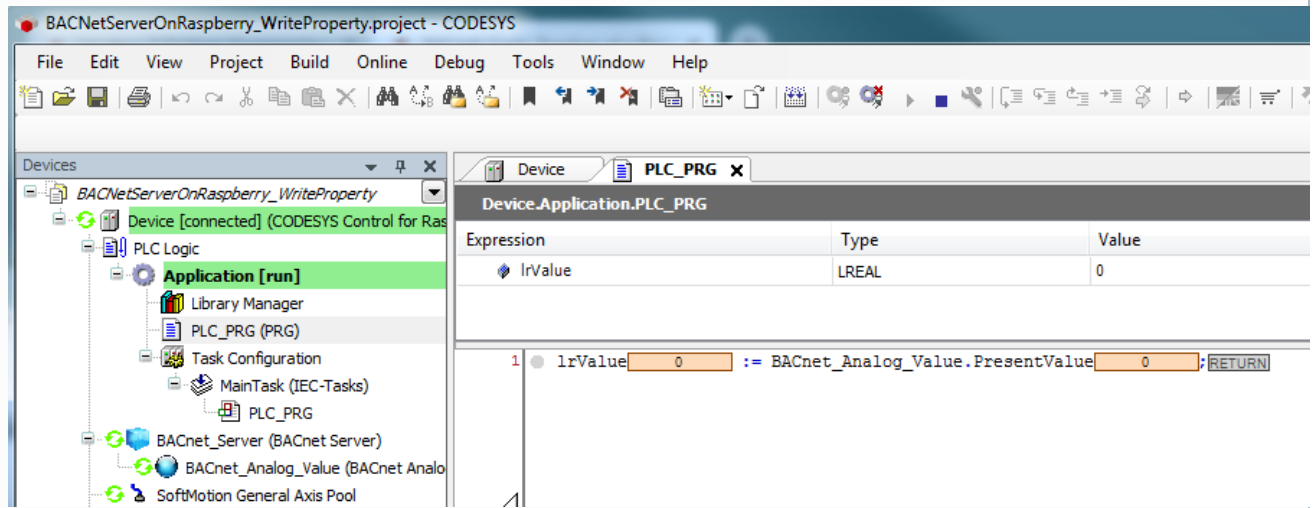


BACnet: Writing of a Property (PresentValue)



The BACnet server from the FAQ [BACnet: Cyclic Reading of a Property](#) is used as the server here.

Reduce this one to reading the analog value.



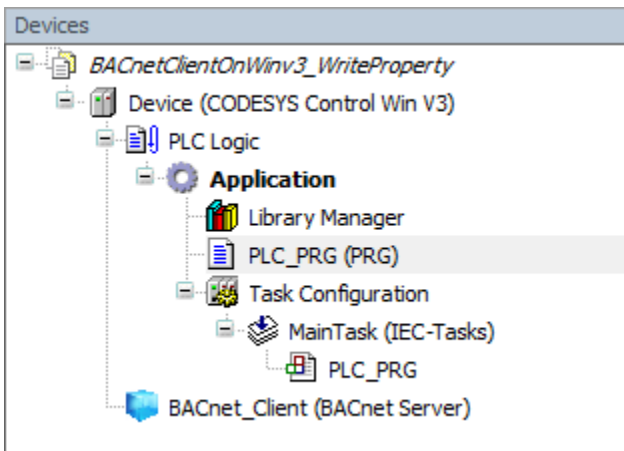
Requirements for the server and client

Check that the following entries are present in the file [CODESYSControl.cfg](#).

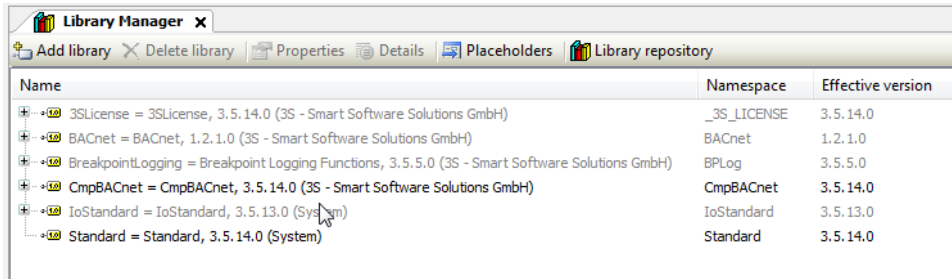
- Integration of the BACnet component:
[ComponentManager]
Component.[n+1]=CmpBACnet
- Integration of the INI file of the BACnet stack (pay attention to the syntax):
[CmpBACnet]
IniFile=bacstacd.ini

BACnet client project

- Create a "Standard project" and select *CODESYS Control Win V3* as the device.
- Define the target system by means of the network scan (see BACnet server).
- Insert a "BACnet Server" object into the device tree and rename it as "BACnet_Client".



- Open the [Library Manager](#) and add the following libraries:
[CmpBACnet](#)



- Edit the [PLC_PRG](#) POU as follows:

Declaration

```
VAR
    fbWriteProperty : BACnet.BACnetClientWriteProperty;
    writePropVal    : BACnet.CmpBACnet.IEC_BACNET_REAL := 16.34;
    writePropCont    : BACnet.CmpBACnet.IEC_BACNET_PROPERTY_CONTENTS;
    xWriteExecute    : BOOL;
    writePrio        : CmpBACnet.IEC_BACNET_SIGNED := 16;
    xInitDone        : BOOL := FALSE;
END_VAR
```

Implementation

```
IF NOT xInitDone THEN
    fbWriteProperty.RegisterToServer(BACnet_Client);
    fbWriteProperty(dwTargetDeviceNumber := 718,
        objType := BACnet.CmpBACnet.IEC_BACNET_OBJECT_TYPE.OBJ_ANALOG_VALUE, objInst :
    = 1,
        propID := BACnet.CmpBACnet.IEC_BACNET_PROPERTY_ID.PROP_PRESENT_VALUE);
    xInitDone := TRUE;
ELSE
    writePropCont.buffer.pBuffer := ADR(writePropVal);
    writePropCont.buffer.nBufferSize := SIZEOF(writePropVal);
    writePropCont.nElements := 1;
    writePropCont.tag := BACnet.CmpBACnet.IEC_BACNET_DATA_TYPE.DATA_TYPE_REAL;

    fbWriteProperty(xExecute := xWriteExecute,
        propertyContents := writePropCont,
        nWritePriority := writePrio);
END_IF
```

- Download the project to the controller and set the [xWriteExecute](#) variable to [TRUE](#).

The value is transferred to the server.

The first screenshot shows the 'Device:Application.PLC_PRG' window. The 'Expression' table lists the following data:

Expression	Type	Value	Prepared
fbWriteProperty	BACnet.BACnetClien...		
writePropVal	REAL	16.34	
writePropCont	BACnet.CmpBACnet...		
xWriteExecute	BOOL	TRUE	
writePrio	DINT	16	
xInitDone	BOOL	TRUE	

The ladder logic shows the following code:

```
1 IF NOT xInitDone THEN
2   fbWriteProperty.RegisterToServer(BACnet_Client);
3   fbWriteProperty(dwTargetDeviceNumber := 718,
4     objType := BACnet.CmpBACnet.IEC_BACNET_OBJECT_TYPE_OBJ_ANALOG_VALUE, objInst := 1,
5     propID := BACnet.CmpBACnet.IEC_BACNET_PROPERTY_ID_PROP_PRESENT_VALUE);
6   xInitDone := TRUE;
7 ELSE
8   writePropCont.buffer.pBuffer := ADDR(writePropVal);
9   writePropCont.buffer.nBufferSize := SIZEOF(writePropVal);
10  writePropCont.nElements := 1;
11  writePropCont.tag := BACnet.CmpBACnet.IEC_BACNET_DATA_TYPE_DATA_TYPE_REAL;
12
13  fbWriteProperty(xExecute := xWriteExecute,
14    propertyContents := writePropCont,
15    nWritePriority := writePrio);
16 END_IF
```

The second screenshot shows the 'Device:Application.PLC_PRG' window for the server. The 'Expression' table lists the following data:

Expression	Type	Value
lrValue	LREAL	16.340000152587891

The ladder logic shows the following code:

```
1 lrValue := BACnet_Analog_Value.PresentValue; RETURN
```



The *PresentValue* property is written with priorities from 1 to 16, where 16 is the lowest.

When you write a value with a higher priority to the server, it is retained until it is deleted or overwritten by an even higher priority.

During the delete operation, the value of the highest set priority is then output as *PresentValue*.

A priority is deleted by writing to it with the data type *DATA_TYPE_NULL*.

The screenshot shows the 'Device:Application.PLC_PRG' window. The 'Expression' table lists the following data:

Expression	Type	Value
fbWriteProperty	BACnet.BACnetClientWriteProperty;	
writePropVal	BACnet.CmpBACnet.IEC_BACNET_REAL := 16.34;	
writePropCont	BACnet.CmpBACnet.IEC_BACNET_PROPERTY_CONTENTS;	
xWriteExecute	BOOL;	
writePrio	CmpBACnet.IEC_BACNET_SIGNED := 16;	
xInitDone	BOOL := FALSE;	
xDelete	BOOL;	

The ladder logic shows the following code:

```
1 PROGRAM PLC_PRG
2 VAR
3   fbWriteProperty : BACnet.BACnetClientWriteProperty;
4   writePropVal : BACnet.CmpBACnet.IEC_BACNET_REAL := 16.34;
5   writePropCont : BACnet.CmpBACnet.IEC_BACNET_PROPERTY_CONTENTS;
6   xWriteExecute : BOOL;
7   writePrio : CmpBACnet.IEC_BACNET_SIGNED := 16;
8   xInitDone : BOOL := FALSE;
9   xDelete : BOOL;
10 END_VAR
11
12 IF NOT xInitDone THEN
13   fbWriteProperty.RegisterToServer(BACnet_Client);
14   fbWriteProperty(dwTargetDeviceNumber := 718,
15     objType := BACnet.CmpBACnet.IEC_BACNET_OBJECT_TYPE_OBJ_ANALOG_VALUE, objInst := 1,
16     propID := BACnet.CmpBACnet.IEC_BACNET_PROPERTY_ID_PROP_PRESENT_VALUE);
17   xInitDone := TRUE;
18 ELSE
19   writePropCont.buffer.pBuffer := ADDR(writePropVal);
20   writePropCont.buffer.nBufferSize := SIZEOF(writePropVal);
21   writePropCont.nElements := 1;
22   writePropCont.tag := BACnet.CmpBACnet.IEC_BACNET_DATA_TYPE_DATA_TYPE_REAL;
23
24   fbWriteProperty(xExecute := xWriteExecute,
25     propertyContents := writePropCont,
26     nWritePriority := writePrio);
27
28   IF xDelete THEN
29     xDelete := FALSE;
30     writePropVal := 0.0;
31     writePropCont.buffer.pBuffer := ADDR(writePropVal);
32     writePropCont.buffer.nBufferSize := SIZEOF(writePropVal);
33     writePropCont.nElements := 1;
34     writePropCont.tag := BACnet.CmpBACnet.IEC_BACNET_DATA_TYPE_DATA_TYPE_NULL;
35
36     fbWriteProperty(xExecute := xWriteExecute,
37       propertyContents := writePropCont,
38       nWritePriority := writePrio);
39   END_IF
40 END_IF
```