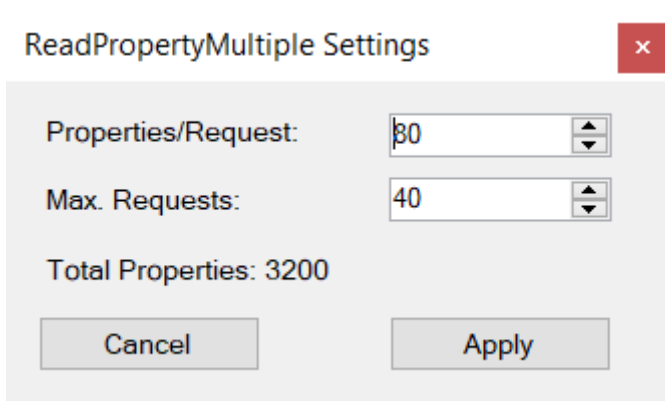
Confirmed COV

Unsubscribed COV

Read Property

RPM=Read Property Multiple

Die Berechnung der Anzahl der ReadPropertyMultiple-Anfragen wird durch die Einstellungen im Dialog ReadProperty Multiple Settings bestimmt.



Der erste Punkt gibt die Höchstzahl der Eigenschaften an, die in einer einzelnen RPM-Anfrage angefordert werden.

Der zweite Punkt gibt die maximale Anzahl von RPM-Anfragen an.

Bitte beachten Sie: Diese Einstellungen sind für Beckhoff-Peer-Geräte optimiert und können reduziert werden, falls andere Geräte keine hohe Anzahl an Anfragen oder Eigenschaften pro Anfrage verarbeiten können.

7.6.4 Das Fenster Settings

Max parallel Requests

Diese Einstellung gibt an, wie viele Anfragen gesendet werden dürfen, bevor die Antwort des Peer-Geräts abgewartet wird. Bitte beachten Sie, dass einige Geräte möglicherweise keine große Anzahl an parallelen Anfragen unterstützen. Wenn Sie unsicher sind, verwenden Sie den Wert 1 und achten Sie auf mögliche Timeout-Meldungen im Fehlerprotokollfenster.

Read Cycle (ms)

Dieser Wert gibt das Leseabfrageintervall in Millisekunden an.

Write Cycle (ms)

Dieser Wert gibt das Schreibintervall in Millisekunden an.

7.6.5 Das Fenster FB Code

Dieses Fenster zeigt den aus den Objektinformationen des ausgewählten Geräts generierten Code in Strukturiertem Text (ST) an.

Dieses Fenster ist in drei Bereiche unterteilt.

Der Abschnitt (** Usage **) zeigt, wie man Instanzen dieses Funktionsbausteins erstellt und fügt im Falle von MS/TP ein Attribut TcLinkTo... ein

Beispiel:

```
{attribute 'TcLinkTo' := '.BACnet_AmsNetId := TIID^Device 3 BACnet MSTP)^Inputs^AmsNetId'}
fbMstpDevice_3 : FB_BACnet_Adapter;
fb_101_Novos_Touch_BACnet_MSTP : FB_BACnet_101_Novos_Touch_BACnet_MSTP :=(Client:=(Adapter :=
fbMstpDevice_3, nDeviceInstance := 101));
```

Der Abschnitt (** Deklaration **) zeigt die Variablen der verwendeten Funktionsbausteine.

Beispiel:

```
FUNCTION_BLOCK FB_BACnet_101_Novos_Touch_BACnet_MSTP
VAR_INPUT CONSTANT
    Client : FB_BACnet_Client:=
        (tReadCycleTime:=T#2S550ms,tWriteCycleTime:=T#2S550ms);
    _101_Novos_Touch_BACnet_MSTP: FB_BACnetRM_Device:=
```



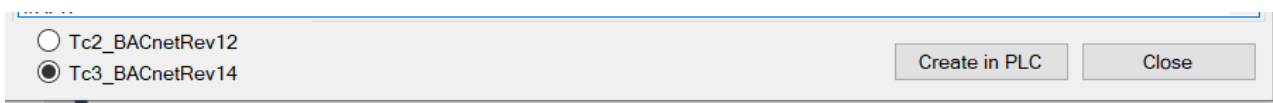
```
(Client:=Client);
Internal_Fan_Stage: FB_BACnetRM_MI :=
  (Client:=Client,nObjectInstance:=104);
ECO_Colour: FB_BACnetRM_MV :=
  (Client:=Client,nObjectInstance:=102);
External_Fan_Stage: FB_BACnetRM_MV :=
  (Client:=Client,nObjectInstance:=105);
END_VAR
```

Der Abschnitt (* Code *) zeigt die Instanzzaufrufe der Funktionsbausteine.

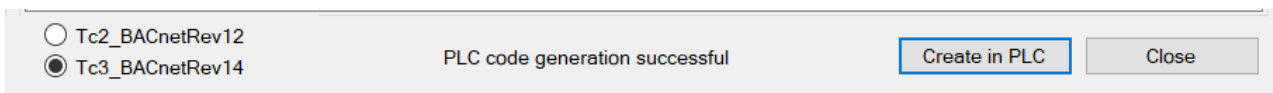
Beispiel:

```
Client();
_101_Novos_Touch_BACnet_MSTP();
Internal_Fan_Stage();
ECO_Colour();
External_Fan_Stage();
```

In der Fußzeile dieses Dialogs befindet sich ein Schalter, der die Codegenerierung beeinflusst. Es ist möglich, den Code für BACnet Revision 12 unter Verwendung der Automapping-Kommentare oder für BACnet Revision 14 unter Verwendung des SPS-Codes zu generieren.



Durch Auswahl von Create in PLC wird die Codegenerierung gestartet. Nach erfolgreicher Erstellung wird in der Fußzeile eine Meldung angezeigt. Wählen Sie Schließen, um den Dialog zu schließen und zum TwinCAT System Manager zurückzukehren.



Nach diesem Schritt ist es nur noch notwendig, den Funktionsbausteinaufruf (z. B. in der POU MAIN) wie oben gezeigt zu platzieren.



Der Inhalt des Fensters FB Code kann zur weiteren Bearbeitung auch in die Zwischenablage kopiert werden.

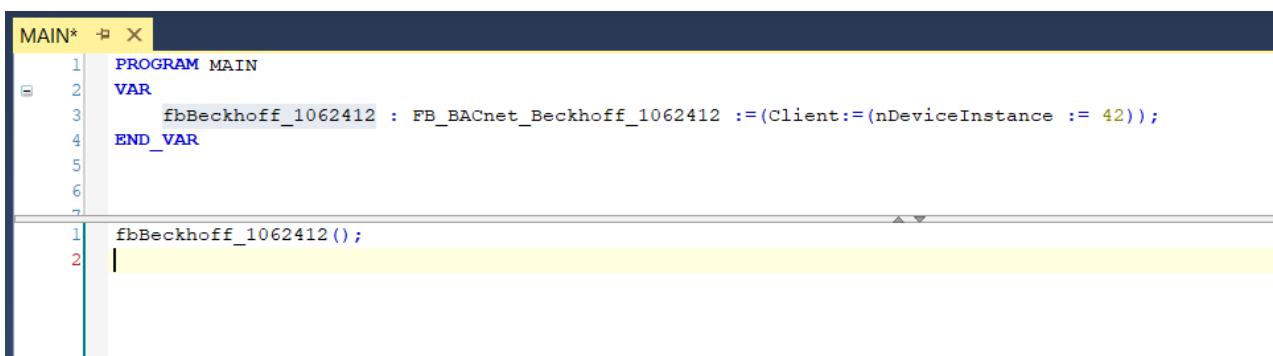
7.6.6 Verwendung des von FB_Code erstellten FB

Der von FB_Code generierte Code wird nicht automatisch aufgerufen. Die ersten Zeilen in der Variablendeklaration des generierten FBs enthalten die Verwendungsinformationen.

Beispiel BACnet IP

```
(* Usage :
fbBeckhoff_1062412 : FB_BACnet_Beckhoff_1062412 :=(Client:=(nDeviceInstance := 42));
*)
```

Zyklischer Aufruf, z. B. in POU Main



Beispiel BACnet MS/TP

```
(* Usage :
{attribute 'ToLinkTo' := '.BACnet_AmsNetId := TIID^Device 3 (BACnet MSTP)^Inputs^AmsNetId'}
fbMstpDevice_3 : FB_BACnet_Adapter;
fb_101_Novos_Touch_BACnet_MSTP : FB_BACnet_101_Novos_Touch_BACnet_MSTP :=(Client:=(Adapter := fbMstpDevice_3, nDevi
fbMstpDevice_3, nDeviceInstance := 101));
*)
```

Zyklischer Aufruf, z. B. in POU Main

```
FB_BACnet_101_No...ouch_BACnet_MSTP* MAIN*
1 PROGRAM MAIN
2 VAR
3   {attribute 'ToLinkTo' := '.BACnet_AmsNetId := TIID^Device 3 (BACnet MSTP)^Inputs^AmsNetId'}
4   fbMstpDevice_3 : FB_BACnet_Adapter;
5   fb_101_Novos_Touch_BACnet_MSTP : FB_BACnet_101_Novos_Touch_BACnet_MSTP :=(Client:=(Adapter := fbMstpDevice_3, nDevi
6 END_VAR
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
259
```

Zyklische Aufrufe in MAIN:

```

VAR
  {attribute 'TcLinkTo' := '.BACnet_AmsNetId := TIID^Device 3 (BACnet MSTP)^Inputs^AmsNetId'}
  fbMstpDevice_3 : FB_BACnet_Adapter;
  fbRoom01 : FB_RoomController :=(Client:=(Adapter := fbMstpDevice_3, nDeviceInstance := 101));
  fbRoom02 : FB_RoomController :=(Client:=(Adapter := fbMstpDevice_3, nDeviceInstance := 102));
  fbRoom03 : FB_RoomController :=(Client:=(Adapter := fbMstpDevice_3, nDeviceInstance := 42));
END_VAR

fbMstpDevice_3 ();
fbRoom01 ();
fbRoom02 ();
fbRoom03 ();
|

```

7.7 Client-Variablen

Die Client-FBs liefern verschiedene Informationen über den Status der Peer-Verbindung.

Adapter:

eDevState: Beschreibt die Zustandsmaschine in der Verbindungsphase. Sollte auf `eComplete` stehen, wenn alle Schritte erfolgreich waren.

bEthLink: TRUE, wenn die Verbindung hergestellt wurde.

bGateway: TRUE, wenn die IP-Adresseinstellung eine Gateway-Information (IT-Router) enthält.

_bHasStarted: TRUE, wenn die Zustandsmaschine gestartet ist.

_nUpdateCount: Dieser Wert wird in jedem Zyklus inkrementiert.

Client:

bAutoResetObjError: Wenn auf TRUE gesetzt, wird die Client-Zustandsmaschine im Falle einer Kommunikationsunterbrechung automatisch zurückgesetzt.

bSuppRpm: TRUE, wenn das Peer-Gerät `ReadPropertyMultiple` unterstützt.

bSuppCov: TRUE bedeutet, dass das Peer-Gerät `ChangeOfValue` unterstützt.

bSuppCovP: TRUE bedeutet, dass das Peer-Gerät `ChangeOfValueProperty` unterstützt.

bReady: Der FB ist initialisiert und bereit, Daten vom Peer-Gerät anzufordern.

bConnected: TRUE bedeutet, dass eine Verbindung zum Peer-Gerät erfolgreich hergestellt wurde.

Falls der Client nicht den automatischen Modus verwendet, bestimmen diese Variablen, welche Services zum Abrufen der Werte verwendet werden: `bSuppRpm`, `bSuppCov`, `bSuppCovP`

7.8 Entfernte Zeitplanobjekte

BACnet-Zeitplanobjekte haben keinen spezifischen Datentyp. Der Datentyp, der für den -Zeitschaltplan verwendet wird, wird durch die Eigenschaften `Weekly-Schedule`, `Exception-Schedule`, `Schedule-Default` und `ListofObjectPropertyReferences` bestimmt. Um externe Zeitplanobjekte zu referenzieren, muss der entsprechende Funktionsbaustein manuell ausgewählt werden (`FB_BACnetRM_SchedA` für analogen Zeitschaltplan, `FB_BACnetRM_SchedB` für binären Zeitschaltplan, `FB_BACnetRM_SchedM` für Multistate-Zeitschaltplan).

7.9 Azyklisches Lesen

Um Eigenschaften aus dem Objekt eines Peer-Geräts zu lesen, bietet die Bibliothek zwei Funktionen für azyklisches Lesen:

FB_BACnetRM_ReadProperty

FB_BACnetRM_ReadPropertyEx

Das BACnet-Objekt des Peer-Geräts wird durch einen Zeiger auf die FB-Remote-Instanz (Variable iObject) referenziert. Im Vergleich zum Basisfunktionsbaustein FB_BACnetRM_ReadProperty bietet der Funktionsbaustein FB_BACnetRM_ReadPropertyEx zwei zusätzliche Variablen zur Angabe des Objekttyps und der Instanznummer (Variablen eObjType und nObjInst).

7.9.1 Beispiel FB_BACnetRM_ReadProperty

Dieses Beispiel zeigt, wie der Funktionsbaustein FB_BACnetRM_ReadProperty verwendet wird, um den Wert der Eigenschaft high_limit vom Analog-Input-Objekt, Instanznummer 1 in der Peer-Geräteinstanznummer 42 zu lesen. Bitte beachten Sie, dass das von iObject referenzierte Objekt zyklisch aufgerufen werden muss.

```
VAR
  fbClient : FB_BACnet_Client := (nDeviceInstance :=
2,tReadCycleTime:=T#10S,nMaxParallelRequests:=255);
  fbDevice : FB_BACnetRM_Device := (Client:=fbClient);
  fbAI : FB_BACnetRM_AI := (Client:=fbClient,nObjectInstance:=1);

  fbRead : FB_BACnetRM_ReadProperty := (Client := fbClient);
  bReadHighLimit : BOOL;
  fHighLimit : REAL;
END_VAR
-----
fbClient();
fbDevice();
fbAI();

// Read HighLimit using FB_BACnetRM_ReadProperty
fbRead.bExecute := bReadHighLimit;
IF fbRead.bExecute THEN
  bReadHighLimit := FALSE;
  fbRead.pData:= ADR( fHighLimit );
  fbRead.nData:= SIZEOF( fHighLimit );
  fbRead.ePropID:= E_BACnet_PropertyIdentifier.PropHighLimit;
  fbRead.iObject := fbAI;
END_IF
fbRead();
```

7.9.2 Beispiel FB_BACnetRM_ReadPropertyEx

Das folgende Beispiel zeigt die Verwendung des Funktionsbausteins FB_BACnetRM_ReadPropertyEx.

```
VAR
  fbClient : FB_BACnet_Client := (nDeviceInstance :=
42,tReadCycleTime:=T#10S,nMaxParallelRequests:=255);
  fbDevice : FB_BACnetRM_Device := (Client:=fbClient);

  fbReadEx : FB_BACnetRM_ReadPropertyEx := (Client := fbClient);
  bReadLowLimitEx : BOOL;
  fLowLimitEx : REAL;
END_VAR
-----
fbClient();
fbDevice();

// Read LowLimit using FB_BACnetRM_ReadPropertyEx
fbReadEx.bExecute := bReadLowLimitEx;
IF fbReadEx.bExecute THEN
  bReadLowLimitEx := FALSE;
  fbReadEx.pData:= ADR( fLowLimitEx );
  fbReadEx.nData:= SIZEOF( fLowLimitEx );
  fbReadEx.ePropID:= E_BACnet_PropertyIdentifier.PropLowLimit;
  fbReadEx.nObjInst:= 1;
END_IF
```

```

    fbReadEx.eObjType:= E_BACnet_ObjectType.ObjAnalogInput;
END_IF
fbReadEx();

```

7.10 Azyklisches Schreiben

Um Eigenschaften in das Objekt eines Peer-Geräts zu schreiben, bietet die Bibliothek zwei Funktionen für azyklisches Schreiben:

FB_BACnetRM_WriteProperty

FB_BACnetRM_WritePropertyEx

Das BACnet-Objekt des Peer-Geräts wird durch einen Zeiger auf die FB-Remote-Instanz (Variable iObject) referenziert. Im Vergleich zum Basisfunktionsbaustein FB_BACnetRM_WriteProperty bietet der Funktionsbaustein FB_BACnetRM_WritePropertyEx zwei zusätzliche Variablen zur Angabe des Objekttyps und der Instanznummer (Variablen eObjType und nObjInst).

7.10.1 Beispiel FB_BACnetRM_WriteProperty

Dieses Beispiel zeigt, wie der Funktionsbaustein FB_BACnetRM_WriteProperty verwendet wird, um den Wert der out_of_service-Eigenschaft des Binary-Output-Objekts, Instanznummer 0 in die Peer-Geräteinstanznummer 42 zu schreiben. Bitte beachten Sie, dass das von iObject referenzierte Objekt zyklisch aufgerufen werden muss.

```

VAR
    fbClient : FB_BACnet_Client := (nDeviceInstance :=
42,tReadCycleTime:=T#10S,nMaxParallelRequests:=255);
    fbDevice : FB_BACnetRM_Device := (Client:=fbClient);
    fbBO : FB_BACnetRM_BO := (Client:=fbClient,nObjectInstance:=0);

    fbWrite : FB_BACnetRM_WriteProperty := (Client := fbClient);
    bWriteOoS : BOOL;
    bOutOfService : BOOL;
END_VAR
-----
fbClient();
fbDevice();
fbBO();

// Write OutOfService using FB_BACnetRM_WriteProperty
fbWrite.bExecute := bWriteOoS;
IF fbWrite.bExecute THEN
    bWriteOoS := FALSE;
    fbWrite.pData:= ADR( bOutOfService );
    fbWrite.nData:= SIZEOF( bOutOfService );
    fbWrite.ePropID:= E_BACnet_PropertyIdentifier.PropOutOfService;
    fbWrite.iObject := fbBO;
END_IF
fbWrite();

```

7.10.2 Beispiel FB_BACnetRM_WritePropertyEx

Das folgende Beispiel zeigt die Verwendung des Funktionsbausteins FB_BACnetRM_WritePropertyEx.

```

VAR
    fbClient : FB_BACnet_Client := (nDeviceInstance :=
42,tReadCycleTime:=T#10S,nMaxParallelRequests:=255);
    fbDevice : FB_BACnetRM_Device := (Client:=fbClient);

    fbWriteEx : FB_BACnetRM_WritePropertyEx := (Client := fbClient);
    bWriteOoSEx : BOOL;
    bOutOfServiceEx : BOOL;
END_VAR
-----
fbClient();
fbDevice();

// Write OutOfService using FB_BACnetRM_WritePropertyEx
fbWriteEx.bExecute := bWriteOoSEx;
IF fbWriteEx.bExecute THEN

```

```
bWriteOoSEx := FALSE;
fbWriteEx.pData:= ADR( bOutOfServiceEx );
fbWriteEx.nData:= SIZEOF( bOutOfServiceEx );
fbWriteEx.ePropID:= E_BACnet_PropertyIdentifier.PropOutOfService;
fbWriteEx.nObjInst:= 1;
fbWriteEx.eObjType:= E_BACnet_ObjectType.ObjBinaryOutput;
END_IF
fbWriteEx();
```

7.11 Überwachung einer Client-Verbindung

Um eine Client-Verbindung zu überwachen, ist es notwendig, den Funktionsbaustein `FB_BACnetRM_Device` zyklisch aufzurufen.

Der aktuelle Status der Verbindung kann dem Funktionsbaustein `FB_BACnet_Client` entnommen werden.

Wird die Verbindung unterbrochen, z. B. durch Trennen des Peer-Gerätes oder durch ein defektes Netzkabel, versucht das BACnet-Supplement, die Verbindung wiederherzustellen. Wenn nach diesen Versuchen (Variable `nErrorCnt` im Funktionsbaustein `FB_BACnetRM_Device`) die Verbindung immer noch unterbrochen ist, wird angenommen, dass das Gerät oder die Netzwerkverbindung nicht mehr vorhanden ist. Dieser Vorgang kann 30 Sekunden oder länger dauern, um den Verbindungsunterbrechung sicher zu erkennen.

In diesem Fall nimmt die Variable `eSysState` des `FB_BACnetRM_Device` den Wert `eNoCommunication` an und `bOperational` wird auf `FALSE` gesetzt. Die Variablen `bReady` und `bConnected` des Funktionsbausteins `FB_BACnet_Client` werden auf `FALSE` gesetzt, die Variable `eState` nimmt den Wert `eInit` an.

Das BACnet-Supplement versucht weiterhin, eine Verbindung zum Peer-Gerät herzustellen. Sobald die Verbindung wieder hergestellt ist, verhält sich die Zustandsmaschine wie beim ersten Start.

7.11.1 Beispiel: Die Verbindung wurde erfolgreich hergestellt

`FB_BACnet_Client`

Expression	Type	Value
Client	FB_BACnet_Client	
Adapter	REFERENCE TO FB_BACnet_Adapter	
nDeviceInstance	UDINT	42
nMaxParallelRequests	UDINT	50
nRPMPropertiesPerRequest	UDINT	80
tCOVLifeTime	TIME	T#4m
eReadMode	E_BACNET_COMMMODE	eCov
eWriteMode	E_BACNET_WRITEMODE	eOnChange
tReadCycleTime	TIME	T#2s550ms
tWriteCycleTime	TIME	T#2s550ms
bAutoResetObjError	BOOL	TRUE
bSuppRpm	BOOL	TRUE
bSuppCov	BOOL	TRUE
bSuppCovP	BOOL	TRUE
bReady	BOOL	TRUE
eErrorID	E_BACNET_ERROR	eNoError
nAdsPort	UINT	1001
bConnected	BOOL	TRUE
tObjectID	T_BACnet_ObjectIdentifier	33554474
eState	E_BACNETRM_CLIENTSTATE	eCycle
stSupportedServices	ST_BACnet_ServicesSupportedBits	
nMaxAdduSize	UDINT	1476

FB_BACnetRM_Device

Expression	Type	Value
Beckhoff_1062412	FB_BACnetRM_Device	
Client	REFERENCE TO FB_BACnet_Client	
nObjectInstance	UDINT	42
eReadMode	E_BACNET_COMMMODE	eRp
bReady	BOOL	TRUE
eObjectType	E_BACNETOBJTYPE	ObjDevice
eErrorID	E_BACNET_ERROR	eNoError
tObjectID	T_BACnet_ObjectIdentifier	33554474
eAutoResetClient	BOOL	TRUE
nAutoResetAfterErrors	UDINT	3
eSysState	E_BACNET_DEVICESTATUS	eOperational
bOperational	BOOL	TRUE
nErrorCnt	UDINT	0
CurrentConfig_xml	FB_BACnetRM_FILE	
BACnetOnline_1010010_bootdata	FB_BACnetRM_FILE	
Term_2_EL2809_Chn1	FB_BACnetRM_BO	
Term_2_EL2809_Chn2	FB_BACnetRM_BO	
Term_2_EL2809_Chn3	FB_BACnetRM_BO	

7.11.2 Beispiel: Die Verbindung ist unterbrochen

FB_BACnet_Client

Expression	Type	Value
Client	FB_BACnet_Client	
Adapter	REFERENCE TO FB_BACnet_Adapter	
nDeviceInstance	UDINT	42
nMaxParallelRequests	UDINT	50
nRPMPPropertiesPerRequest	UDINT	80
tCOVLifeTime	TIME	T#4m
eReadMode	E_BACNET_COMMMODE	eCov
eWriteMode	E_BACNET_WRITEMODE	eOnChange
tReadCycleTime	TIME	T#2s550ms
tWriteCycleTime	TIME	T#2s550ms
bAutoResetObjError	BOOL	TRUE
bSuppRpm	BOOL	TRUE
bSuppCov	BOOL	TRUE
bSuppCovP	BOOL	TRUE
bReady	BOOL	FALSE
eErrorID	E_BACNET_ERROR	eNoError
nAdsPort	UINT	1001
bConnected	BOOL	FALSE
tObjectID	T_BACnet_ObjectIdentifier	33554474
eState	E_BACNETRM_CLIENTSTATE	eInit
stSupportedServices	ST_BACnet_ServicesSupportedBits	
nMaxAdduSize	UDINT	1476

FB_BACnetRM_Device

Expression	Type	Value
Beckhoff_1062412	FB_BACnetRM_Device	
Client	REFERENCE TO FB_BACnet_Client	
nObjectInstance	UDINT	42
eReadMode	E_BACNET_COMMMODE	eRp
bReady	BOOL	FALSE
eObjectType	E_BACNETOBJTYPE	ObjDevice
eErrorID	E_BACNET_ERROR	eNoError
tObjectID	T_BACnet_ObjectIdentifier	33554474
eAutoResetClient	BOOL	TRUE
nAutoResetAfterErrors	UDINT	3
eSysState	E_BACNET_DEVICESTATUS	eNoCommunication
bOperational	BOOL	FALSE
nErrorCnt	UDINT	0
CurrentConfig_xml	FB_BACnetRM_FILE	
BACnetOnline_1010010_bootdata	FB_BACnetRM_FILE	
Term_2_EL2809_Chn1	FB_BACnetRM_BO	
Term_2_EL2809_Chn2	FB_BACnetRM_BO	

7.12 Verwendung von ReadPropertyMultiple

Wenn aus einem angeschlossenen BACnet-Gerät mehrere Properties ausgelesen werden sollen, empfiehlt sich die Verwendung des Dienstes **ReadPropertyMultiple** (RPM). Im Gegensatz zur Verwendung des Dienstes **ReadProperty**, welcher nur einzelne Properties lesen kann, fasst **ReadPropertyMultiple** mehrere Abfragen zu einem Telegramm zusammen und arbeitet daher erheblich effizienter.

Das nachfolgende Beispiel zeigt die Verwendung von RPM. Es setzt allerdings voraus, dass angeschlossene Geräte diesen Dienst unterstützen. Die drei in diesem Beispiel gezeigten Lesebefehle werden zu einem einzelnen RPM zusammengesetzt.

Variablen

```

(attribute 'TcLinkTo' := '.BACnet_AmsNetId := TIID^Device 1 (BACnet MSTP)^Inputs^AmsNetId')
fbMstpDevice_1 : FB_BACnet_Adapter;

fbClient : FB_BACnet_Client := (Adapter := fbMstpDevice_1, nDeviceInstance := 116005,
tReadCycleTime:=T#10S, nMaxParallelRequests:=20);
fbDevice : FB_BACnetRM_Device := (Client:=fbClient);

fbTon : TON;

fbReadEx1 : FB_BACnetRM_ReadPropertyEx := (Client := fbClient);
fbReadEx2 : FB_BACnetRM_ReadPropertyEx := (Client := fbClient);
fbReadEx3 : FB_BACnetRM_ReadPropertyEx := (Client := fbClient);

bRead : BOOL;

fPVSupplyTemp : REAL;
fPVReturnTemp : REAL;
fDesignFlow : REAL;

```

Code

```

fbMstpDevice_1();
fbClient();
fbDevice();

fbTon( IN:= NOT fbTon.Q, PT:= T#5S );
IF fbTon.Q THEN
  bRead := TRUE;
END_IF

fbReadEx1.bExecute := bRead;
fbReadEx2.bExecute := bRead;
fbReadEx3.bExecute := bRead;

```

```
IF bRead THEN
  bRead := FALSE;

  fbReadEx1.pData:= ADR( fPVSupplyTemp );
  fbReadEx1.nData:= SIZEOF( fPVSupplyTemp );
  fbReadEx1.ePropID:= E_BACnet_PropertyIdentifier.PropPresentValue;
  fbReadEx1.nObjInst:= 1;
  fbReadEx1.eObjType:= E_BACnet_ObjectType.ObjAnalogInput;

  fbReadEx2.pData:= ADR( fPVReturnTemp );
  fbReadEx2.nData:= SIZEOF( fPVReturnTemp );
  fbReadEx2.ePropID:= E_BACnet_PropertyIdentifier.PropPresentValue;
  fbReadEx2.nObjInst:= 2;
  fbReadEx2.eObjType:= E_BACnet_ObjectType.ObjAnalogInput;

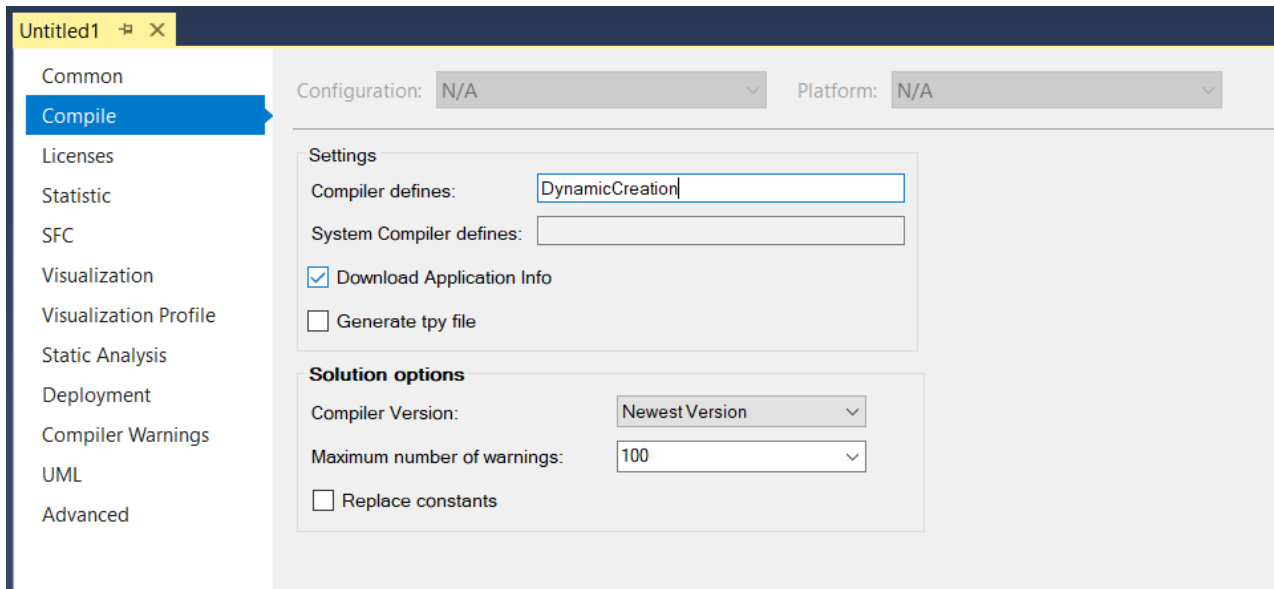
  fbReadEx3.pData:= ADR( fDesignFlow );
  fbReadEx3.nData:= SIZEOF( fDesignFlow );
  fbReadEx3.ePropID:= E_BACnet_PropertyIdentifier.PropPresentValue;
  fbReadEx3.nObjInst:= 0;
  fbReadEx3.eObjType:= E_BACnet_ObjectType.ObjAnalogValue;
END_IF

fbReadEx1();
fbReadEx2();
fbReadEx3();
```


8 Dynamic Object Manager

Mit dem Dynamic Object Manager können BACnet-Objekte zur Laufzeit erstellt oder gelöscht werden.

Um diese Funktion zu nutzen, muss das Pragma "*DynamicCreation*" in den Compiler-Einstellungen wie unten gezeigt gesetzt werden:



8.1 FB_BACnet_DynObjectManager

Der `FB_BACnet_DynObjectManager` bietet die Möglichkeit, Objekte zur Laufzeit dynamisch zu erstellen oder zu löschen. Typische Anwendungsfälle sind lokale Visualisierungen oder die Erstellung von BACnet-Objekten auf Basis von Konfigurationsdateien.

Es ist möglich, den Dynamic Object Manager zusammen mit statisch erzeugten Objekten zu verwenden. Bitte beachten Sie jedoch, dass die Objektinstanznummern eindeutig sein müssen!

8.2 Zyklische Aufrufe

Der Dynamic Object Manager und alle dynamisch erstellten Objekte müssen zyklisch aufgerufen werden (wie alle anderen Objekte einmal und nur einmal pro Zyklus). Das Setzen des Flags `bCycleObjects := TRUE` ermöglicht zyklische Aufrufe der erzeugten Objekte. In diesem Fall dürfen keine weiteren zyklischen Aufrufe erfolgen (außer dem Aufruf der Instanz von `FB_BACnet_DynObjectManager`).

8.3 Vordefinierter Objektpool

Dynamische Objekte werden mit dem Operator `__NEW` erstellt. Ist die Anzahl der zu erstellenden Objekte zum Zeitpunkt des ersten SPS-Starts bekannt oder abschätzbar, bietet die Bibliothek einen vordefinierten Objektpool. Dies ist speichereffizienter als die Verwendung des `__NEW`-Operators. Die Elemente mit dem Namen `nPool_XX` im Abschnitt `BACnet_Param` der Bibliotheks-GVLs können eingerichtet werden, um Ressourcen zu sparen.

Name	Type	Value (editable)
nPool_AI	UINT	0
nPool_AIRaw	UINT	0
nPool_AO	UINT	0
nPool_AO5P	UINT	0
nPool_AORaw	UINT	0
nPool_AORawSP	UINT	0
nPool_AV	UINT	0
nPool_AVRawSP	UINT	0
nPool_AVEvent	UINT	0
nPool_AVDisp	UINT	0
nPool_AVSetp	UINT	0
nPool_BI	UINT	0
nPool_BIRaw	UINT	0

Wenn z. B. 42 Instanzen eines FB_BACnet_AV verwendet werden, kann die Variable nPoolAV auf 42 als Standardwert gesetzt werden. Wenn mehr Objekte erstellt werden, als im Pool verfügbar sind, wird der Erstellungsprozess weiterhin den __NEW-Operator verwenden.

8.4 Beispiel

Der folgende Code zeigt, wie die Funktion der dynamischen Erzeugung/Löschung genutzt werden kann.

Variablen:

```
PROGRAM DYN_OBJECTS
VAR
  fbDynObject : FB_BACnet_DynObjectManager := (bCycleObjects := TRUE);

  bCreate : BOOL;
  bDelete : BOOL;

  DynView : POINTER TO FB_BACnet_View;
  DynAV01 : POINTER TO FB_BACnet_AV;
  DynBV01 : POINTER TO FB_BACnet_BV;
  nCounter : UDINT;
END_VAR
```

Code:

```
// Management FB has to be called in every cycle
fbDynObject();

IF (fbDynObject.Ready) THEN
  IF (bCreate) THEN
    bCreate := FALSE;
    // [Variant 1] Create standard object types:
    IF (fbDynObject.CreateObject(
      DynView,
      E_BACnet_CreateObjType.eStructuredView,
      BACnet_Globals.nBACnetInstId_Auto,
      'DynView',
      'Dynamic View',
      0))
    THEN
      // [Optional] set object properties...
    END_IF
    IF (fbDynObject.CreateObject(
      DynAV01,
      E_BACnet_CreateObjType.eAnalogValue,
      BACnet_Globals.nBACnetInstId_Auto,
      '\DynAV01',
      '\Dynamic AV 1',
      DynView))
    THEN
      // [Optional] set object properties...
      DynAV01^.eUnit := E_BA_Unit.eTemperature_DegreesCelsius;
    END_IF
    IF (fbDynObject.CreateObject(
      DynBV01,
```

```

        E_BACnet_CreateObjType.eBinaryValue,
        BACnet_Globals.nBACnetInstId_Auto,
        '\DynBV01',
        '\Dynamic BV 1',
        DynView))
    THEN
        // [Optional] set object properties...
        DynBV01^.sInactiveText := 'Off';
        DynBV01^.sActiveText := 'On';
    END_IF
END_IF

IF (bDelete) THEN
    bDelete := FALSE;
    // [Variant 1] Delete all objects conveniently via object manager:
    fbDynObject.Reset();

    // [Variant 2] Delete all objects manually:
    (*
    fbDynObject.DeleteObject(DynAV01);
    fbDynObject.DeleteObject(DynBV01);
    fbDynObject.DeleteObject(DynView);
    *)
END_IF

// Sample PLC code:
// > Take care of valid object pointers!
IF (fbDynObject.CreatedObjects > 0) THEN
    nCounter := (nCounter+1);
    // Simulate changing value:
    DynAV01^.bEnPgm := TRUE;
    DynAV01^.fValPgm := (TO_REAL(nCounter MOD 1000)/100);

    // [Variant 2] Call created objects manually
    IF (NOT fbDynObject.bCycleObjects) THEN
        DynView^();
        DynAV01^();
        DynBV01^();
    END_IF
END_IF
END_IF

```

8.5 Initialisierung der dynamischen Objekte abschließen

Die Initialisierung der dynamisch erzeugten Objekte erfolgt im Regelfall automatisch einige Zyklen nachdem das letzte Objekt angelegt wurde.

Wenn jedoch die Initialisierung per SPS -Programm abgeschlossen werden soll (weil z. B. das Auslesen einer Konfigurationsdatei länger dauern könnte), so kann der Dynamic Object Manager so aufgerufen werden, dass das Ende der Initialisierung aus dem SPS-Programm heraus erfolgt.

Dazu ist der zweite Parameter `bAutoFinishInit` auf den Wert `FALSE` zu setzen, z. B. `fbDynMngr : FB_BACnet_DynObjectManager := (bCycleObjects := TRUE, bAutoFinishInit := FALSE);`

Zum Abschluss der Initialisierung ist dann die Methode `FinishInit` aufzurufen, z. B.

```
fbDynMngr.FinishInit();
```

8.6 Anlegen und Löschen eigener BACnet Funktionsbausteine (FB)

Wenn neben den in der Bibliothek `Tc3_BACnetRev14` vorhandenen FB eigene BACnet FB instanziiert werden sollen, müssen diese bei Beenden des SPS-Programms in einer Methode `FB_exit` gelöscht werden, damit der dynamisch allozierte Speicher wieder freigegeben wird.

Dynamisch angelegte FB-Instanzen von Funktionsbausteinen aus der Bibliothek `Tc3_BACnetRev14` werden über die Methode `FB_exit` des Dynamic Object Manager automatisch gelöscht und aus dem Speicher entfernt.

Das nachfolgende Beispiel zeigt das dynamische Anlegen von FB-Instanzen eigener FB (die Implementierungen selbst werden in diesem Beispiel nicht gezeigt) sowie das Löschen dieser Instanzen in der Methode `FB_exit`.



Es ist darauf zu achten, dass der richtige Typ des FB bei der Freigabe verwendet wird.

Variable MAIN

```
PROGRAM MAIN
VAR
    fbDynObj : FB_DYN_OBJECTS;
END_VAR
```

Code MAIN

```
fbDynObj();
```

Variablen FB_DynObj

```
FUNCTION_BLOCK FB_DYN_OBJECTS
VAR
    DynMgmt : FB_BACnet_DynObjectManager := (bCycleObjects := TRUE, bAutoFinishInit := FALSE);

    bCreate : BOOL := TRUE;
    bDelete : BOOL;

    TestFbBVOwn : POINTER TO FB_BACnet_BV_Event;
    TestFbAVOwn : POINTER TO FB_BACnet_AV_EventSetp;
END_VAR
```

Code FB_DynObj

```
DynMgmt();
IF (DynMgmt.Ready) THEN
    IF (bCreate) THEN
        bCreate := FALSE;

        TestFbBVOwn := __NEW( FB_BACnet_BV_Event );
        IF (DynMgmt.CreateObjectEx(TestFbBVOwn, BACnet_Globals.nBACnetInstId_Auto, '\\TestBV own', '\\TestBV own', 0)) THEN
            // Initialize properties:
            TestFbBVOwn^.sInactiveText := 'AUS';
            TestFbBVOwn^.sActiveText := 'EIN';
        END_IF

        TestFbAVOwn := __NEW( FB_BACnet_AV_EventSetp );
        IF (DynMgmt.CreateObjectEx(TestFbAVOwn, BACnet_Globals.nBACnetInstId_Auto, '\\TestEvent own', '\\TestEvent own', 0)) THEN
            // Initialize properties:
            TestFbAVOwn^.fHighLimit := 470;
            TestFbAVOwn^.fLowLimit := -100;
        END_IF
        DynMgmt.FinishInit();
    END_IF

    IF (bDelete) THEN
        bDelete := FALSE;
        FB_exit(FALSE);
    END_IF
END_IF
```

Variablen der Methode FB_DynObj.FB_exit

```
METHOD FB_exit : BOOL
VAR_INPUT
    bInCopyCode : BOOL; // if TRUE, the exit method is called for exiting an instance that is copied afterwards (online change).
END_VAR
```

Code der Methode FB_DynObj.FB_exit

```
DynMgmt.RemoveObjectEx(TestFbAVOwn);
__DELETE(TestFbAVOwn);
DynMgmt.RemoveObjectEx(TestFbBVOwn);
__DELETE(TestFbBVOwn);
```

9 Beispiele

9.1 Variablendeklaration

Die Vorgabewerte der BACnet Properties sollten bei der Variablendeklaration vorgegeben werden und zur Laufzeit ausschließlich bedingt beschrieben werden (damit weiterhin ein Schreibzugriff seitens BACnet möglich ist). Im nachfolgenden Beispiel wird das Property *Description* zur Laufzeit geändert.

Zur Verwendung der BACnet EngineeringUnit (Maßeinheit) muss die Bibliothek Tc3_BA2_Common eingebunden werden.

Variablen

```
fbAv : FB_BACnet_AV := (
    sObjectName := 'X51LU01xAMH_-SW31',
    sDescription := 'SetpointVariable Ventilator',
    eUnit := E_BA_Unit.ePressure_Pascals,
    fRelinquishDefault := 250
);
bDescriptionChanged : BOOL;
```

Code

```
if bDescriptionChanged then
    fbAV.sDescription := 'TEST';
    bDescriptionChanged := FALSE;
END_IF
fbAv();
```

9.2 BACnet Properties

Dieses Beispiel zeigt die Vorgabe weiterer BACnet Eigenschaften (Properties), wie z. B. die Zustandstexte oder **Change-of-Value-Einstellungen (COV)**. Ebenfalls wird gezeigt, wie Funktionsbausteine mit Hilfe einer for-Schleife aufgerufen werden können.

Bei Multistateobjekten wird die Anzahl der Stufen über die Stufentexte (Property *State_Text*) festgelegt. Als Vorgabewert gilt eine Begrenzung von 12 Stufen. Diese Anzahl kann in den Globalen Variablen in der Parameterliste *BACnet_Param* geändert werden.

Variablen

```
// optional unit, range and COV properties
fbAi : FB_BACnet_AI := (
    eUnit := E_BA_Unit.eTemperature_DegreesCelsius,
    fCovIncrement := 2.0,
    fMinPresValue := 0.0,
    fMaxPresValue := 100.0
);

// optional state text information
fbBi : FB_BACnet_BI := (
    sInactiveText := 'DOWN',
    sActiveText := 'UP'
);

// number of states determined by aStateText
fbMi : FB_BACnet_MI := (
    aStateText := ['AUTO', 'Low', 'Medium', 'High', 'Turbo']
);

// array of BACnet FBs
afbAV : ARRAY[0..499] of FB_BACnet_AV;
nCount : INT;
```

Code

```
fbAi();
fbBi();
fbMi();
```

```
FOR nCount := 0 to 499 do
  afbAV[nCount]();
END_FOR
```

9.3 Verknüpfung mit dem Attribut 'TcLinkTo'

Dieses Beispiel zeigt, wie das Attribut 'TcLinkTo' zur Verknüpfung mit EtherCAT- oder K-Bus-Klemmen verwendet werden kann. Es stehen jeweils die FB-Varianten mit der Endung IO (K-Bus) sowie ECAT (EtherCAT-Klemmen) zur Verfügung.

Das Attribut 'TcLinkToOSO' im Beispiel der Klemme EL3068 zeigt darüber hinaus, wie die Zustände Underrange und Ovrerrange auf den Zustand des Klemmenkanals (nRawState) abgebildet werden können.

Allgemein gilt, dass der Zustand des Klemmenkanals über die Variable nRawState abgebildet wird. Die Variable nRawECatState wird verwendet, um zusätzlich den EtherCAT-Status der Kommunikation mit der Klemme abzubilden.

Der TIID-Pfad kann in der Entwicklungsumgebung in den Eigenschaften des jeweiligen Klemmenkanals entnommen und von dort kopiert werden.

Variablen

```
// general structured view object
View : FB_BACnet_View := (sObjectName:='IoBus_0_Device_3_EtherCAT');

// EtherCAT terminals

(* Term 2 (EL1809) *)
{attribute 'TcLinkTo' :=
  '.bRawVal :=TIID^Device 3 (EtherCAT)^Term 1 (EK1100)^Term 2 (EL1809)^Channel 1^Input;
  .nRawECatState:=TIID^Device 3 (EtherCAT)^Term 1 (EK1100)^Term 2 (EL1809)^InfoData^State'}
Term_2_EL1809_Chnl : FB_BACnet_BI_ECAT := (iParent:=View, sObjectName:='Term_2_EL1809_Chnl',
nIoBusNr:=0, sDeviceType:='EL1809 16Ch. Dig. Input 24V, 3ms');

(* Term 3 (EL2809) *)
{attribute 'TcLinkTo' :=
  '.bRawVal :=TIID^Device 3 (EtherCAT)^Term 1 (EK1100)^Term 3 (EL2809)^Channel 1^Output;
  .nRawECatState:=TIID^Device 3 (EtherCAT)^Term 1 (EK1100)^Term 3 (EL2809)^InfoData^State'}
Term_3_EL2809_Chnl : FB_BACnet_BO_ECAT := (iParent:=View, sObjectName:='Term_3_EL2809_Chnl',
nIoBusNr:=0, sDeviceType:='EL2809 16Ch. Dig. Output 24V, 0.5A');

(* Term 4 (EL3068) *)
{attribute 'TcLinkTo' :=
  '.nRawVal :=TIID^Device 3 (EtherCAT)^Term 1 (EK1100)^Term 4 (EL3068)^AI Standard Channel
1^Value;
  .nRawECatState:=TIID^Device 3 (EtherCAT)^Term 1 (EK1100)^Term 4 (EL3068)^InfoData^State'}
{attribute 'TcLinkToOSO' :=
  '.nRawState:=<0,1,0>TIID^Device 3 (EtherCAT)^Term 1 (EK1100)^Term 4 (EL3068)^AI Standard Channel
1^Status^Underrange;
  .nRawState:=<1,1,0>TIID^Device 3 (EtherCAT)^Term 1 (EK1100)^Term 4 (EL3068)^AI Standard Channel
1^Status^Ovrerrange'}
Term_4_EL3068_Chnl : FB_BACnet_AI_ECAT := (iParent:=View, sObjectName:='Term_4_EL3068_Chnl',
nIoBusNr:=0, sDeviceType:='EL3068 8Ch. Ana. Input 0-10V');

(* Term 5 (EL4132) *)
{attribute 'TcLinkTo' :=
  '.nRawVal :=TIID^Device 3 (EtherCAT)^Term 1 (EK1100)^Term 5 (EL4132)^Channel 1^Output;
  .nRawECatState:=TIID^Device 3 (EtherCAT)^Term 1 (EK1100)^Term 5 (EL4132)^InfoData^State'}
Term_5_EL4132_Chnl : FB_BACnet_AO_ECAT := (iParent:=View, sObjectName:='Term_5_EL4132_Chnl',
nIoBusNr:=0, sDeviceType:='EL4132 2Ch. Ana. Output +/-10V');

// Bus Terminals (K-Bus)

(* Term 2 (KL1104) *)
{attribute 'TcLinkTo' :=
  '.bRawVal :=TIID^Device 3 (EtherCAT)^Box 6 (BK1150)^Term 2 (KL1104)^Channel 1^Input'}
Term_2_KL1104_Chnl : FB_BACnet_BI_IO := (iParent:=View, sObjectName:='Term_2_KL1104_Chnl',
nIoBusNr:=0, sDeviceType:='KL 1104, 4 Ch. Input (24V, 3.0ms)');

(* Term 3 (KL2408) *)
{attribute 'TcLinkTo' :=
  '.bRawVal :=TIID^Device 3 (EtherCAT)^Box 6 (BK1150)^Term 3 (KL2408)^Channel 1^Output'}
Term_3_KL2408_Chnl : FB_BACnet_BO_IO := (iParent:=View, sObjectName:='Term_3_KL2408_Chnl',
nIoBusNr:=0, sDeviceType:='KL 2408, 8 Ch. Output (24V, 0.5 A, 3 A max)');
```

```
(* Term 4 (KL3208-0010-4) *)
{attribute 'TcLinkTo' :=
  '.nRawVal :=TIID^Device 3 (EtherCAT)^Box 6 (BK1150)^Term 4 (KL3208-0010-4)^Channel 1^Data In;
  .nRawState:=TIID^Device 3 (EtherCAT)^Box 6 (BK1150)^Term 4 (KL3208-0010-4)^Channel 1^State'}
Term_4_KL3208_0010_4_Chnl : FB_BACnet_AI_IO := (iParent:=View,
sObjectName:='Term_4_KL3208_0010_4_Chnl', nIoBusNr:=0, sDeviceType:='KL 3208-0010, 4 of 8 Ch. ana.
Input PT1000, Ni1000 (RTD)');

(* Term 6 (KL4022) *)
{attribute 'TcLinkTo' :=
  '.nRawVal :=TIID^Device 3 (EtherCAT)^Box 6 (BK1150)^Term 6 (KL4022)^Channel 1^Data Out'}
Term_6_KL4022_Chnl : FB_BACnet_AO_IO := (iParent:=View, sObjectName:='Term_6_KL4022_Chnl',
nIoBusNr:=0, sDeviceType:='KL 4022, 2 Ch. ana. Output (4...20mA)');
```

Code

```
View();
Term_2_EL1809_Chnl();
Term_3_EL2809_Chnl();
Term_4_EL3068_Chnl();
Term_5_EL4132_Chnl();
Term_2_KL1104_Chnl();
Term_3_KL2408_Chnl();
Term_4_KL3208_0010_4_Chnl();
Term_6_KL4022_Chnl();
```

9.4 Property Auswahl und Schreibschutz

Dieses Beispiel zeigt, wie Eigenschaften der BACnet-Objekte (Properties) mit einem Schreibschutz versehen werden können. Weiterhin zeigt das Beispiel, wie nicht benötigte Properties aus Objekten entfernt werden können.

Als Vorgabewert sind alle Properties (die nach dem BACnet-Standard schreibbar sein dürfen), schreibbar ausgeführt. Ebenfalls sind alle im jeweiligen Objekttyp möglichen Properties enthalten.

Properties, die nur lesbar sein sollen, werden im Array `aWriteProtected` aufgeführt. Properties, die nicht verwendet werden und damit aus dem Objekt entfernt werden sollen, werden im Array `aDisabled` aufgeführt.



Zusammenhängende Properties (diese sind im Standard mit einer gleichlautenden Fußnote gekennzeichnet), müssen gesamthaft vorhanden sein oder gesamthaft entfernt werden. Im Beispiel unten werden die Properties `ChangeOfStateCount`, `ChangeOfStateTime` und `TimeOfStateCountReset` deaktiviert.

Variablen

```
fbBi : FB_BACnet_BI := (
  sObjectName := 'Example Binary Input Object',
  sDescription := 'Objectname and Description properties are read-only',
  stSettings := (
    aDisabled := [
      E_BACnetPropIdentifier.PropChangeOfStateCount,
      E_BACnetPropIdentifier.PropChangeOfStateTime,
      E_BACnetPropIdentifier.PropTimeOfStateCountReset
    ],
    aWriteProtected := [
      E_BACnetPropIdentifier.PropObjectName,
      E_BACnetPropIdentifier.PropDescription
    ]
  )
);
```

Code

```
fbBi();
```

9.5 Prioritätssteuerung

Dieses Beispiel zeigt die Verwendung der Bausteine mit der Endung `_5P` (5 Prioritäten). Von den zur Verfügung stehenden 16 Prioritäten des BACnet-Standards, wurden 5 Prioritäten ausgewählt, die in den meisten Fällen für Gebäudeautomationsprojekte ausreichen:

LifeSafety (1): Zum Beispiel Notabschaltung

Critical Equipment Control (5): Zum Beispiel Frostschutz

Manual Local Operator (7): Zum Beispiel Vor-Ort Bedienung am Schaltschrank

Manual Operator (8): Zum Beispiel Bediener an der BACnet Management Bedienebene

SPS (15): Priorität des SPS-Programms

Die Zahl in Klammern gibt den Defaultwert der Priorität an. Dieser kann global in der Sektion `BACnet_Param` der Bibliothek `Tc3_BACnetRev14` geändert werden.

Für die Steuerung der Priorität stehen die folgenden booleschen Flags zur Verfügung:

bEnSfty: Übersteuerung der LifeSafety Priorität durch das SPS-Programm. Der Wert wird durch die Variablen `fValSfty` (Analog), `bValSfty` (Binär) und `nValSfty` (Multistate) vorgegeben.

bEnCrit: Übersteuerung der Critical Equipment Priorität. Der Wert wird durch die Variablen `fValCrit` (Analog), `bValCrit` (Binär) und `nValCrit` (Multistate) vorgegeben.

bEnManLoc: Übersteuerung der Manual Operator Priorität. Der Wert wird durch die Variablen `fValManLoc` (Analog), `bValManLoc` (Binär) und `nValManLoc` (Multistate) vorgegeben.

bEnPgm: Übersteuerung der SPS-Priorität. Der Wert wird durch die Variablen `fValPgm` (Analog), `bValPgm` (Binär) und `nValPgm` (Multistate) vorgegeben.

bEnManualOperator: Übersteuerung der Bediener Priorität. Der Wert wird durch die Variablen `fValManualOperator` (Analog), `bValManualOperator` (Binär) und `nValManualOperator` (Multistate) vorgegeben.



Diese Prioritätsstufe wird in BACnet-Projekten für den Zugriff durch die MBE (Management Bedien Einrichtung) verwendet und sollte daher nur verwendet werden, wenn dieser Zugriff seitens BACnet nicht erfolgen soll oder bei Fehlen einer MBE.

Variablen

```
// Analog objects supporting 5 priorities
fbAO5P : FB_BACnet_AO_5P;
fbAOIO5P : FB_BACnet_AO_IO5P;
fbAORaw5P : FB_BACnet_AO_RAW5P;
fbAV5P : FB_BACnet_AV_5P;

// Binary objects supporting 5 priorities
fbBO5P : FB_BACnet_BO_5P;
fbBOIO5P : FB_BACnet_BO_IO5P;
fbBORaw5P : FB_BACnet_BO_RAW5P;
fbBV5P : FB_BACnet_BV_5P;

// Multistate objects supporting 5 priorities
fbMO5P : FB_BACnet_MO_5P;
fbMOIO5P : FB_BACnet_MO_IO5P;
fbMORaw5P : FB_BACnet_MO_RAW5P;
fbMV5P : FB_BACnet_MV_5P;
```

Code

```
fbAO5P();
fbAOIO5P();
fbAORaw5P();
fbAV5P();
fbBO5P();
fbBOIO5P();
fbBORaw5P();
```



```
fbBV5P();
fbMO5P();
fbMOIO5P();
fbMORaw5P();

// example access to critical equipment control priority
fbMV5P.bEnCrit := TRUE;
fbMV5P.nValCrit := 3; // select state no. 3
fbMV5P();
```

9.6 Prioritäten zurücksetzen

In einigen Fällen kann es notwendig sein, über BACnet beschriebene Prioritätsstufen von der SPS oder Visualisierung aus zurückzusetzen (d. h. den Wert NULL auf diese Prioritätsstufe zu schreiben). Dies kann Server-seitig mit dem Aufruf `WritePropertyNull` erfolgen.

Dabei ist zu beachten, dass bei Binärobjekten die Prioritätsstufe 6 für die minimalen Ein- und Ausschaltzeiten reserviert sind und daher nicht schreibbar sind.

Variablen

```
fbBV          : FB_BACnet_BV;
nCount        : INT;
bEmptyPrioArray : BOOL;
nRet          : DINT;
```

Code

```
fbBV();
IF bEmptyPrioArray THEN
  bEmptyPrioArray := FALSE;
  FOR nCount := 1 TO 16 DO
    IF nCount = 6 THEN
      CONTINUE;
    END_IF
    nRet := fbBV.WritePropertyNull( E_BACnetPropIdentifier.PropPresentValue, bPrio :=
TO_BYTE( nCount) );
  END_FOR
END_IF
```

9.7 Regelkreise und Loop Objekte

Für die Abbildung von Regelkreisen stellt der BACnet Standard das Loop-Objekt bereit. Dabei werden die Regelparameter, wie z. B. P, I, D, Begrenzungen des Regelbereichs oder die Wirkrichtung als Properties des Loop Object Typs an den Reglerbaustein weitergegeben. Der Typ des Reglers selbst ist im BACnet-Standard nicht vorgegeben.

In der Bibliothek `Tc3_BA2_Common` findet sich die Implementierung des PID-Reglers (`FB_BA_PIDCtrl`), welcher über die Bausteine `FB_BACnet_Loop` bzw. `FB_BACnet_LoopRef` abgebildet wird.

`FB_BACnet_Loop` implementiert einen Regler, dessen Sollwert, Ausgangswert sowie Istwert, über Baustein-interne Variablen abgebildet wird.

`FB_BACnet_LoopRef` verwendet als Sollwert eine Referenz auf ein Analog Value Objekt, als Ausgangswert eine Referenz auf ein Analog Output Objekt sowie für den Istwert eine Referenz auf ein Analog-Input-Objekt.

Während für die meisten Anwendungen der Bausteintyp `FB_BACnet_Loop` ausreicht, kann dem Bediener der Management Bedienebene mit dem Bausteintyp `FB_BACnet_LoopRef` der Zugriff auf die Reglerwerte (Sollwert, Ausgangswert und Istwert) über BACnet ermöglicht werden. Damit erhöht sich jedoch die Anzahl der BACnet-Objekte pro Regelkreis auf 4.

Die nachfolgenden Beispiele zeigen exemplarisch die Realisierung eines PI-Reglers.

Variablen

```
// control loop using internal variables
fbLoopInternal : FB_BACnet_Loop := (
  bEn := TRUE,
  sDescription := 'Loop using internal control parameters',
```

```

        eOutputUnit := E_BA_Unit.eOther_Percent,
        eAction := E_BA_Action.eReverse,
        fProportionalConstant := 5.0,
        fIntegralConstant := 180,
        fSetpoint := 20
    );
fCtrlVal : REAL := 18;

// control loop using external BACnet objects
fbLoopRef_Setpt : FB_BACnet_AV_Setpt := (
    fValue := 20 );
fbLoopRef_CtrlVar : FB_BACnet_AI := (
    fVal := 18 );
fbLoopRef_Y : FB_BACnet_AO := ();
fbLoopRef : FB_BACnet_Loop_Ref := (
    bEn := TRUE,
    sDescription := 'Loop using reference objects',
    stControlledVariableReference :=
        F_BACnet_Reference( fbLoopRef_CtrlVar, PropPresentValue),
    stSetpointReference :=
        F_BACnet_Reference( fbLoopRef_Setpt, PropPresentValue),
    stManipulatedVariableReference :=
        F_BACnet_Reference( fbLoopRef_Y, PropPresentValue),
    eOutputUnit := E_BA_Unit.eOther_Percent,
    eAction := E_BA_Action.eReverse,
    fProportionalConstant := 5.0,
    fIntegralConstant := 180
);

```

Code

```

// internal control loop
fbLoopInternal.fCtrlVar := fCtrlVal;
fbLoopInternal();

// control loop using external object references
fbLoopRef_Setpt();
fbLoopRef_CtrlVar();
fbLoopRef_Y();
fbLoopRef();

```

9.8 Alarmempfänger einrichten

BACnet ermöglicht Alarmempfängern (z. B. der MBE) sich in ein oder mehrere Notification Class Objekte in die Empfängerliste (Recipientlist) einzutragen.

Stehen zum Zeitpunkt der SPS-Programmierung die Empfänger bereits fest, können diese auch seitens des SPS-Programms vorgegeben werden.

Als Referenz auf den Empfänger kann entweder die Device-Instanznummer verwendet werden. In diesem Fall löst TwinCAT die tatsächliche Adresse mit einem Who-Is Telegramm auf (die Antwort I-Am des Empfängers enthält die tatsächliche Adresse).

Alternativ kann diese Adresse direkt in der Empfängerliste angegeben werden, jedoch wird diese Möglichkeit in der Praxis eher selten verwendet.

Das nachfolgende Beispiel zeigt eine Notification Class NC1 ohne Empfängervorgaben (d. h. die Empfänger tragen sich über BACnet ein). Die Notification Class 2 enthält ein Beispiel für einen Empfänger, der mit der Device-Instanznummer referenziert wird, als auch einen Empfänger, der über die IP-Adresse, den UDP-Port und die BACnet-Netzwerknummer referenziert wird.

Die Reihenfolge der Alarmarten in den Arrays `aEventEnable` und `aEventMessageTextsConfig` sowie `aAckRequired` und `aPriority` entspricht der Reihenfolge im BACnet-Standard:

TO_OFFNORMAL: Kommender Alarm / Ereignis

TO_FAULT: Sensorfehler, Geberstörung

TO_NORMAL: Rückkehr in den Normalbereich

Variablen

```
// simple Notification class object with empty recipient list
fbBV1 : FB_BACnet_BV := (
  nNotificationClass := 1,
  aEventEnable := [TRUE,TRUE,TRUE],
  bAlarmValue := TRUE,
  aEventMessageTextsConfig := [ 'ALARM', 'FAULT', 'NORMAL' ]
);
fbNC01_Standard : FB_BACnet_NC := (
  nObjectInstance := 1,
  nNotificationClass := 1,
  sDescription := 'NC01 Standard',
  aAckRequired := [ TRUE, TRUE, FALSE ],
  aPriority := [ 10, 11, 12 ]
);

// Notification class object with pre-defined recipient (Notification Sink module)
fbBV2 : FB_BACnet_BV := (
  nNotificationClass := 2,
  aEventEnable := [TRUE,TRUE,TRUE],
  bAlarmValue := TRUE
);
fbNC02_Recipient : FB_BACnet_NC := (
  nObjectInstance := 2,
  nNotificationClass := 2,
  sDescription := 'NC02 RecipientTest',
  aAckRequired := [ TRUE, TRUE, TRUE ],
  aPriority := [ 224, 223, 222 ],
  aRecipientList := [
    (
      stValidDays := (bMonday:=TRUE, bTuesday:=TRUE, bWednesday:=TRUE, bThursday:=TRUE,
bFriday:=TRUE),
      stFromTime := F_BA_ToSTTime(T#0H),
      stToTime := F_BA_ToSTTime(T#23H59M59S),
      stRecipient := F_BACnet_DeviceRecipient(nDeviceInstance:=42),
      nProcessId := 10000, // Notification Sink module
      bIssueConfirmed := FALSE,
      stEventTransitions := (bToOffNormal:=TRUE, bToFault:=TRUE, bToNormal:=TRUE )
    ),
    (
      stValidDays := (bSunday:=TRUE, bSaturday:=TRUE),
      stFromTime := F_BA_ToSTTime(T#7H),
      stToTime := F_BA_ToSTTime(T#15H30M),
      stRecipient := F_BACnet_EthernetRecipient(
        nIPAddress1:=192,168,10,200,
        nPort:=47808,
        nNetworkNr:=444
      ),
      nProcessId := 30100,
      bIssueConfirmed := TRUE,
      stEventTransitions := (bToOffNormal:=TRUE)
    )
  ]
);
```

Code

```
fbBV1();
fbNC01_Standard();

fbBV2();
fbNC02_Recipient();
```

9.9 Empfang von Alarmen und Events aus anderen Geräten

Dieses Beispiel zeigt die Verwendung eines Eventlog-Objektes als Empfänger für Alarme und Ereignisse eines externen BACnet-Gerätes. Dabei wird in die Notification Class des Alarmgebers das Eventlog-Objekt des Alarmempfängers eingetragen. Die Zuordnung erfolgt dabei über den Process Identifier.

Gerät 1: Alarmgeber

Variablen

```
// Notification Class object in the alarm generating device
fbNC01 : FB_BACnet_NC := (
  nObjectInstance := 1,
  nNotificationClass := 1,
  sDescription := 'NC01',
  aAckRequired := [ TRUE, TRUE, TRUE ],
  aPriority := [ 224, 223, 222 ],
  aRecipientList :=
  [
    (
      stValidDays := (
        bMonday:=TRUE,
        bTuesday:=TRUE,
        bWednesday:=TRUE,
        bThursday:=TRUE,
        bFriday:=TRUE,
        bSaturday:=TRUE,
        bSunday:=TRUE
      ),
      stFromTime := F_BA_ToSTTime(T#0H),
      stToTime := F_BA_ToSTTime(T#23H59M59S),
      stRecipient := F_BACnet_DeviceRecipient(nDeviceInstance:=12345),
      nProcessId := 42,
      bIssueConfirmed := FALSE,
      stEventTransitions := (bToOffNormal:=TRUE, bToFault:=TRUE, bToNormal:=TRUE)
    )
  ]
);
```

Gerät 2: Alarmempfänger (Device 12345)

Variablen

```
fbELogBuf : FB_BACnet_ELogBuf := (
  sObjectName := 'Event Log for external alarms',
  bLogEnable := TRUE,
  nProcessId := 42
);
```

9.10 Vorwarngrenzen

Mit Hilfe des EventEnrollment-Objekts erlaubt BACnet die Überwachung von Properties aus BACnet-Objekten. Ein typischer Anwendungsfall in der Gebäudeautomation sind Vorwarngrenzen, die auf einen möglicherweise später folgenden Alarm aufmerksam machen.

Das nachfolgende Beispiel zeigt, wie die Objekt-interne Alarmerkennung (Intrinsic Reporting) mit einem zusätzlichen Paar Vorwarngrenzen kombiniert wird (Algorithmic Change Reporting) und diese Zustände (Vorwarnung und Alarm) über zwei unterschiedliche Notification Class-Objekte gemeldet werden können.

Dabei dient das BACnet-Objekt fbAV, dessen Present Value aus der REAL-Variablen fRealValue vorgegeben wird, als Auslöser für die Warnungen und Alarme.

Variablen

```
fRealValue : REAL := 50;

// Analog Value object using Intrinsic Reporting
fbAv : FB_BACnet_AV := (
  sObjectName := 'AV using Intr. Reporting',
  aEventEnable := [ TRUE, TRUE, TRUE ],
  eNotifyType := E_BACnet_NotifyType.eAlarm,
  bLowLimitEnable := TRUE,
  fLowLimit := 15.0,
  bHighLimitEnable := TRUE,
  fHighLimit := 87.0,
  fDeadband := 3,
  nTimeDelay := 2, // waits 2 seconds before TO_OFFNORMAL
  nTimeDelayNormal := 4, // wait 4 seconds before TO_NORMAL
  bEventDetectionEnable := TRUE,
  nNotificationClass := 10, // alarm class
  aEventMessageTextsConfig := [ 'Alarm', 'Fault', 'Warning' ],
  bEnPgm := TRUE,
  eUnit := E_BA_Unit.eOther_Percent
);
```

```
// Additional warning limits using Algorithmic Change Reporting
fbEE : FB_BACnet_EE := (
    sObjectName := 'Event Enrollment',
    nNotificationClass := 20,
    eNotifyType := E_BACnet_NotifyType.eNotifyEvent,
    aEventEnable := [ TRUE, TRUE, TRUE ],
    aEventMessageTextsConfig := [ 'Warning', 'Fault', 'Normal' ],
    stEventParameter := (
        eEventType := E_BACnet_EventType.eOutOfRange,
        stEventArgs := (
            stOutOfRange := (
                nTimeDelay := 0,
                fLowLimit := 25.0,
                fHighLimit := 82.0,
                fDeadband := 0.0
            )
        )
    ),
    stObjectPropertyReference := F_BACnet_Reference(fbAv, PropPresentValue)
);

// Notification Classes 10=alarms, 20=warnings
fbNC10 : FB_BACnet_NC := (
    sObjectName := 'NC10',
    sDescription := 'Alarms',
    nObjectInstance := 10,
    nNotificationClass := 10,
    aAckRequired := [ TRUE, TRUE, TRUE ],
    aPriority := [ 10, 11, 12 ]
);
fbNC20 : FB_BACnet_NC := (
    sObjectName := 'NC20',
    sDescription := 'Warnings',
    nObjectInstance := 20,
    nNotificationClass := 20,
    aAckRequired := [ FALSE, TRUE, FALSE ],
    aPriority := [ 100, 110, 120 ]
);
```

Code

```
fbAv.fValPgm := fRealValue;
fbAv();
fbEE();
fbNC10();
fbNC20();
```

9.11 Kalender und Zeitschaltfunktionen

Für die Umsetzung von Zeitschaltfunktionen stellt der BACnet Standard das Schedule-Objekt bereit. Es erlaubt die Verwendung wiederkehrender Wochenprogramme (Weekly_Schedule) als auch die Festlegung von Ausnahmen (Exception_Schedule). Ausnahmen können entweder direkt im Schedule-Objekt eingetragen werden oder diese werden auf Basis eines Calendar-Objekts (das sich im gleichen Gerät befinden muss) ermittelt. Für die Festlegung einer Ausnahme können ein einzelnes Datum, ein Datumsbereich oder ein kombinierter Typ aus Tag, Woche und Monat verwendet werden.

Der Wochenzeitplan wird über das Array `aWeek` notiert.

Kalender-basierte Ausnahmen werden im Array `aCalendar` notiert.

Im Zeitplan direkt enthaltene Ausnahmen werden über das Array `aException` notiert.

Für die Festlegung der Datumsangaben stehen Hilfsfunktionen zur Verfügung, z. B. `F_BA_DateVal`.

Bei Datumsangaben gilt gemäß BACnet-Standard, dass das Jahr ab 1900 beginnt. Das Element Monat kann zusätzlich zu den regulären Monatsangaben auch ungerade (13) und gerade (14) Monate enthalten. Der Tag des Monats kann zusätzlich zu den regulären Tagesangaben auch ungerade (33) und gerade (34) Tage des Monats sowie den letzten Tag (32) des Monats enthalten.

Wichtig ist bei der Angabe spezifischer Datumsangaben, dass der Wochentag zum angegebenen Datum passen muss.

Für die Festlegung der Zeit/Werte-Paare des Zeitschaltplans stehen ebenfalls Funktionen zur Verfügung, z. B. `F_BACnet_SchedWeekly3xA`. Dabei bezeichnet die Angabe `3x` die Anzahl (3 Einträge). Das Kennzeichen `A` steht für Analogwerte (REAL), das Kennzeichen `B` für Binärwerte (BOOL) und `M` für Multistate-Zustände (Integer).

Soll der Zeitschaltplan direkt ein oder mehrere Properties aus BACnet-Objekten beeinflussen, können diese Referenzen im Element `aObjectPropertyReferences` angegeben werden. Fehlt diese Angabe, wird eine leere Liste im Property List of `ObjectPropertyReferences` erzeugt und der Zeitplan greift nicht auf externe Objekte/Properties ein. Dieses kann verwendet werden, wenn innerhalb der SPS der Zustand des Zeitplans überwacht werden soll, aber keine BACnet-Objekte direkt beeinflusst werden sollen.

Variablen

```
// Calendar object examples
fbCAL01 : FB_BACnet_Cal := (
    sObjectName := 'Calendar 1',
    sDescription := 'demonstrates a date-list for each choice',
    aDateList := [
        (
            eType := E_BA_DateValChoice.eDate,
            uDate := F_BA_DateVal(2021,E_BA_Month.eJanuary,19)
        ),
        (
            eType := E_BA_DateValChoice.eDateRange,
            uDate := F_BA_DateRangeVal(nFromYear:=2021,E_BA_Month.eJanuary,19,
nToYear:=2021,E_BA_Month.eJanuary,21)
        ),
        (
            eType := E_BA_DateValChoice.eWeekNDay,
            uDate := F_BA_WeekNDayVal(E_BA_Weekday.eFriday, E_BA_Week.eWeek1,
E_BA_Month.eFebruary)
        )
    ]
);
fbCAL02 : FB_BACnet_Cal := (
    sObjectName := 'Calendar 2'
);
fbCAL03 : FB_BACnet_Cal := (
    sObjectName := 'Calendar 3'
);

// schedule object for analog scheduling
fbAnalogOutput : FB_BACnet_AO;
fbSchedA : FB_BACnet_SchedA := (
    sObjectName := 'Schedule Analog',
    aObjectPropertyReferences :=
    [
        (iObject := fbAnalogOutput, ePropertyId := PropPresentValue)
    ],
    aWeek := F_BACnet_SchedWeekly3xA(E_BA_Weekday.eMonday, E_BA_Weekday.eFriday, T#0H, 0.0, T#6H,
5.0, T#20H, 0.0),
    aCalendar := [(
        iRefCalendar := fbCAL01,
        aEntry := F_BACnet_SchedEntry1xA(T#0H, 2.0)
    ),
    (
        iRefCalendar := fbCAL02,
        aEntry := F_BACnet_SchedEntry1xA(T#10H, 3.4)
    ),
    (
        iRefCalendar := fbCAL03,
        aEntry := F_BACnet_SchedEntry3xA(T#0H, 5, T#6H, 7, T#20H, 8)
    )],
    aException := [
        (
            eType := E_BA_DateValChoice.eDate,
            uDate := F_BA_DateVal(2020,E_BA_Month.eApril,10),
            aEntry := F_BACnet_SchedEntry1xA(T#0H, 0.0)
        ),
        (
            eType := E_BA_DateValChoice.eDateRange,
            uDate := F_BA_DateRangeVal(nFromYear:=2020,E_BA_Month.eApril,10,
nToYear:=2021,E_BA_Month.eMay,11),
            aEntry := F_BACnet_SchedEntry3xA(T#0H, 0.0, T#6H, 5.0, T#20H, 0.0)
        )
    ],
);
```

```

    (
        eType := E_BA_DateValChoice.eWeekNDay,
        uDate := F_BA_WeekNDayVal(E_BA_Weekday.eFriday, E_BA_Week.eWeek1, E_BA_Month.eFebruary),
        aEntry := F_BACnet_SchedEntry3xA(T#0H, F_BA_NullA(), T#6H, 5.0, T#20H, F_BA_NullA())
    ),
    (
        eType := E_BA_DateValChoice.eDate,
        uDate := F_BA_DateVal(2019, E_BA_Month.eJune, 20),
        aEntry := [
            (
                eState := E_BA_SchedEntryState.eValue,
                stTime := F_BA_ToSTime(T#10H),
                uValue := F_BA_RVal(1.0)
            ),
            (
                eState := E_BA_SchedEntryState.eNull,
                stTime := F_BA_ToSTime(T#11H)
            )
        ]
    )
]);

// schedule object for binary scheduling
bScheduledValue : BOOL;
fbSchedB : FB_BACnet_SchedB := (
    sObjectName := 'Schedule Bool',
    aWeek := F_BACnet_SchedWeekly3xB(E_BA_Weekday.eMonday, E_BA_Weekday.eFriday, T#0H, FALSE, T#6H,
TRUE, T#20H, FALSE),
    aCalendar := [
        (
            iRefCalendar := fbCAL01,
            aEntry := F_BACnet_SchedEntry1xB(T#0H, FALSE)
        ),
        (
            iRefCalendar := fbCAL02,
            aEntry := F_BACnet_SchedEntry1xB(T#10H, TRUE)
        ),
        (
            iRefCalendar := fbCAL03,
            aEntry := F_BACnet_SchedEntry3xB(T#0H, F_BA_NullB(), T#6H, TRUE, T#20H, FALSE)
        )
    ]
]);

// schedule object for multistate scheduling
nMultistateValue : UDINT;
bSchedM_AssignCalReference : BOOL;
fbSchedM : FB_BACnet_SchedM :=
    (
        sObjectName := 'Schedule Multistate',
        aWeek := F_BACnet_SchedWeekly3xM(E_BA_Weekday.eMonday, E_BA_Weekday.eFriday, T#0H, 1, T#6H,
2, T#20H, 1),
        nScheduleDefault := 3
    );

```

Code

```

fbCAL01();
fbCAL02();
fbCAL03();

fbAnalogOutput();
fbSchedA();

fbSchedB();
bScheduledValue := fbSchedB.bPresVal;

// example how to assign a calendar reference at runtime
IF( bSchedM_AssignCalReference ) THEN
    bSchedM_AssignCalReference := FALSE;
    fbSchedM.aCalendar[1].iRefCalendar := fbCAL01;
    fbSchedM.aCalendar[1].aEntry := F_BACnet_SchedEntry1xM(T#11H, 2);
    fbSchedM.bWriteException := TRUE;
END_IF
fbSchedM();
nMultistateValue := fbSchedM.nPresVal;

```

9.12 Logging Objekte

BACnet erlaubt die Verwendung von drei verschiedenen Objekten zur Speicherung von Logdaten:

Trendlog: Dieser Objekttyp repräsentiert die aufgezeichneten Logdaten einer einzelnen Datenquelle, z. B. dem Present Value eines Analog Input Objektes.

Dabei kann die Aufzeichnung per Polling (d. h. in einem festgelegten Aufzeichnungsintervall in 1/100s), per Change of Value (COV) oder mit Hilfe eines booleschen Triggers erfolgen.

Trendlog Multiple: Dieser Objekttyp erlaubt die gleichzeitige Aufzeichnung von Daten aus mehreren Quellen. Prinzipbedingt ist die Aufzeichnung nur per Polling oder Trigger, nicht jedoch per COV möglich. In der Praxis wird dieser Objekttyp in BACnet-Projekten eher selten verwendet.

Eventlog: Dieser Objekttyp erlaubt die Speicherung von BACnet-Ereignismeldungen (Events und Alarmer). Existiert für die Instanznummer eines Eventlog-Objektes ein Notification Class Objekt mit derselben Instanznummer im Server, so wird das Notification Class Objekt automatisch als Datenquelle des Eventlog Objektes konfiguriert. Alarmer, die über diese Notification Class gemeldet werden, werden also automatisch im Logspeicher des Eventlogobjektes gespeichert.

Das nachfolgende Beispiel zeigt die Verwendung dieser drei Objekttypen.

Variablen

```
fAnalogValue : REAL;
fbTon : TON;
fbAv : FB_BACnet_AV := (
    sObjectName := 'Object to simulate changes and generate alarms',
    fLowLimit := 1.0,
    bLowLimitEnable := TRUE,
    fHighLimit := 8.0,
    bHighLimitEnable := TRUE,
    bEnPgm := TRUE,
    bEventDetectionEnable := TRUE,
    nNotificationClass := 42,
    aEventEnable := [TRUE, TRUE, TRUE],
    eUnit := E_BA_Unit.eElectrical_Volts,
    nTimeDelay := 0,
    nTimeDelayNormal := 0,
    fDeadband := 0.0
);

fbNC42 : FB_BACnet_NC := (
    nObjectInstance := 42,
    nNotificationClass := 42,
    sDescription := 'NC42',
    aAckRequired := [ TRUE, TRUE, TRUE ],
    aPriority := [ 1, 2, 3 ]
);

fbELog_NC42 : FB_BACnet_ELogBuf := (
    sObjectName := 'Event Log for NC42',
    nObjectInstance := 42,
    bLogEnable := TRUE
);

fbTLog : FB_BACnet_TLog := (
    sObjectName := 'Trend',
    nNotificationClass := 42,
    aEventEnable := [ TRUE, TRUE, FALSE ],
    stStartTime := F_BA_ToSTDateTime(DT#2010-01-01-00:00),
    bLogEnable := TRUE,
    eLoggingType := E_BA_LoggingType.ePolled,
    nLogInterval := 3 * 100, // log every 3 seconds, value in 1/100 seconds!
    stObjectPropertyReference := F_BACnet_Reference(fbAv, PropPresentValue)
);

fbTLogCov : FB_BACnet_TLog := (
    sObjectName := 'Trend Cov',
    bLogEnable := TRUE,
    stObjectPropertyReference := F_BACnet_Reference(fbAv, PropPresentValue),
    eLoggingType := E_BA_LoggingType.eCOV,
    nCOVResubscriptionInterval := 600,
    stClientCOV := (
        eChoice := E_BACnet_ClientCOVChoice.eCovReal,
        fIncrement := 5.0
    )
);
```



```

    );

    fbTLogBuf : FB_BACnet_TLogBuf := (
        sObjectName := 'Trend with PLC buffer',
        bLogEnable := TRUE,
        eLoggingType := E_BA_LoggingType.eTriggered,
        stObjectPropertyReference := F_BACnet_Reference(fbAv, PropPresentValue)
    );

    fbTLogM : FB_BACnet_TLM := (
        sObjectName := 'Trend Multiple',
        bLogEnable := TRUE,
        eLoggingType := E_BA_LoggingType.ePolled,
        nLogInterval := 50, // log every 0.5 seconds, value in 1/100 seconds!
        aObjectPropertyReferences := [
            (iObject := fbAv, ePropertyId := PropPresentValue),
            (iObject := fbAv, ePropertyId := PropStatusFlags),
            (iObject := fbAv, ePropertyId := PropEventState) ]
    );

```

Code

```

// generate value changes every second
fbTon( IN:= NOT fbTon.Q, PT:=T#1S );
IF fbTon.Q THEN
    fAnalogValue := fAnalogValue + 1.0;
    IF fAnalogValue > 10.0 THEN
        fAnalogValue := 0;
    END_IF
END_IF
fbAv.fValPgm := fAnalogValue;
fbAv();

// Notification Class
fbNC42();

// Event Log with buffer in PLC
fbELog_NC42();

// Trend Log every 3 seconds
fbTLog();

// COV based Trend Log
fbTLogCov();

// Trigger based Trend Log with buffer in PLC
fbTlogBuf();

// Trendlog Multiple
fbTLogM();

```

9.13 Verarbeitung des Logspeichers in der SPS

BACnet definiert für verschiedene Properties den Datentyp CHOICE. Dieser repräsentiert eine Auswahl verschiedener möglicher Propertywerte.

Um diese Datentypen in der SPS verarbeiten zu können, wird der SPS-Datentyp UNION zusammen mit einer Enumeration verwendet. Dabei enthält die Enumeration die Information, auf welches Element der Union zugegriffen werden muss.

Logspeicher des Trendlog-Objektes

Im nachfolgenden Beispiel wird die Verarbeitung des Logspeichers eines Trendlog-Objektes dargestellt.

Dabei kommt ein Objekt vom Typ FB_BACnet_TLogBuf zum Einsatz. Dieses stellt einen Logspeicher bereit, dessen Inhalt von der SPS ausgelesen werden kann. Die Variable aLogBuffer steht dabei als Array vom Typ T_BACnet_TLogBuffer zur Verfügung. Die einzelnen Elemente des Logbuffer sind vom Typ ST_BA_TrendEntry mit folgenden Elementen:

dtTime: Enthält das Datum und die Uhrzeit des Logeintrags im BACnet Format (ST_BA_Date und ST_BA_Time).

eType: Enthält die Information über den Typ des Logeintrags:

eBinary: Der Logeintrag enthält einen Wert vom Typ BinaryPV. Im Element uValue ist der Wert der Variable bVal zu verarbeiten.

eAnalog: Der Logeintrag enthält einen Wert vom Typ REAL. Im Element uValue ist der Wert der Variable fVal zu verarbeiten.

eMultistate: Der Logeintrag enthält einen Wert vom Typ UDINT. Im Element uValue ist der Wert der Variable udiVal zu verarbeiten.

eEvent: Der Logeintrag enthält ein Ereignis. In der Struktur stEvent stehen vier Bits zur Verfügung:

bStart: Die Aufzeichnung wurde gestartet.

bStop: Die Aufzeichnung wurde gestoppt.

bBufferPurged: Der Logspeicher wurde geleert.

bInterrupted: Die Aufzeichnung wurde unterbrochen (z. B. durch eine Netzwerkunterbrechung)

Zusätzlich stehen im Strukturelement stState die vier Statusflags zur Verfügung:

bInAlarm: Das Objekt befindet sich in einem aktiven Alarmzustand.

bFault: Das Objekt befindet sich in einem fehlerhaften Zustand, z. B. Geberstörung.

bOverridden: Das Objekt wurde durch einen internen Mechanismus, z. B. durch die SPS übersteuert.

bOutOfService: Das Objekt wurde außer Betrieb gesetzt. Werte werden simuliert.

Logspeicher des Eventlog-Objektes

Im nachfolgenden Beispiel wird die Verarbeitung des Logspeichers eines Eventlog-Objektes dargestellt.

Dabei kommt ein Objekt vom Typ FB_BACnet_ELogBuf zum Einsatz. Dieses stellt einen Logspeicher bereit, dessen Inhalt von der SPS ausgelesen werden kann. Die Variable aLogBuffer steht dabei als Array vom Typ T_BACnet_ELogBuffer zur Verfügung. Die einzelnen Elemente des Logbuffer sind vom Typ ST_BACnet_EventLogEntry mit folgenden Elementen:

dtTime: Enthält das Datum und die Uhrzeit des Logeintrags im BACnet Format (ST_BA_Date und ST_BA_Time).

eType: Enthält die Information über den Typ des Logeintrags:

eStatus: Der Logeintrag enthält die Information der StatusFlags. In der Struktur stStatus stehen vier Bits zur Verfügung:

bInAlarm: Das Objekt befindet sich in einem aktiven Alarmzustand.

bFault: Das Objekt befindet sich in einem fehlerhaften Zustand, z. B. Geberstörung.

bOverridden: Das Objekt wurde durch einen internen Mechanismus, z. B. durch die SPS übersteuert.

bOutOfService: Das Objekt wurde außer Betrieb gesetzt. Werte werden simuliert.

eTimesync: Es wurde eine Uhrzeitsynchronisationsnachricht empfangen. Das Element fTimeSync enthält das Delta zur vorherigen Uhrzeit.

eNotification: Der Logeintrag enthält eine Ereignismeldung. In der Struktur stNotification stehen die Parameter der Ereignisnachricht zur Verfügung, z. B. die Prozess-ID, das auslösende Gerät und Objekt, die Alarmpriorität, usw.

9.14 Primitive Value-Objekte

Primitive Value-Objekte können zur Abbildung einfacher Datentypen, wie z. B. Zeichenketten (Strings), Integerwerten oder für Datums- oder Zeitangaben verwendet werden.

Bei Datumsangaben gilt gemäß BACnet Standard, dass das Jahr ab 1900 beginnt. Das Element Monat kann zusätzlich zu den regulären Monatsangaben auch ungerade (13) und gerade (14) Monate enthalten. Der Tag des Monats kann zusätzlich zu den regulären Tagesangaben auch ungerade (33) und gerade (34) Tages des Monats sowie den letzten Tag (32) des Monats enthalten.

Wichtig ist bei der Angabe spezifischer Datumsangaben, dass der Wochentag zum angegebenen Datum passen muss.

Für Uhrzeiten gilt der Bereich von 0-23 Stunden, 0-59 Minuten, 0-59 Sekunden sowie 0-99 Hundertstel Sekunden.

Der Wert 255 steht als Platzhalter bei den Objekten vom Typ *Pattern* für einen beliebigen Wert (z. B. jedes Jahr oder jede Stunde). Für die Aufzählungstypen kann alternativ auch der enumertierte Wert *Unspecified* verwendet werden.

Variablen

```
fbPositiveInteger : FB_BACnet_INT;
fbLargeAnalog : FB_BACnet_LAV;
fbCharacterString : FB_BACnet_String;
fbUnsignedInteger : FB_BACnet_UINT;

nValue : INT := 15;
fValue : LREAL := 42.3;
sValue : STRING := 'TwinCAT BACnet';
uiValue : UINT := 12345;

fbDate : FB_BACnet_Date;
fbDatePattern : FB_BACnet_DateP;
fbDateTime : FB_BACnet_DateTime;
fbDatetimePattern : FB_BACnet_DateTimeP;
fbTime : FB_BACnet_Time;
fbTimePattern : FB_BACnet_TimeP;

// specific date
stDate : ST_BA_Date := ( nYear := 122, eMonth := E_BA_MONTH.eDecember, nDay := E_BA_DAY.eDay02,
eDayOfWeek := E_BA_WEEKDAY.eFriday );

// every year christmas eve (regardless of the day of week)
stDatePattern : ST_BA_Date := ( nYear := 255, eMonth := E_BA_MONTH.eDecember, nDay :=
E_BA_DAY.eDay24, eDayOfWeek := E_BA_WEEKDAY.Unspecified );

// specific date and specific time
stDateTime : ST_BA_DateTime := (
stDate := ( nYear := 122, eMonth := E_BA_MONTH.eDecember, nDay := E_BA_DAY.eDay02, eDayOfWeek :=
E_BA_WEEKDAY.eFriday ),
stTime := ( nHour := 17, nMinute := 53, nSecond := 42, nHundredths := 19 ) );

// every year where the 1st of May is a Monday each hour / minute at 11 seconds
stDateTimePattern : ST_BA_DateTime := (
stDate := ( nYear := 255, eMonth := E_BA_MONTH.eMay, nDay := E_BA_DAY.eDay01, eDayOfWeek :=
E_BA_WEEKDAY.eMonday ),
stTime := ( nHour := 255, nMinute := 255, nSecond := 11, nHundredths := 0 ) );

// specific time
stTime : ST_BA_Time := ( nHour := 11, nMinute := 42, nSecond := 38, nHundredths := 45 );

// every hour at minute 42
stTimePattern : ST_BA_Time := ( nHour := 255, nMinute := 42, nSecond := 0, nHundredths := 0 );
```

Code

```
fbPositiveInteger.nValue := nValue;
fbPositiveInteger();

fbLargeAnalog.fValue := fValue;
fbLargeAnalog();

fbCharacterString.sValue := sValue;
fbCharacterString();

fbUnsignedInteger.nValue := uiValue;
fbUnsignedInteger();

fbDate.stValue := stDate;
fbDate();
```

```
fbDatePattern.stValue := stDatePattern;
fbDatePattern();

fbDateTime.stValue := stDateTime;
fbDateTime();

fbDatetimePattern.stValue := stDateTimePattern;
fbDatetimePattern();

fbTime.stValue := stTime;
fbTime();

fbTimePattern.stValue := stTimePattern;
fbTimePattern();
```

9.15 Structured View-Objekte

Structured View-Objekte ermöglichen die Abbildung der Bedienersicht auf die in einem BACnet-Gerät vorhandenen Objekte, oft anhand eines Anlagenkennzeichnungssystems.

Mit Ausnahme der oberen Ebene (root level) werden alle Elemente mit Hilfe des Elements iParent mit dem jeweils übergeordneten Objekt verbunden.

Mit Hilfe der Zeichen \ (Backslash und Slash) kann eine Textzusammensetzung mit dem jeweils übergeordneten Objekt hergestellt werden.

Dabei wird für die Properties Objectname, Description und EventMessageTextsConfig das Trennzeichen verwendet, welches in den BACnet_Param der Bibliotheksinstanz festgelegt ist.

Für die Nutzung der Funktion STRING_TO_UTF8 muss die Bibliothek Tc2_Utilities eingebunden werden.

Variablen

```
// use \ to concat to the string provided by parent node
// countries (root level)
fbGermany : FB_BACnet_View := (
  eNodeType := E_BACnet_NodeType.eOrganizational,
  sObjectName := 'Germany',
  sDescription := 'BECKHOFF offices in Germany' );

fbSwitzerland : FB_BACnet_View := (
  eNodeType := E_BACnet_NodeType.eOrganizational,
  sObjectName := 'Switzerland',
  sDescription := 'BECKHOFF offices in Switzerland');

fbSpain : FB_BACnet_View := (
  eNodeType := E_BACnet_NodeType.eOrganizational,
  sObjectName := 'Spain',
  sDescription := 'BECKHOFF offices in Spain');

// view structure Verl
fbVerl : FB_BACnet_View := (
  iParent := fbGermany,
  eNodeType := E_BACnet_NodeType.eOrganizational,
  sObjectName := '\Verl',
  sDescription := '\Verl offices');

fbEiserstr : FB_BACnet_View := (
  iParent := fbVerl,
  eNodeType := E_BACnet_NodeType.eOrganizational,
  sObjectName := '\Eiserstr',
  sDescription := '\Eiserstr offices');

fbFirstFloor : FB_BACnet_View := (
  iParent := fbEiserstr,
  eNodeType := E_BACnet_NodeType.eOrganizational,
  sObjectName := '\Floor 2',
  sDescription := '\Hardware Development');

fbCabinet : FB_BACnet_View := (
  iParent := fbFirstFloor,
  eNodeType := E_BACnet_NodeType.eCollection,
  sObjectName := '\Controllers',
```

```

    sDescription := '\Cabinet 07');

fbAv01 : FB_BACnet_AV := (
    sObjectName := '\AV01',
    sDescription := '\Analog Value01',
    iParent := fbCabinet );

fbController : FB_BACnet_View := (
    iParent := fbCabinet,
    eNodeType := E_BACnet_NodeType.eDevice,
    sObjectName := '\C6015',
    sDescription := '\Edge Device');

fbAv02 : FB_BACnet_AV := (
    sObjectName := '\AV02',
    sDescription := '\Analog Value02',
    iParent := fbController );

// view structure Madrid
fbMadrid : FB_BACnet_View := (
    iParent := fbSpain,
    eNodeType := E_BACnet_NodeType.eOrganizational,
    sObjectName := '\Madrid',
    sDescription := '\Madrid offices');

fbAv03 : FB_BACnet_AV := (
    sObjectName := '\AV03',
    sDescription := '\Test äöüÄÖÜß',
    iParent := fbMadrid );

bChangeAv03 : BOOL;
sTest : STRING(255);

```

Code

```

// Germany
fbGermany();
fbVerl();
fbEiserstr();
fbFirstFloor();
fbCabinet();
fbAv01();
fbController();
fbAv02();

// Switzerland
fbSwitzerland();

// Spain
fbSpain();
fbMadrid();

IF bChangeAv03 THEN
    bChangeAv03 := FALSE;
    sTest := 'ßÄÖüäöü This is a test!';
    // requires library Tc2_Uutilities
    STRING_TO_UTF8( ADR( fbAv03.sDescription ), ADR( sTest ), SIZEOF( fbAv03.sDescription) );
END_IF

fbAv03();

```

9.16 Arrayinitialisierung

Sollen die Propertywerte von BACnet-FBs als Array mit identischen Werten initialisiert werden, lässt sich die nachfolgend beschriebene Syntax verwenden. Dabei werden auf alle Elemente der Arrays AI, AO und AV die Eigenschaften Server und iParent zugewiesen. Dieses Beispiel zeigt ebenfalls den zyklischen Aufruf mit Hilfe einer for-Schleife.

Variablen

```

FUNCTION_BLOCK FB_BACnetServer
VAR
    Adapter : FB_BACnet_Adapter;
    Server : FB_BACnet_Server := (Adapter := Adapter);

```

```
View_AI : FB_BACnet_View := (Server := Server, sObjectName := 'AI_Objects', sDescription :=
'Collector AI Objects');

View_AO: FB_BACnet_View := (Server := Server, sObjectName := 'AO_Objects', sDescription :=
'Collector AO Objects');

View_AV : FB_BACnet_View := (Server := Server, sObjectName := 'AV_Objects', sDescription :=
'Collector AV Objects');

AI : ARRAY[1..MAX_OBJECTS] OF FB_BACnet_AI := [MAX_OBJECTS((Server := Server, iParent :=
View_AI))];

AO : ARRAY[1..MAX_OBJECTS] OF FB_BACnet_AO := [MAX_OBJECTS((Server := Server, iParent :=
View_AO))];

AV : ARRAY[1..MAX_OBJECTS] OF FB_BACnet_AV := [MAX_OBJECTS((Server := Server, iParent :=
View_AV))];

iFor : INT;
END_VAR
VAR CONSTANT
MAX_OBJECTS : INT := 50;
END_VAR
```

Code

```
Adapter();
Server();
View_AI();
View_AO();
View_AV();

FOR iFor := 1 TO MAX_OBJECTS DO
AI[iFor]();
AO[iFor]();
AV[iFor]();
END_FOR
```

10 Support und Service

Beckhoff und seine weltweiten Partnerfirmen bieten einen umfassenden Support und Service, der eine schnelle und kompetente Unterstützung bei allen Fragen zu Beckhoff Produkten und Systemlösungen zur Verfügung stellt.

Downloadfinder

Unser [Downloadfinder](#) beinhaltet alle Dateien, die wir Ihnen zum Herunterladen anbieten. Sie finden dort Applikationsberichte, technische Dokumentationen, technische Zeichnungen, Konfigurationsdateien und vieles mehr.

Die Downloads sind in verschiedenen Formaten erhältlich.

Beckhoff Niederlassungen und Vertretungen

Wenden Sie sich bitte an Ihre Beckhoff Niederlassung oder Ihre Vertretung für den [lokalen Support und Service](#) zu Beckhoff Produkten!

Die Adressen der weltweiten Beckhoff Niederlassungen und Vertretungen entnehmen Sie bitte unserer Internetseite: www.beckhoff.com

Dort finden Sie auch weitere Dokumentationen zu Beckhoff Komponenten.

Beckhoff Support

Der Support bietet Ihnen einen umfangreichen technischen Support, der Sie nicht nur bei dem Einsatz einzelner Beckhoff Produkte, sondern auch bei weiteren umfassenden Dienstleistungen unterstützt:

- Support
- Planung, Programmierung und Inbetriebnahme komplexer Automatisierungssysteme
- umfangreiches Schulungsprogramm für Beckhoff Systemkomponenten

Hotline: +49 5246 963-157

E-Mail: support@beckhoff.com

Beckhoff Service

Das Beckhoff Service-Center unterstützt Sie rund um den After-Sales-Service:

- Vor-Ort-Service
- Reparaturservice
- Ersatzteilservice
- Hotline-Service

Hotline: +49 5246 963-460

E-Mail: service@beckhoff.com

Beckhoff Unternehmenszentrale

Beckhoff Automation GmbH & Co. KG

Hülshorstweg 20
33415 Verl
Deutschland

Telefon: +49 5246 963-0

E-Mail: info@beckhoff.com

Internet: www.beckhoff.com

Mehr Informationen:
www.beckhoff.com/tf8020

Beckhoff Automation GmbH & Co. KG
Hülshorstweg 20
33415 Verl
Deutschland
Telefon: +49 5246 9630
info@beckhoff.com
www.beckhoff.com

