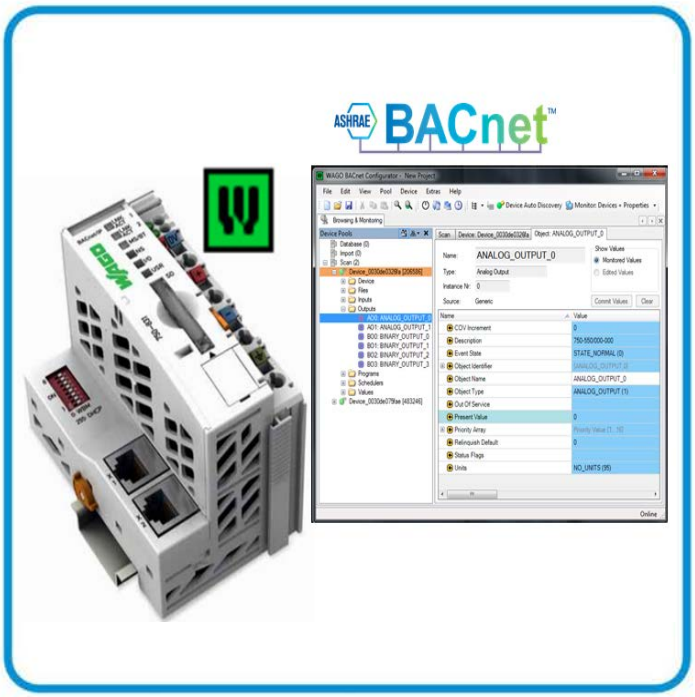
Application Note

AUTOMATION



WAGO-I/O-SYSTEM 750
BACnet/IP Controller 750-831
WAGO BACnet Library

Version: 08.10.2013

WAGO® INNOVATIVE CONNECTIONS

WAGO Kontakttechnik GmbH & Co. KG

Hansastraße 27
D-32423 Minden

Phone: +49 (0) 571/8 87 – 0
Fax: +49 (0) 571/8 87 – 1 69

E-mail: info@wago.com

Web: http://www.wago.com

Technical Support

Phone: +49 (0) 571/8 87 – 5 55
Fax: +49 (0) 571/8 87 – 85 55

E-mail: support@wago.com

Every conceivable measure has been taken to ensure the accuracy and completeness of this documentation. However, as errors can never be fully excluded, we always appreciate any information or suggestions for improving the documentation.

We wish to point out that the software and hardware terms as well as the trademarks of companies used and/or mentioned in the present manual are generally protected by trademark or patent.

Table of Contents

1 Important Notes

To ensure fast installation and start-up of the units, we strongly recommend that the following information and explanations are carefully read and adhered to.

1.1 Legal Principles

1.1.1 Subject to Change

WAGO Kontakttechnik GmbH & Co. KG reserves the right to make any alterations or modifications that serve to increase the efficiency of technical progress. WAGO Kontakttechnik GmbH & Co. KG owns all rights arising from the granting of patents or from the legal protection of utility patents. Third-party products are always mentioned without any reference to patent rights. Thus, the existence of such rights cannot be excluded.

1.1.2 Copyright

This documentation, including all figures and illustrations contained therein, is subject to copyright protection. Any use of this documentation that infringes upon the copyright provisions stipulated herein is prohibited. Reproduction, translation, electronic and phototechnical filing/archiving (e.g., photocopying), as well as any amendments require the written consent of WAGO Kontakttechnik GmbH & Co. KG, Minden, Germany. Non-observance will entail the right of claims for damages.

1.1.3 Personnel Qualification

The use of the product described in this document is exclusively geared to specialists having qualifications in SPS programming, electrical specialists or persons instructed by electrical specialists who are also familiar with the appropriate current standards. WAGO Kontakttechnik GmbH & Co. KG assumes no liability resulting from improper action and damage to WAGO products and third-party products due to non-observance of the information contained in this document.

1.1.4 Intended Use

For each individual application, the components are supplied from the factory with a dedicated hardware and software configuration. Modifications are only admitted within the framework of the possibilities documented in this document. All other changes to the hardware and/or software and the non-conforming use of the components entail the exclusion of liability on part of WAGO Kontakttechnik GmbH & Co. KG.

Please send your requests for modified and new hardware or software configurations directly to WAGO Kontakttechnik GmbH & Co. KG.

1.2 Scope of Validity

This application note is based on the stated hardware and software from the specific manufacturer, as well as the associated documentation. This application note is therefore only valid for the described installation.
New hardware and software versions may need to be handled differently.

Please note the detailed description in the specific manuals.

1.3 Symbols

NOTE 	**NOTE** Boundary conditions that must always be observed to ensure smooth operation.
Note 	**Important note** Routines or advice for efficient use of a device and software optimization.
Information 	**Additional information** Refers to additional information which is not an integral part of this documentation (e.g., the Internet).

2 Description

The BACnet/IP controller from the WAGO-I/O-SYSTEM 750 supports the BACnet protocol, which is standardized according to DIN EN ISO 16484-5. These instructions describe the basic way to work with the BACnet/IP controller. The WAGO-BACnet configurator is used to integrate the BACnet/IP controller into BACnet/IP networks.

3 Required Libraries

Table 1: Required libraries

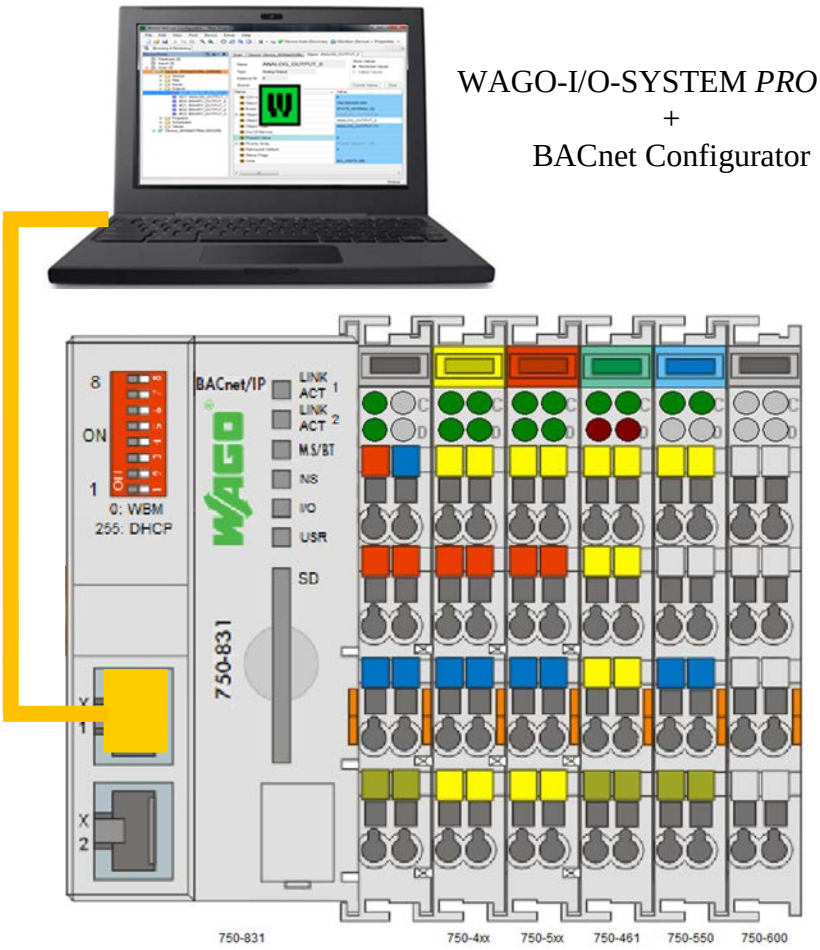
Library	Description
BACnet_02.lib	BACnet library
BACnetAccess.lib	BACnet access library
BACNETOBJECTS.EXP	CODESYS export data for BACnet objects
SYSLIBCALLBACK.lib	System library
Standard.lib	Standard functions

4 Components

Table 2: Components

Supplier	Qty.	Designation	Item No.
WAGO	1	BACnet/IP controller	750-831
WAGO	1	4-channel digital input module	750-402
WAGO	1	4-channel digital output module	750-504
WAGO	1	2-channel, resistance sensors, PT100 / RTD	750-461
WAGO	1	2-channel analog output module	750-550
WAGO	1	End module	750-600
WAGO	1	WAGO BACnet Configurator	-
WAGO	1	WAGO-I/O-PRO V2.3	759-333

5 Setup



Information

Assign the IP address to the fieldbus node

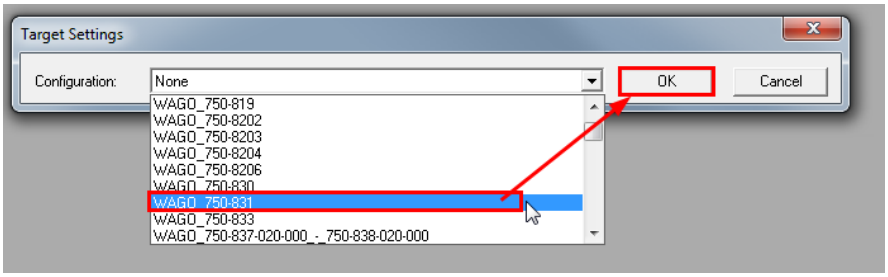
The WAGO ETHERNET Settings 759-316 or WAGO BootP Server software can be used to assign the IP address of the BACnet fieldbus controller. In the WAGO ETHERNET Settings, you can configure an IP address via the USB communication interface (750-923). Learn more about assigning the IP address using the WAGO BootP Server in the manual for the BACnet/IP controller 750-831.

6 Quick-Start Guide

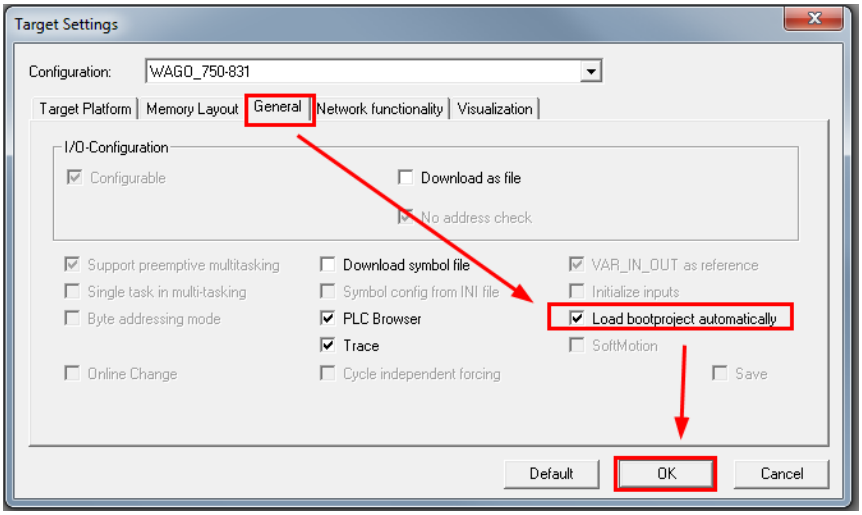
This quick-start guide describes the steps required to create an IEC application with the BACnet/IP fieldbus controller 750-831 in the IEC-61131-3-compliant WAGO-I/O-PRO programming environment (order No. 759-333).

6.1 Create New Project

1. Launch the WAGO-I/O-PRO programming environment to create a new project (**File** > **New**). Select "WAGO 750-831" as the target system.



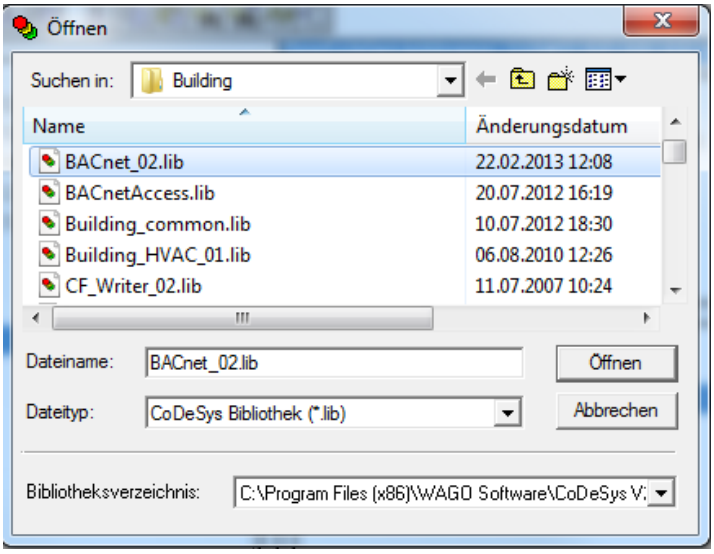
2. In the General menu, enable the load bootproject automatically" setting. Click **[OK]** to apply the setting.

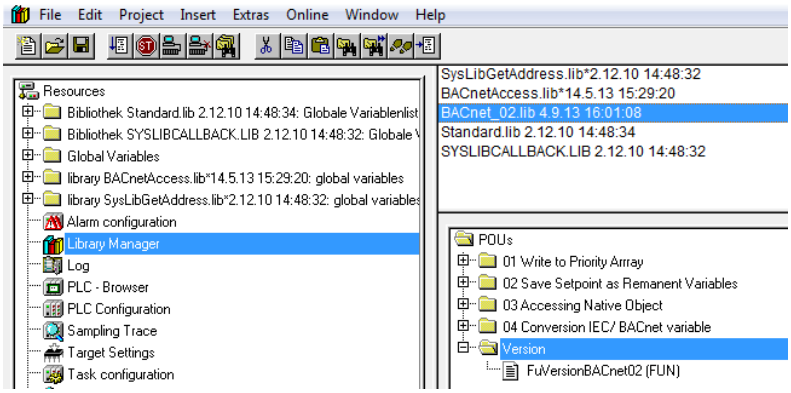


3. The PLC_PRG function module is automatically created. Go to **File** > **Save** to save the project.

6.2 Integrate BACnet library.

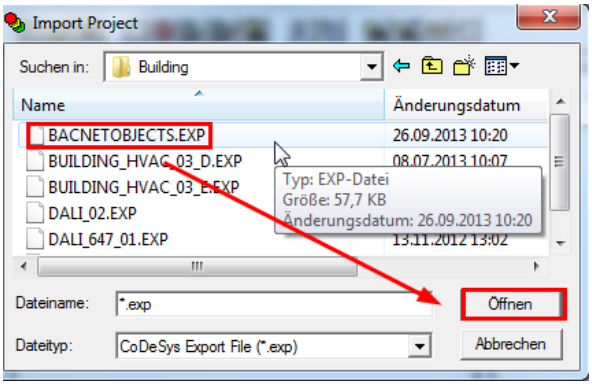
1. The *BACnet_02.lib* library must be integrated in the project via the library manager. The *BACnetAccess.lib* library is automatically included if present in the library directory.



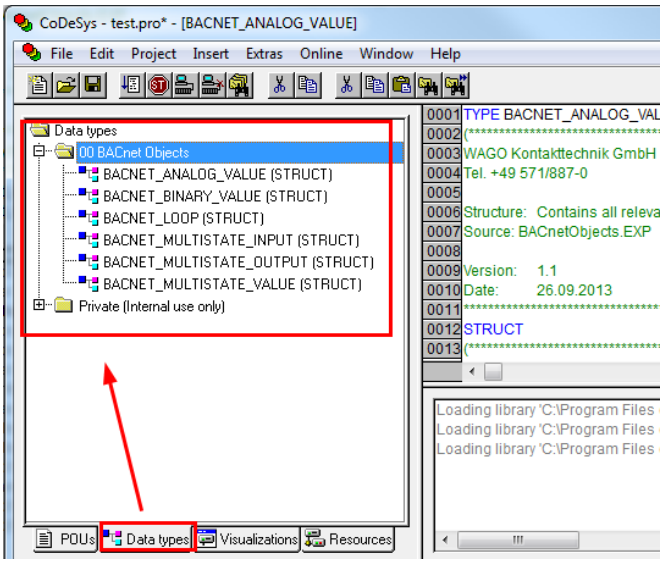


6.3 Import BACnet Objects from the Export File.

1. Import the *BACNETOBJECTS.EXP* import file (**Project** > **Import**). After importing the file, BACnet structures are added to the **Data Types** tab.



2. The export file contains data types that correspond to the structure of the BACnet object types. Other BACnet properties can be added to the BACnet object by commenting out individual structure variables.



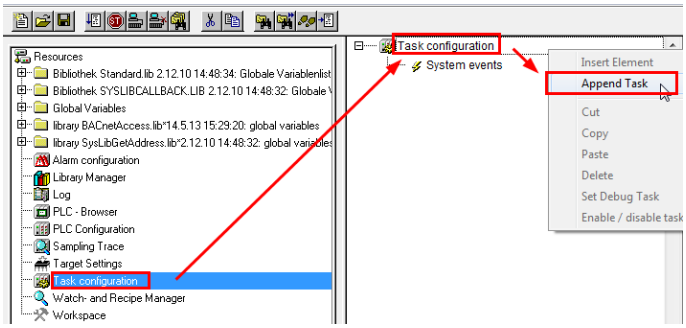
NOTE

Mandatory properties

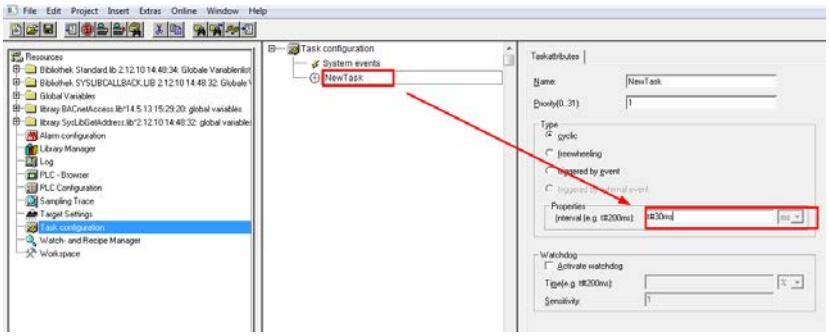
Some properties cannot be commented out or deleted. In the BACnet object structure, these properties are listed under the *Mandatory Properties* line.

6.4 Task Configuration

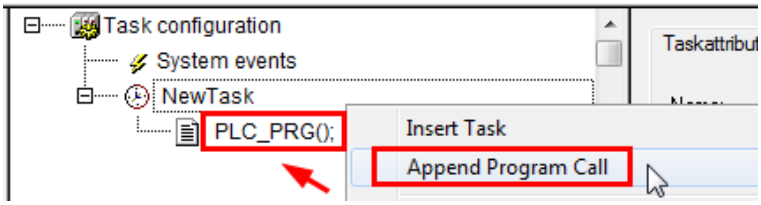
1. Add new tasks in the task configuration.



2. Define task properties (Cyclic, Interval, e.g. t#30ms).



3. Append the PLC_PRG program call.



Note

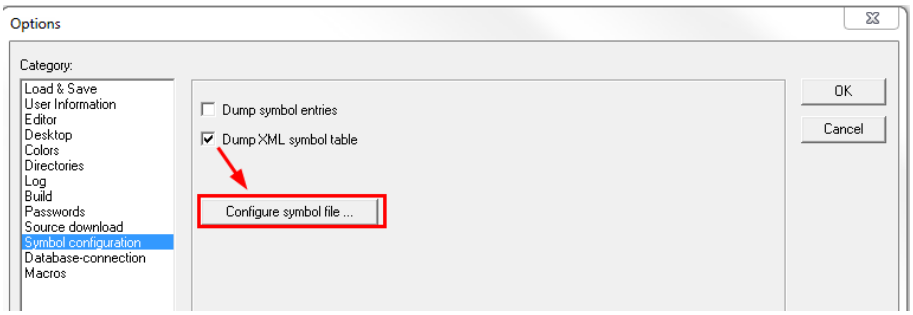
Task configuration
Create a task configuration for all IEC applications in connection with the BACnet/IP controller. Otherwise, the initialization phase of the controller after a reset will last significantly longer.

6.5 Configuration of the *SYM_XML* Symbol File

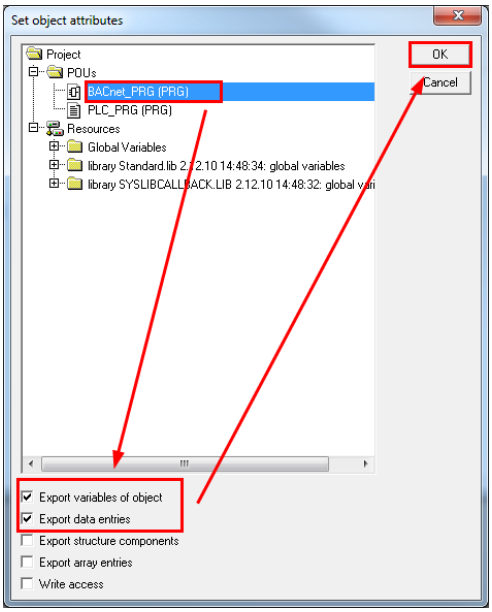
The WAGO BACnet Configurator uses the so-called *SYM_XML* symbol file for importing the IEC variables. All the necessary information for being able to form an assignment between the names of the PLC variables and the associated memory address is located in this file. So that the variables are written to the symbol file when compiling the PLC program, it must be configured in the project of the WAGO-I/O-PRO software.

1. Open the symbol configuration in **Project** > **Options**.

2. Select the checkbox **Dump XML symbol table**. The XML symbol file is created in the project directory and receives the name <ProjectName>.SYM_XML. Click the **[Configure symbol file]** button.

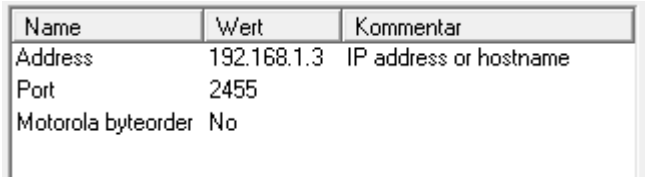


3. In the popup window, select the module in which BACnet variables are used. In this example, the **BACnet_PRG** project module is selected. The "Export variables of objects" and "Export data entries" settings must be selected as object attributes. Click **[OK]** to apply the setting.



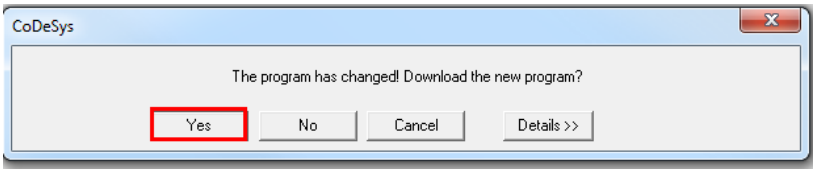
6.6 Downloading the Program to the BACnet/IP Controller 750-831 and Importing the Symbol File

1. First, the communication driver must be set up. In the **Online > Communication parameters** menu, click **[New]** to create a new communication channel. Select **TCP/IP (3S TCP/IP drive)**. Under **Address**, enter the IP address of your BACnet/IP controller.



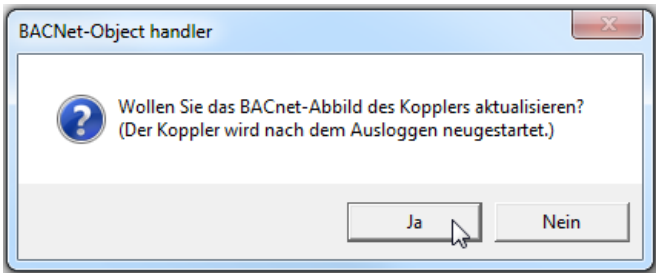
2. Transfer the program via **Online** > **Log in**.

3. Click **[Yes]** to confirm the prompt in the message window.



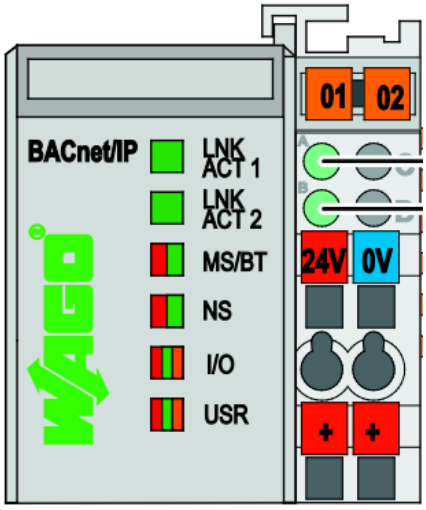
4. You can also created a comparison between SYM_XML and BACnet Configurator. When performing a comparison, confirm the prompt by clicking **[Yes]**. The comparison only takes effect after restarting the controller (log out).

   Alternatively, comparison of the SYM_XML can also be carried you via the WAGO BACnet Configurator. In this case, confirm the prompt by clicking **[No]**.



5. If you click **[Yes]** in 4 above, the *SYM_XML* symbol file is updated.

6. The red I/O LED of the BACnet controller flashes with error code 6/10. The error code indicates that the IEC application does not match the SYM_XML file. To correct this condition, log out of the program via **Online** > **Log out**. The controller is automatically restarted. After restarting, the I/O LED lights up green.



7 BACnet Objects in the IEC Application

7.1 Create BACnet Objects

This example describes how IEC variables can be created as BACnet objects in the WAGO BACnet Configurator.

The *BACNETOBJECTS.EXP* export file contains data types that correspond to BACnet objects (see Table 3). The export file must first be imported into the project (see Chapter 6.3 on page 10).

Table 3: Data types and their corresponding BACnet objects.

Data type	Corresponding BACnet object
BACNET_ANALOG_VALUE	Analog Value Object Type
BACNET_BINARY_VALUE	Binary Value Object Type
BACNET_MULTISTATE_VALUE	Multi-state Value Object Type
BACNET_LOOP	Loop Object Type
BACNET_MULTISTATE_INPUT	Multi-state Input Object Type
BACNET_MULTISTATE_OUTPUT	Multi-state Output Object Type

If an IEC variable of one of the abovementioned data types (see Figure 1) is declared, the BACnet/IP Controller 750-831 generates a corresponding BACnet object.

```
0002 VAR
0003     (*BACnet Objects. Can be generated using BACnet/ IP Controller 750-830 or 750-831*)
0004     IEC_Analog_Value        : BACNET_ANALOG_VALUE;      (*BACnet AV Object*)
0005     IEC_Binary_Value        : BACNET_BINARY_VALUE;      (*BACnet BV Object*)
0006     IEC_Multistate_Value    : BACNET_MULTISTATE_VALUE; (*BACnet MV Object*)
0007
0008     (*BACnet Objects that can ONLY be generated using BACnet/ IP Controller 750-831*)
0009     IEC_Multistate_Input    : BACNET_MULTISTATE_INPUT; (*BACnet MI Object*)
0010     IEC_Multistate_Output   : BACNET_MULTISTATE_OUTPUT;   (*BACnet MO Object*)
0011     IEC_LoopObject          : BACNET_LOOP;      (*BACnet Loop Object*)
0012 END_VAR
```

Figure 1: Declaration of the BACnet objects in an IEC application

Note

Selecting the data type

With the help of the **[F2]** function key, you can open the input help to make selecting the data types from the library easier.

After importing the symbol file (see Chapter 6.6 on page 13), the BACnet objects are automatically created in the WAGO BACnet Configurator (see Figure 2).

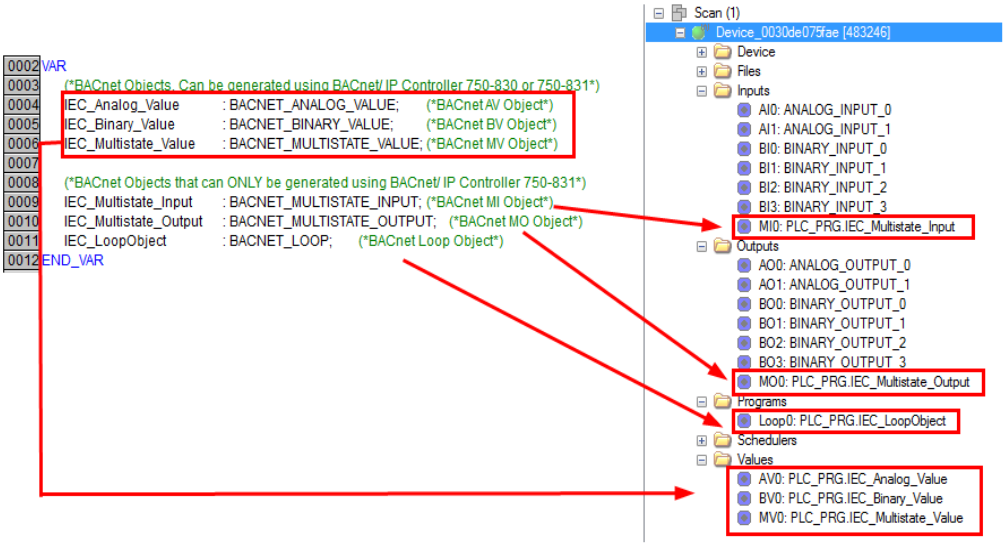


Figure 2: Generated IEC variables in the BACnet Configurator

NOTE

Compatibility of BACnet Object using BACnet/ IP Controller 750-830

The following BACnet objects are not supported by BACnet/IP Controller 750-830:

-BACnet Loop Object,
-Multi-State Input,
-Multi-State Output.

7.2 Priority-controlled Writing

For some applications, it is necessary that output variables can be controlled from the IEC application and via the BACnet network in parallel.

So that it is determined clearly which of the two access points writes the output value, the priority-controlled writing of the *Present_Value* property is provided in the BACnet objects. Function blocks from the BACnet_02.lib library support the priority-controlled writing (see Figure 3).

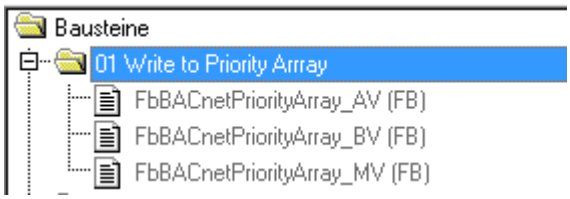


Figure 3: Function module for priority-controlled writing

Figure 4 shows how the priority array of an **IEC_Analog_Value_Object_Type** BACnet object created in the IEC application can be written. The **bPriorityAV** variable is used to determine the write priority of the IEC application. A value of 1 denotes highest priority, and a value of 16 denotes lowest priority.

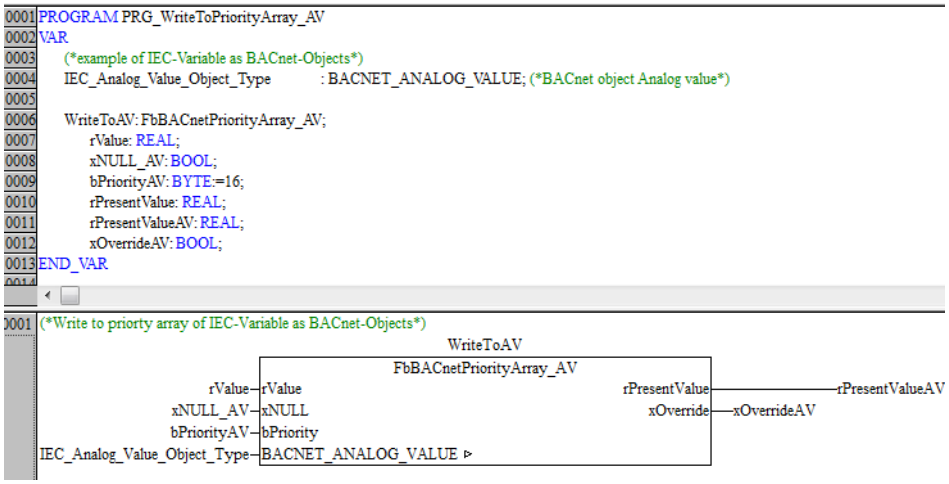


Figure 4: Priority-controlled writing

The **rValue** is only written to the priority array of the **IEC_Analog_Value_Object_Type** BACnet object if the **xNULL_AV** input is "FALSE". If the input is "TRUE", the "NULL" value is written to the array element (specified via **bPriority_AV**) of the **IEC_Analog_Value_Object_Type** variables. The "NULL" value can be used to reset write access with a specific priority.

If the BACnet network writes a value with a higher priority, thus overriding the value from the IEC application, then the **xOverrideAV** output is set to "TRUE".

7.3 Save BACnet Objects as Remanent Variables

For some applications, it is necessary that the values written via BACnet also remain unchanged after a reset. For this purpose, two values from the priority array of a BACnet object created in the IEC application can be retained:

- **Setpoint** is always the priority array value with priority 16. This value can be entered in the IEC application or via the WAGO BACnet Configurator.
- **Override setpoint** is the priority array value with higher priority than the setpoint, i.e. lower than 16. The override setpoint can be written e.g. via the WAGO BACnet Configurator or a Building management system.

The function blocks of the *BACnet_02.lib* library are used to save the abovementioned setpoint (see Figure 5).

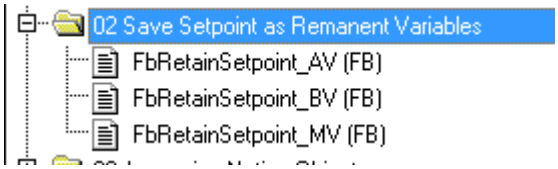


Figure 5: Function block for saving the setpoints as remanent variables

Figure 6 shows the program for retaining the value of a BACnet analog object. In this example, the values from priority index 16 and 8 are retained for the **IEC_Analog_Value_Object_Type** object. The **bPrioritySelection_AV** variable specifies the priority of the override setpoint to be saved ("8" in this example).

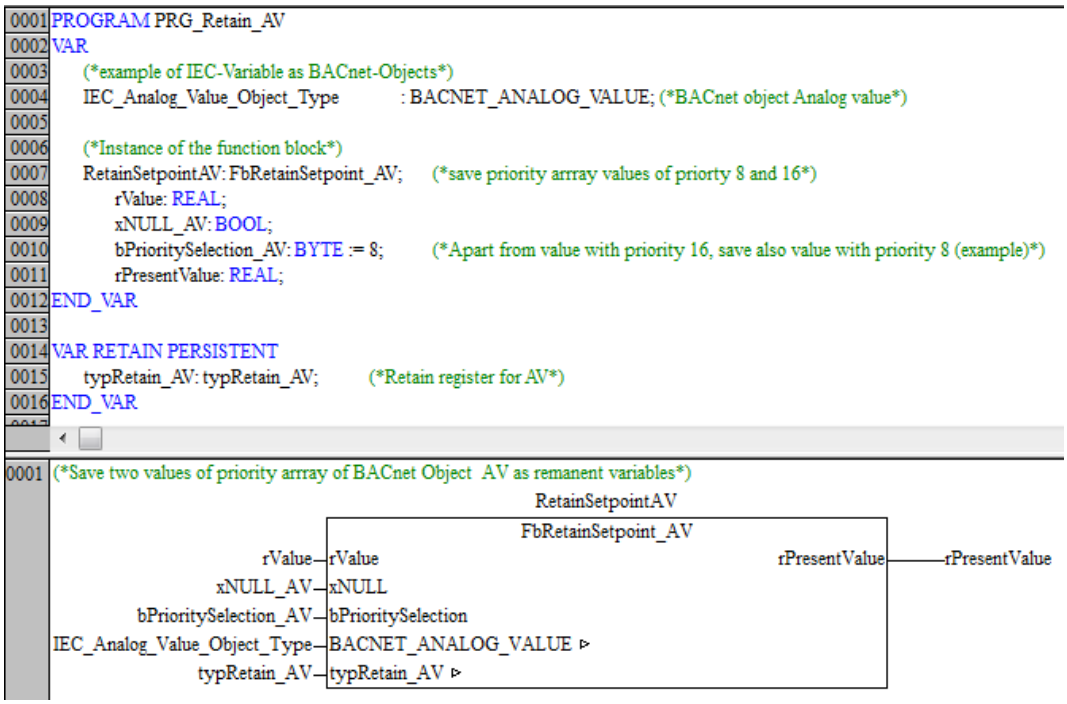


Figure 6: Program for saving the setpoints of a BACnet analog object as remanent variables and its variables

Note

RETAIN PERSISTENT declaration

To retain the values of an analog value object after reset or application download, the **typRetain_AV** structure variable must be declared RETAIN and PERSISTENT.

Note

Saving the setpoints of a BACnet multi-state object

Restoration of the setpoints depends on the initial value of the "Number of State" property. The "Number of State" property defines how many states the "Present Value" property can accept. The "Number of State" property is initialized by configuring the respective BACnet multi-state object via the BACnet Configurator. If the "Number of State" property is not configured, the property is initialized with "1" after reset or after application download.

8 Native BACnet Objects

For all digital and analog I/O modules recognized on the internal data bus, the BACnet/IP controller automatically generates corresponding BACnet object instances (ANALOG_INPUT, ANALOG_OUTPUT, BINARY_INPUT or BINARY_OUTPUT). The generated object instances are available to the BACnet network without additional configuration or programming.

In this application example, the following modules are connected:

- 1x 2-channel analog input module 750-461
- 1x 2-channel analog output module 750-550
- 1x 4-channel digital input module 750-402
- 1x 4-channel digital output module 750-504

The result is configuration shown in Figure 7 with 12 native BACnet objects.

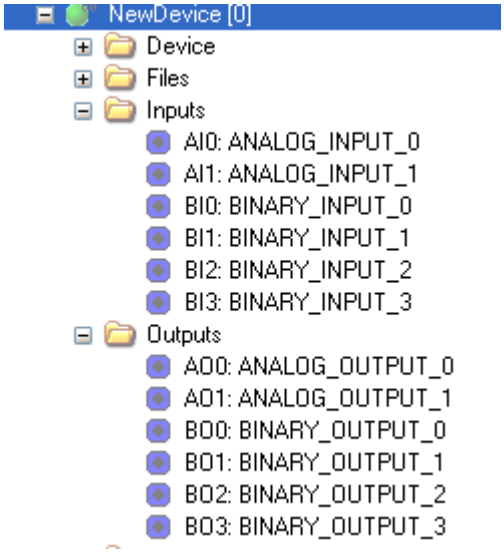


Figure 7: Automatically generated BACnet object instances

Note

BACnet-native
A definition of "BACnet-native" as a system property is (still) not specified by standard.

In the WAGO BACnet controller, the term "BACnet-native" refers to the automatically generated object instances available to the BACnet network without additional configuration or IEC programming.

8.1 BACnet-native Input Object (AI and BI)

8.1.1 "Out of Service" Operating Mode

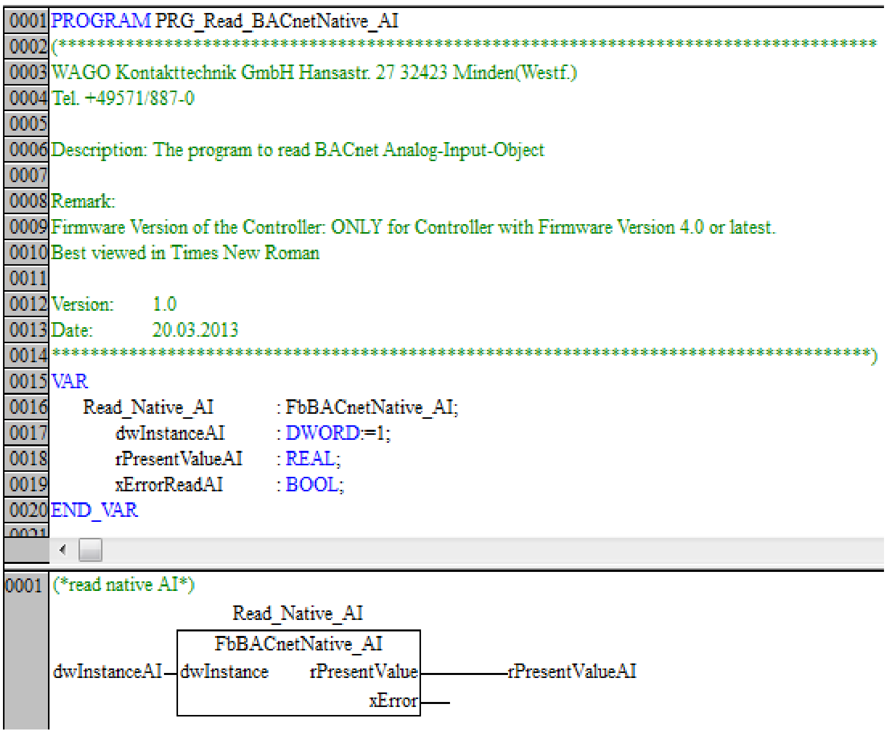
An "OutOfService" operating state is defined for BACnet-native input objects (AI or BI).

- Out of Service means that the *Present_Value* value of the property is decoupled from the physical input. The *Out_of_Service* property of the input object is set to "TRUE" in this operating state.

  In this operating state, the value of the *Present_Value* property can be manually set to any value for purposes of simulation or testing.

8.1.2 Reading a BACnet-native Input Object

So that the "Out Of Service" operating state described in Chapter 7.1.1 can also be evaluated by the IEC application, the **FbBACnetNative_AI** function block must be called up from the BACnet_02.lib library. Figure 8 shows the program for reading a BACnet-native analog input object (AI). The instance number of the object is specified at the **dwInstance** input. The value read by the function block is made available at that **rPresentValue** output.



```
PROGRAM PRG_Read_BACnetNative_AI
(*************************************************************************************
WAGO Kontakttechnik GmbH Hansastr. 27 32423 Minden(Westf.)
Tel. +49571/887-0

Description: The program to read BACnet Analog-Input-Object

Remark:
Firmware Version of the Controller: ONLY for Controller with Firmware Version 4.0 or latest.
Best viewed in Times New Roman

Version:     1.0
Date:        20.03.2013
*************************************************************************************)
VAR
    Read_Native_AI        : FbBACnetNative_AI;
        dwInstanceAI      : DWORD:=1;
        rPresentValueAI   : REAL;
        xErrorReadAI      : BOOL;
END_VAR
```


Figure 8: Program for reading a BACnet-native analog input object (AI).

The instance number (*Instance No.*) of the object concerned can be determined via the WAGO BACnet Configurator (see Figure 9).

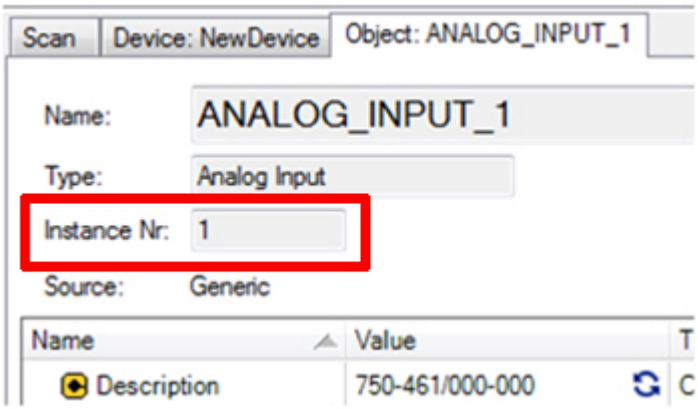


Figure 9: Instance number of a BACnet-native analog input object♣

8.2 BACnet-native Output Object (AO and BO)

8.2.1 Writing a BACnet-native Output Object

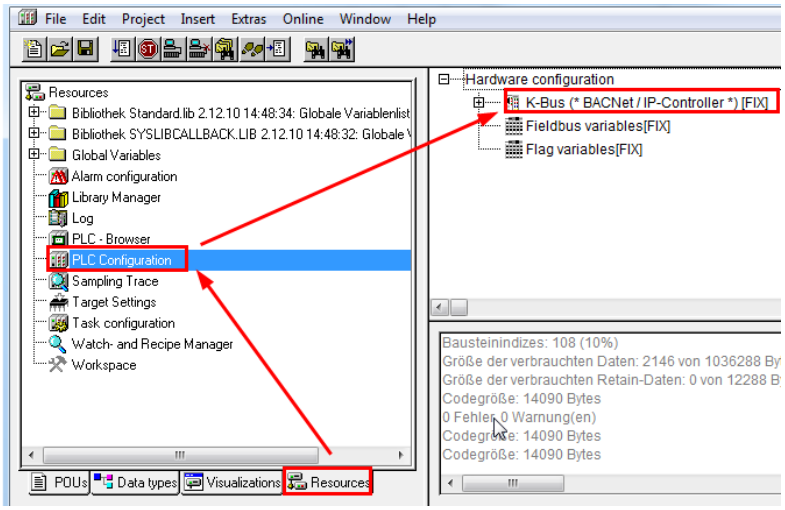
Write access to BACnet-native output objects (AO and BO) depends on the settings of the controller configuration in the project of the WAGO-I/O-PRO software. There are two types of write access:

- "PLC" write access means that the BACnet-native output objects have no write access to the output modules of the WAGO-I/O-SYSTEM. Writing the outputs can only be controlled via the IEC application.
- "BACnet" write access means that the BACnet-native output objects can control the output modules of the WAGO-I/O-SYSTEM directly.

8.2.2 Change Write Access of the Output Module

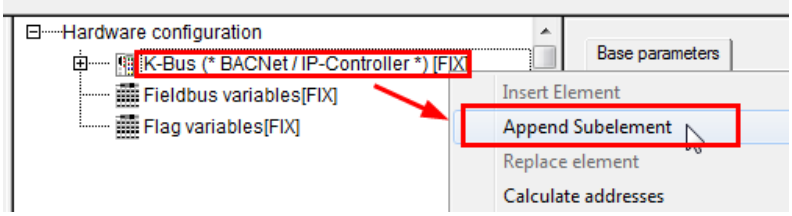
The write access of the output module can be individually set via the controller configuration of the WAGO-I/O-PRO.

1. In the **Resources** tab, you can change the write access of an output module via the *PLC configuration*. The output modules connected to the BACnet/IP controller are created under *Hardware configuration* > *K-Bus (*BACnet*)*.



2. Right click on *K-Bus*[1] (*BACnet*)* and select **Append Subelement** in the context menu.

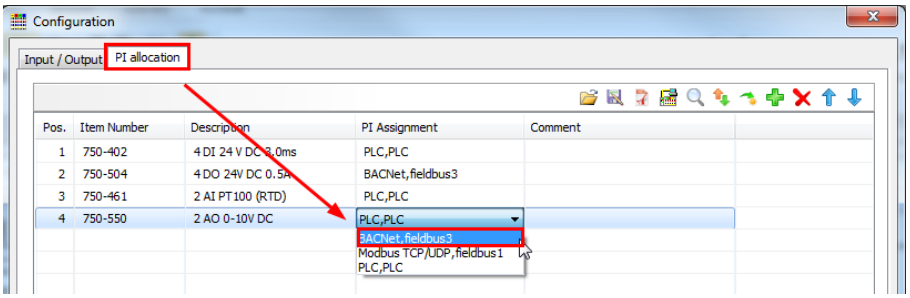
[1] K-Bus = Internal data bus



3. You can change the write access of the output module in the **PI allocation** tab of the *Configuration* manager.

   The "BACnet" parameter determines that the output module can be controlled directly and without use of the IEC application from a BACnet object (BACnet-native).

   If the "PLC" parameter is set, then the outputs of the output module can be addressed exclusively with the help of the IEC application.



8.2.3 Writing a BACnet-native Output Object via the IEC Application

Figure 10 shows the program for writing a BACnet-native analog output object (AO). The **FbBACnetNative_AO** function block is used for this purpose. The instance number of the object is specified at the **dwInstance** input.

```
0001 PROGRAM PRG_Write_BACnetNative_AO
0002 (*****************************************************************************
0003 WAGO Kontakttechnik GmbH Hansastr. 27 32423 Minden(Westf.)
0004 Tel. +49571/887-0
0005
0006 Description: The program writes to BACnet Analog-Output-Object
0007
0008 Firmware Version of the Controller: ONLY for Controller with Firmware Version 4.0 or latest.
0009
0010 Remark: The write access of the corresponding Analog Input module must be set to "BACnet". This can be done
0011       in the PLC-Configuration. Best viewed in Times New Roman
0012
0013 Version:    1.0
0014 Date:       20.03.2013
0015 *****************************************************************************)
0016 VAR
0017     Write_Native_AO          : FbBACnetNative_AO;
0018       rValue_IEC             : REAL;
0019       rSendOnDelta           : REAL;
0020       xNULL_Write_AO         : BOOL;
0021       bPriorityWrite_AO      : BYTE:=16;
0022       dwInstanceAOWrite      : DWORD:=1;
0023       xUpdate                : BOOL;
0024       rPresentValueAO        : REAL;
0025       xOverride              : BOOL;
0026       xErrorWriteAO          : BOOL;
0027 END_VAR
```

```
0001 (*write native AO*)
```

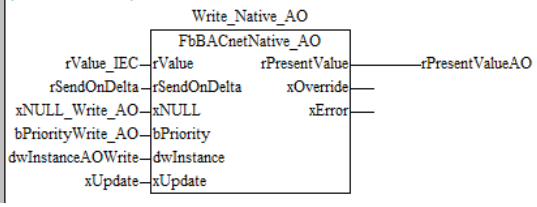


Figure 10: Program for writing a BACnet-native analog output object

The instance number (*Instance No.*) of the object concerned can be determined via the WAGO BACnet Configurator (see Figure 11).

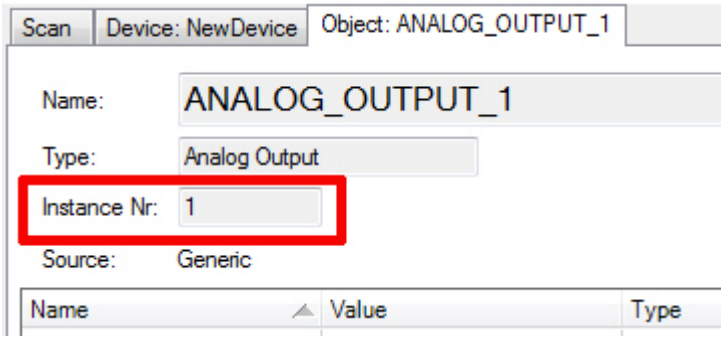


Figure 11: Instance number of a BACnet-native analog output object

The value to be written via the IEC application is specified at the **rValue** input. The write priority is configured at the **bPriority** input.
If the ANALOG_OUTPUT object is written to via BACnet with a higher priority, then the **xOverride** output is set to "TRUE". The current value of the BACnet object always appears at the **rPresentValue** output.

WAGO Kontakttechnik GmbH & Co. KG
Postfach 2880 D-32385 Minden
Hansastraße 27
D-32423 Minden
Phone: +49 (0)5 71/8 87 – 0
Fax: +49 (0)5 71/8 87 – 1 69
E-mail: info@wago.com
Internet: http://www.wago.com