

Unitec BACnet providers BACnet/IP communication Version 1.1.0

User's Guide

March 22, 2023

Note: In order to use this provider, a DLL of Windows compatible BACnet drivers from Unitec is required.

[Revision History]

Version	Date	Content
1.0.0	2019-06-12	First edition

	2021-12-08	Add operation check model Changed driver name description
1.0.1	2023-01-17	Bug fixes when multiple units are connected Correcting Error Codes in the User's Guide
1.1.0	2023-03-22	Corrected not to synchronize the time at initialization. Corrected to not synchronize PC time when the time synchronization is notified. Addition of SendTimeSync,SendTimeSyncUnicast,SendUTCTimeSync command

[Operation check model]

Model	Version	Notes
ICONTSimulator2004	1.0.1.0	Confirmed operation with BACnet development support tool made by Unitec.

Contents

1. Introduction	5
2. Environment setup for application development.....	8
2.1. Connecting a B-BC to a Client-Based PC	8
2.1.1. B-BC monitoring.....	8
2.1.1.1. Healthy Check	8
2.2. Setting Up a PC Developer	8
2.2.1. Preparing the Communication Library	8
2.2.2. Installing BACnet Providers Manually	8
3. Provider Overview	9
3.1. List of Methods/Properties	9
3.2. Method Properties.....	9
3.2.1. CaoWorkspace class-.....	9
3.2.1.1. AddController method.....	9
3.2.2. CaoController class-.....	10
3.2.2.1. FileNames Properties	10
3.2.2.2. AddFile method	11
3.2.2.3. VariableNames Properties	12
3.2.2.4. AddVariable method.....	12
3.2.2.5. OnMessage Events	12
3.2.2.6. Execute method.....	12
3.2.3. CaoFile class-.....	13
3.2.3.1. FileNames Properties	13
3.2.3.2. AddFile method.....	13
3.2.3.3. VariableNames Properties	15
3.2.3.4. AddVariable method.....	16
3.2.4. CaoVariable class-.....	16
3.2.4.1. Value Properties.....	16
3.3. Variable List.....	16
3.3.1. CaoController class-variable	17
3.3.1.1. @VERSION.....	17
3.3.2. CaoFile class-variable	17
3.3.2.1. @PRESENTVALUE.....	18
3.3.2.2. @STATUSFLAGS	19
3.3.2.3. PRIORITYVALUE<??>	20

3.4. Event List.....	21
3.4.1. Details of the data of the message that occurred.....	22
3.4.1.1. BACnet Controllers Status Change	22
3.4.1.2. COV notification information	23
3.4.1.3. Event notification information.....	23
4. Command Reference	29
4.1. CaoController class-.....	29
4.1.1. SendTimeSync Commands	29
4.1.2. SendTimeSyncUnicast Commands	29
4.1.3. SendUTCTimeSync Commands.....	30
5. Programming by BACnet providers.....	31
5.1. Sample programming to get/set the value stored in object priority	31
5.1.1. Sample program	32
5.1.1.1. Entry.....	35
5.1.1.2. Adding CaoFile Objects (3-tier)	36
5.1.1.3. Get/Set Value Stored in Priority	37
5.1.1.4. Deleting CaoFile Objects (3-tier)	37
5.1.1.5. Detachment	38
5.2. Sample Programming to Automatically Receive BACnet Controllers Status Changes	38
5.2.1. Sample program	39
5.2.1.1. Entry.....	41
5.2.1.2. Message reception	42
5.2.1.3. Disconnect.....	42
6. BACnet provider error code	44

1. Introduction

This document is a user's guide for providers accessing BACnet controllers connected to BACnet using UNITEC's BACnet communication API. Figure 1-1 shows the overall configuration of this provider and the device. Hereafter, this provider is referred to as BACnet provider. BACnet also refers to the central monitoring system, "BACnet Operator Workstation," which is called "B-OWS," and BACnet controllers are called "B-BC." Figure 1-1 Configuration Diagram 11

BACnet providers connect to BACnet controllers using a Unitec BACnet communications API based on a communications standard for building networks, known as BACnet. In addition, when creating providers this time, we are preparing them by referring to the operating procedures described in BACnet disclosure specifications provided on the Azbil Corporation website. In the subsequent sections of this manual, BACnet public specification document provided on the Azbil Corporation website is called BACnet public specification document. We recommend that you learn about BACnet before you use it.¹

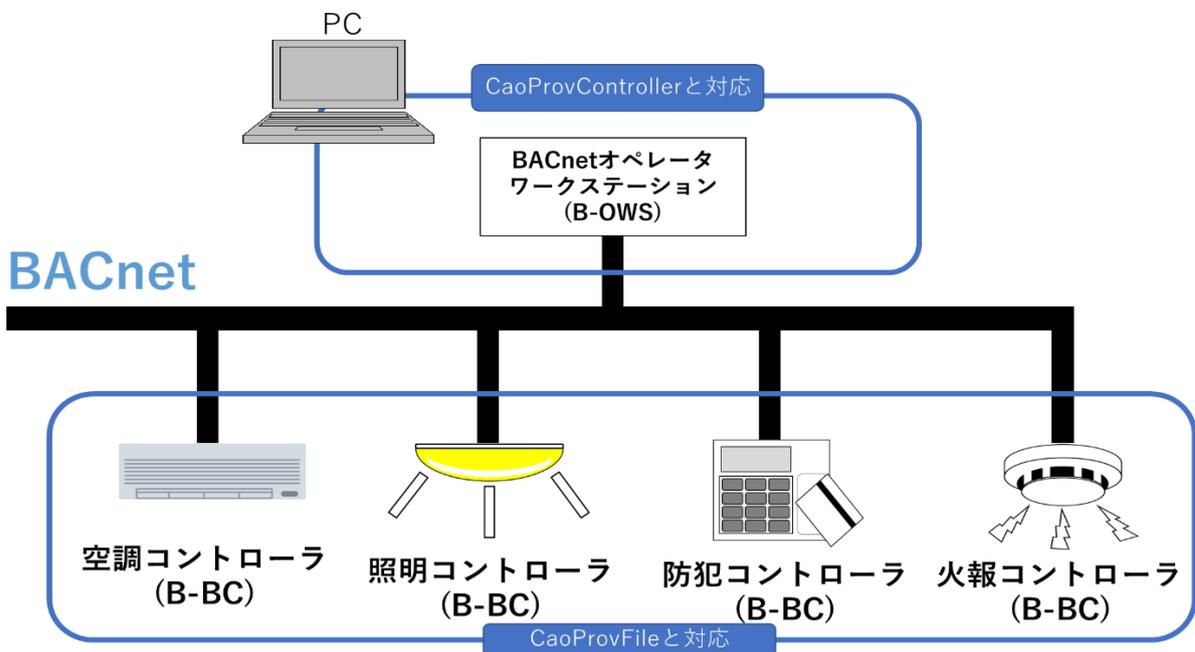


Figure 1-1 Configuration Diagram 11

Figure 1-2 shows an example of the correspondence between this provider and devices. Figure 1-2: Correspondence between provider configuration and device information¹²

¹<https://www.azbil.com/jp/product/building/system/BACnet.html>

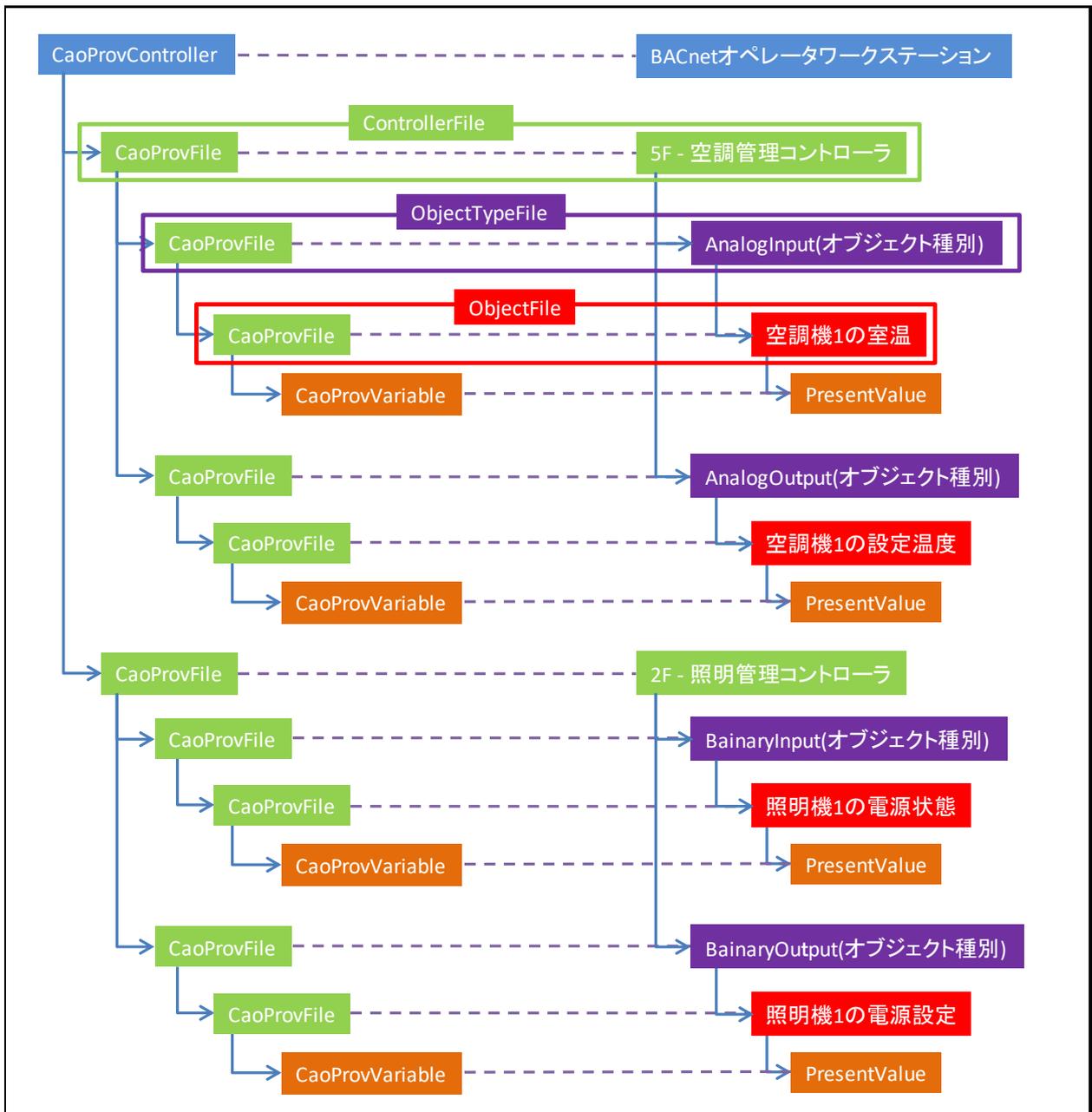


Figure 1-2: Correspondence between provider configuration and device information12

CaoProvController:

Corresponds to BACnet operator workstations.

※ Due to the limitations of the Unitec BACnet communication API, you cannot add more than one BACnet operator workstation. Therefore, only one CaoProvController can be added.

CaoProvFile:

BACnet is divided into three levels: BACnet controllers that identify the data, the type of the object, and the object. Providers must create a CaoProvFile for every tier as follows:

ControllerFile:

A CaoProvFile corresponding to the controller-hierarchy in BACnet where.ControllerFile is created directly under CaoProvController.

ObjectTypeFile:

CaoProvFile corresponding to the type of object defined in BACnet.ObjectTypeFile is created directly under ControllerFile.ObjectTypeFile can now create the objects listed in Table 1-1 below. For more information about object-type, see BACnet Publishing Specification.Table 1-1 Object list that can be created11

Table 1-1 Object list that can be created11

Object Type Number	Object type name
0	AnalogInput object-object
1	AnalogOutput object-object
2	AnalogValue object-object
3	BinaryInput object-object
4	BinaryOutput object-object
5	BinaryValue object-object
13	MultistateInput object-object
14	MultistateOutput object-object
19	MultistateValue object-object
23	Accumulator object-object
128	Weighing object
130	Power Demand Monitor Object
131	Power demand control object
132	Generator load control object

ObjectFile:

CaoProvFile corresponding to the object-hierarchy in BACnet,.ObjectFile is created directly under ObjectTypeFile.

2. Environment setup for application development

2.1. Connecting a B-BC to a Client-Based PC

BACnet providers enter BACnet using Unitec's BACnet communication API and connect to the connected B-BC, while Unitec's BACnet communication API connects to BACnet using UDP communication. 6. Appendix B describes the steps for communicating with BACnet.6.Appendix B

2.1.1. B-BC monitoring

This provider monitors the status of B-BC in BACnet networking. The monitoring method performed by this provider is described below.

2.1.1.1. Healthy Check

Who-Is/I-Am monitors B-BC to keep alive. This provider sends a Who-Is every 60 seconds, and checks B-BC alive by I-Am sent by the provider. For B-BC that do not send I-Am for 150 seconds or more, the provider determines that they are not in operation.

2.2. Setting Up a PC Developer

2.2.1. Preparing the Communication Library

In order to use BACnet providers in development PC, Windows compatible DLL provided by Unitec must be prepared in advance in the development PC. Place the files below in "C:\Windows\System32" for 32bitOS and "C:\Windows\SysWow64" for 64bitOS.

Table 2-1 Required Communication Libraries21

DLL	Description
BACnetDriver.dll	Unitec's BACnet communications library

2.2.2. Installing BACnet Providers Manually

To use BACnet providers, you must manually register the registry below. To register the registry, start the command prompt with administrator privileges and execute regsvr32 command. When executing, move to the path where the file is located or specify the file path.

Tabular 2-2 BACnet Providers22

File name	CaoProvunitecBACnet.dll
ProgID	CaoProv.unitec.BACnet
Registry registration	Regsvr32 CaoProvunitecBACnet.dll
Deleting registry entries	Regsvr32 /u CaoProvunitecBACnet.dll

3. Provider Overview

3.1. List of Methods/Properties

Table 3-1: List of methods and properties³¹

Category	Methods/Properties ²		Function	Reference
CaoWorkspace				
	AddController	M	Connect to controller	P.99
CaoController				
	FileNames	P	Get a list of connectable file names	P.1110
	AddFile	M	Add file/folder objects	P.1111
	VariableNames	P	Get list of variable names that can be connected	P.1212
	AddVariable	M	Adding Variable Objects	P.1312
	OnMessage	E	Message reception event	P.1312
	Execute	M	Execution of extended instructions	P.1312
CaoFile				
	FileNames	P	Get a list of connectable file names	P.1413
	AddFile	M	Add file/folder objects	P.1513
	VariableNames	P	Get list of variable names that can be connected	P.1615
	AddVariable	M	Adding Variable Objects	P.1716
CaoVariable				
	Value	P	Get/Set Value	P.1816

3.2. Method Properties

3.2.1. CaoWorkspace class-

3.2.1.1. AddController method

Adds a controller object to CaoWorkspace. BACnet providers refer to the parameters passed when AddController method is executed and enter BACnet. However, you cannot add more than one controller. The following describes the specifics of AddController method:

Format

CaoController AddController

```
(
    "<controller name>",           // Controller name (optional)
    "CaoProv.unitec.BACnet",      // Provider name (fixed)
    "<machine name>",             // Provider execution machine name (not used)
    "<Option>"                    // Option string
)
```

Option

The following options are specified in the option string. An option string is a string consisting of the following options connected by a comma (,).

² M:Indicates the method, P: property, and E: event.

Option	Required	Description	Value Range	Default value
MyIP	○	Own IP addressing	IPv4 format	-----
MyPort	--	Own port number	0 - 65535	47808
Broadcast	○	Broadcast address ³	IPv4 format	-----
InstanceNo	○	Own device instance number ※Device instance numbers must be set to unique values within the same network number.	0 - 4194302	-----
NetworkNo	--	Network number	0 - 2147483646	0

Examples of use

```

Private CaoEngine m_caoEng; // Engine object-object
Private CaoWorkspace m_caoWs; // Workspace object-object
Private CaoController m_caoCtrl; // Controller object-object

Enter // BACnet
Private void Entry()
{
    m_caoEng = new CaoEngine();
    m_caoWs = m_caoEng.AddWorkspace("NewWrks", "");
    m_caoCtrl = m_caoWs.AddController("BACnet", "CaoProv.unitec.BACnet", "",
        "MyIP = 192.168.100.5, MyPort = 47808,
        Broadcast = 192.168.100.255, InstanceNo = 5,
        NetworkNo = 0");
}

Public Main()
{
    // Entry
    This.Entry();
}

```

3.2.2. CaoController class-

3.2.2.1. FileNames Properties

Get the filename listing of B-BC currently on the network. The retrieved filename can be used as the first parameter of AddFile method. AddFile method

Examples of use

```

Private CaoEngine m_caoEng; // Engine object-object
Private CaoWorkspace m_caoWs; // Workspace object-object
Private CaoController m_caoCtrl; // Controller object-object

Enter // BACnet
Private void Entry()
{
    m_caoEng = new CaoEngine();
    m_caoWs = m_caoEng.AddWorkspace("NewWrks", "");
    m_caoCtrl = m_caoWs.AddController("BACnet", "CaoProv.unitec.BACnet", "",
        "MyIP = 192.168.100.5, MyPort = 47808,
        Broadcast = 192.168.100.255, InstanceNo = 5,
        NetworkNo = 0");
}

Public Main()
{
    // Entry
}

```

³ A broadcast address is a network address for all devices connected to a multiaccess communication network to receive datagrams. Normally, messages sent to a broadcast address are received by all hosts connected to the network, not specific hosts. (From Wikipedia)

```

        This.Entry();

        // Get file name list
        Var ctrlFileNames = m_caoCtrl.FileNames;
    }

```

3.2.2.2. AddFile method

In CaoController, add CaoFile objectI will. The following describes the specifics of AddFile method:

Format

CaoController AddFile

```

(
    "<filename>",           // File name (optional)
    "<Option>"             // Option string
)

```

Option

The following options are specified in the option string. However, if the filename obtained by FileNames is used, the option is automatically entered, so the option is ignored even if the option is entered. If the filename does not exist on thenetwork and you specify a filename other than the filename obtained by FileNames, be sure to specify InstanceNo option.

Option	Required	Description	Value Range	Default value
InstanseNo=	○	Specifies the device instance number of BACnet controllers to be connected.	0 - 4194302	--

Examples of use

```

Private CaoEngine m_caoEng; // Engine object-object
Private CaoWorkspace m_caoWs; // Workspace object-object
Private CaoController m_caoCtrl; // Controller object-object
Private CaoFile m_ctrlFile; // File object-object

Enter // BACnet
Private void Entry()
{
    m_caoEng = new CaoEngine();
    m_caoWs = m_caoEng.AddWorkspace("NewWrks", "");
    m_caoCtrl = m_caoWs.AddController("BACnet", "CaoProv.unitec.BACnet", "",
        "MyIP = 192.168.100.5, MyPort = 47808,
        Broadcast = 192.168.100.255, InstanceNo = 5,
        NetworkNo = 0");
}

Public Main()
{
    // Entry
    This.Entry();

    // Adding Files
    m_ctrlFile = m_caoCtrl.AddFile("Dev-100", "InstanceNo = 100");
}

```

3.2.2.3. VariableNames Properties

Gets a list of variable names that can be connected. The retrieved variable name can be used as the first argument of AddVariable method. AddVariable method

Examples of use

```

Private CaoEngine m_caoEng; // Engine object-object
Private CaoWorkspace m_caoWs; // Workspace object-object
Private CaoController m_caoCtrl; // Controller object-object

Enter // BACnet
Private void Entry()
{
    m_caoEng = new CaoEngine();
    m_caoWs = m_caoEng.AddWorkspace("NewWrks", "");
    m_caoCtrl = m_caoWs.AddController("BACnet", "CaoProv.unitec.BACnet", "",
        "MyIP = 192.168.100.5, MyPort = 47808,
        Broadcast = 192.168.100.255, InstanceNo = 5,
        NetworkNo = 0");
}

Public Main()
{
    // Entry
    This.Entry();

    // Get Variable Name List
    Var variables = m_caoCtrl.VariableNames;
}

```

3.2.2.4. AddVariable method

Adds a variable object to CaoController. For the variable name, specify the variable name described in 3.3.1.3.3.1

The following describes the specifics of AddVariable.

Format

CaoVariable AddVariable

```

(
    "<variable name>", // Variable Name
    "<Option>" // Optional String
)

```

3.2.2.5. OnMessage Events

Receives B-BC error notifications and status changes as OnMessage events. For details, see chapter 3.4.3.4

3.2.2.6. Execute method

Execute method of CaoController is the method for executing the command. For details of each command, refer to 4 Command Reference. 4Command Reference

Format Execute (<bstrCommandName:VT_BSTR>,[<vntParam : VT_VARIANT>])

<bstrCommandName> : [in] Command name
 <vntParam> : [in] Parameters

Examples

of use

```
Var result = this.ctrl.Execute("SendTimeSync ");
```

3.2.3. CaoFile class-

3.2.3.1. FileNames Properties

Gets the list of filenames that can be added at AddFile time. Only.ControllerFile tier can be used. The retrieved variable name can be used as the first argument of AddFile method.AddFile method

Examples of use

```
Private CaoEngine m_caoEng;    // Engine object-object
Private CaoWorkspace m_caoWs; // Workspace object-object
Private CaoController m_caoCtrl; // Controller object-object
Private CaoFile m_ctrlFile;    // File Objects (ControllerFile Hierarchy)

Enter // BACnet
Private void Entry()
{
    m_caoEng = new CaoEngine();
    m_caoWs = m_caoEng.AddWorkspace("NewWrks", "");
    m_caoCtrl = m_caoWs.AddController("BACnet","CaoProv.unitec.BACnet","",
                                     "MyIP = 192.168.100.5,MyPort = 47808,
                                     Broadcast = 192.168.100.255, InstanceNo = 5,
                                     NetworkNo = 0");
}

Public Main()
{
    // Entry
    This.Entry();

    // Adding Files
    m_ctrlFile = m_caoCtrl.AddFile("Dev-100", "InstanceNo = 100");

    // Get file name list
    Var fileNames = m_ctrlFile.FileNames;
}

```

3.2.3.2. AddFile method

Add a file object to CaoFile of the layers. The following describes the specifics of AddFile method:

Format

CaoFile AddFile

```
(
    "<filename>",           // File name
    "<Option>"             // Option string
)
```

□ Adding a ObjectTypeFile Hierarchy

Only certain filenames can be specified for <filename> at AddFile of adding a ObjectTypeFile tier. See Table 3-2 for the available file names. List of filenames that can be added at the table 3-2 ControllerFile level32

List of filenames that can be added at the table 3-2 ControllerFile level32

File name	Corresponding object type
AnalogInput	AnalogInput object-object
AnalogOutput	AnalogOutput object-object
AnalogValue	AnalogValue object-object
BinaryInput	BinaryInput object-object
BinaryOutput	BinaryOutput object-object
BinaryValue	BinaryValue object-object
Multi-stateInput	Multi-stateInput object-object
Multi-stateOutput	Multi-stateOutput object-object
Multi-stateValue	Multi-stateValue object-object
Accumulator	Accumulator object-object
Measurement	Weighing object
PowerDemandMonitoring	Power Demand Monitor Object
PowerDemandControl	Power demand control object
GeneratorLoadControl	Generator load control object

□ Adding a ObjectFile Hierarchy

The <filename> of the first parameter at the time of AddFile to add ObjectFile layer can be optionally set.

Option

The following are the options that you specify when you AddFile to add a ObjectFile tier:

Option	Required	Description	Value Range	Default value
ObjectInstanceNo	○	Specifies the instance number of BACnet object to be accessed.	0 - 4194303	--

Examples of use

```
Private CaoEngine m_caoEng; // Engine object-object
Private CaoWorkspace m_caoWs; // Workspace object-object
Private CaoController m_caoCtrl; // Controller object-object
```

```

Private CaoFile m_ctrlFile;           // File Objects (ControllerFile Hierarchy)
Private CaoFile m_objectTypeFile;     // File Objects (ObjectTypeFile Hierarchy)
Private CaoFile m_objectFile;        // File Objects (ObjectFile Hierarchy)

Enter // BACnet
Private void Entry()
{
    m_caoEng = new CaoEngine();
    m_caoWs = m_caoEng.AddWorkspace("NewWrks", "");
    m_caoCtrl = m_caoWs.AddController("BACnet", "CaoProv.unitec.BACnet", "",
        "MyIP = 192.168.100.5, MyPort = 47808,
        Broadcast = 192.168.100.255, InstanceNo = 5,
        NetworkNo = 0");
}

Public Main()
{
    // Entry
    This.Entry();

    // Adding Files (ControllerFile Hierarchy)
    m_ctrlFile = m_caoCtrl.AddFile("Dev-100", "InstanceNo = 100");

    // Adding Files (ObjectTypeFile Hierarchy)
    m_objectTypeFile = m_ctrlFile.AddFile("AnalogInput", "");

    // Adding Files (ObjectFile Hierarchy)
    m_objectFile = m_objectTypeFile.AddFile("AI0", "ObjectInstanceNo=0");
}

```

3.2.3.3. VariableNames Properties

Retrieves the list of variable names that can be connected. Only ObjectFile hierarchy can be used. The list of variable names to be acquired differs depending on the object type specified in ObjectTypeFile class. The retrieved variable name can be used as the first argument of AddVariable method. AddVariable method

Examples of use

```

Private CaoEngine m_caoEng;           // Engine object-object
Private CaoWorkspace m_caoWs;        // Workspace object-object
Private CaoController m_caoCtrl;     // Controller object-object
Private CaoFile m_ctrlFile;          // File Objects (ControllerFile Hierarchy)
Private CaoFile m_objectTypeFile;    // File Objects (ObjectTypeFile Hierarchy)
Private CaoFile m_objectFile;        // File Objects (ObjectFile Hierarchy)

Enter // BACnet
Private void Entry()
{
    m_caoEng = new CaoEngine();
    m_caoWs = m_caoEng.AddWorkspace("NewWrks", "");
    m_caoCtrl = m_caoWs.AddController("BACnet", "CaoProv.unitec.BACnet", "",
        "MyIP = 192.168.100.5, MyPort = 47808,
        Broadcast = 192.168.100.255, InstanceNo = 5,
        NetworkNo = 0");
}

Public Main()
{
    // Entry
    This.Entry();

    // Adding Files (ControllerFile Hierarchy)
    m_ctrlFile = m_caoCtrl.AddFile("Dev-100", "InstanceNo = 100");

    // Adding Files (ObjectTypeFile Hierarchy)
    m_objectTypeFile = m_ctrlFile.AddFile("AnalogInput", "");

    // Adding Files (ObjectFile Hierarchy)
    m_objectFile = m_objectTypeFile.AddFile("AI0", "ObjectInstanceNo=0");
}

```

```

// Get Variable Name List
Var fileVariables = m_objectFile.VariableNames;
}
    
```

3.2.3.4. AddVariable method

Adds a variable object to a CaoFile Only.ObjectFile hierarchy is available. The variable name shown in 3.3.2 is specified for the variable name.3.3.2

The following describes the specifics of AddVariable.

Format

CaoVariable AddVariable

```

(
    "<variable name>",           // Variable Name
    "<Option>"                   // Optional String
)
    
```

3.2.4. CaoVariable class-

3.2.4.1. Value Properties

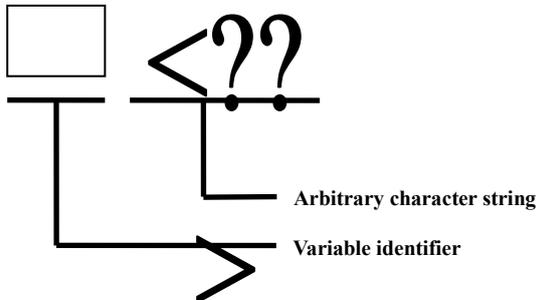
Acquires/sets the data from B-BC to be connected. The behavior varies depending on the variable name. For details, see section 3.3, Variable List.3.3Variable List

3.3. Variable List

Defines a list of variables available for each class. Variables refer to objects of CaoVariable class. Any character string can be added to register multiple variables (useful when changing only options, etc.).

The format for assigning an arbitrary string to a variable name is shown below.

Multiple Variable Common Specification Format



3.3.1. CaoController class-variable

Variable Name	Description	Value		Reference
		Get	Put	
@VERSION	Get DLL versioning.	☐	-	P.1817

3.3.1.1. @VERSION

Get the versioning of DLL.

Data type

Type	Description
VT_BSTR	Get the versioning of DLL. *.*.*

Examples of use

```

Private CaoEngine m_caoEng;           // Engine object-object
Private CaoWorkspace m_caoWs;        // Workspace object-object
Private CaoController m_caoCtrl;     // Controller object-object
Private CaoVariable m_ctrlVariable;  // Variable object-object

Enter // BACnet
Private void Entry()
{
    m_caoEng = new CaoEngine();
    m_caoWs = m_caoEng.AddWorkspace("NewWrks", "");
    m_caoCtrl = m_caoWs.AddController("BACnet", "CaoProv.unitec.BACnet", "",
        "MyIP = 192.168.100.5, MyPort = 47808,
        Broadcast = 192.168.100.255, InstanceNo = 5,
        NetworkNo = 0");
}

Public Main()
{
    // Entry
    This.Entry();

    // Add Variable
    m_ctrlVariable = m_caoCtrl.AddVariable("@VERSION");
    // Get value
    String value = m_ctrlVariable.Value;
}
    
```

3.3.2. CaoFile class-variable

Below is a list of variable names used in ObjectFile layer.

Variable Name	Description	Value		Reference
		Get	Put	
@PRESENTVALUE	Gets the current value of the object.	☐	-	P.1918
@STATUSFLAGS	Gets the state of the object.	☐	-	P.2119
PRIORITYVALUE<??>	PresentValue property of the writable object is accessed by specifying Priority (preference).	☐	☐	P.2220

3.3.2.1. @PRESENTVALUE

Retrieves the present value of BACnet object based on the information of CaoFile class..

Data type

Type Description	
VT_VARIANT	Get the present value of BACnet object I will. The data type that you receive depends on BACnet object type. Table 3-3 shows the data types corresponding to each object type. Table 3-3: List of acquired data types for each object type33

Table 3-3: List of acquired data types for each object type33

Object type	Acquired data type	Value Range
AnalogInput object-object	VT_R4	Depend on the data type
AnalogOutput object-object	VT_R4	Depend on the data type
AnalogValue object-object	VT_R4	Depend on the data type
BinaryInput object-object	VT_UI4	0 - 1
BinaryOutput object-object	VT_UI4	0 - 1
BinaryValue object-object	VT_UI4	0 - 1
MultistateInput object-object	VT_UI4	Depends on the number of Number Of State properties held by the target object.
MultistateOutput object-object	VT_UI4	Depends on the number of Number Of State properties held by the target object.
MultistateValue object-object	VT_UI4	Depends on the number of Number Of State properties held by the target object.
Accumulator object-object	VT_UI4	Depend on the data type
Weighing object	VT_UI4	Depend on the data type
Power Demand Monitor Object	VT_UI4	Depend on the data type
Power demand control object	VT_UI4	Depend on the data type
Generator load control object	VT_UI4	Depend on the data type

Examples of use

```
Private CaoEngine m_caoEng; // Engine object-object
Private CaoWorkspace m_caoWs; // Workspace object-object
Private CaoController m_caoCtrl; // Controller object-object
Private CaoFile m_ctrlFile; // File Objects (ControllerFile Hierarchy)
Private CaoFile m_objectTypeFile; // File Objects (ObjectTypeFile Hierarchy)
Private CaoFile m_objectFile; // File Objects (ObjectFile Hierarchy)
Private CaoVariable m_fileVariable; // Variable object
```

Enter // BACnet

```

Private void Entry()
{
    m_caoEng = new CaoEngine();
    m_caoWs = m_caoEng.AddWorkspace("NewWrks", "");
    m_caoCtrl = m_caoWs.AddController("BACnet", "CaoProv.unitec.BACnet", "",
        "MyIP = 192.168.100.5, MyPort = 47808,
        Broadcast = 192.168.100.255, InstanceNo = 5,
        NetworkNo = 0");
}

Public Main()
{
    // Entry
    This.Entry();

    // Adding Files (ControllerFile Hierarchy)
    m_ctrlFile = m_caoCtrl.AddFile("Dev-100", "InstanceNo = 100");

    // Adding Files (ObjectTypeFile Hierarchy)
    m_objectTypeFile = m_ctrlFile.AddFile("AnalogInput", "");

    // Adding Files (ObjectFile Hierarchy)
    m_objectFile = m_objectTypeFile.AddFile("AI0", "ObjectInstanceNo=0");

    // Add Variable
    m_fileVariable = m_objectFile.AddVariable("@PRESENTVALUE");
    // Get value
    Float value = m_fileVariable.value;
}

```

3.3.2.2. @STATUSFLAGS

Gets the status of BACnet object based on CaoFile data.

Data type

Type Description	
VT_UI4	<p>Gets the status of BACnet object..</p> <p>The status value is represented by the lower 4 bits as bit flags. When the flag is True, the target bit is set to 1.</p> <p>Lower 1 bit: Alarm status</p> <p>Lower 2 bits: Device status</p> <p>Lower 3rd bit: Invalid</p> <p>Lower 4th bit: Maintenance status</p>

Examples of use

```

Private CaoEngine m_caoEng; // Engine object-object
Private CaoWorkspace m_caoWs; // Workspace object-object
Private CaoController m_caoCtrl; // Controller object-object
Private CaoFile m_ctrlFile; // File Objects (ControllerFile Hierarchy)
Private CaoFile m_objectTypeFile; // File Objects (ObjectTypeFile Hierarchy)
Private CaoFile m_objectFile; // File Objects (ObjectFile Hierarchy)
Private CaoVariable m_fileVariable; // Variable object-object

Enter // BACnet
Private void Entry()
{
    m_caoEng = new CaoEngine();
    m_caoWs = m_caoEng.AddWorkspace("NewWrks", "");
    m_caoCtrl = m_caoWs.AddController("BACnet", "CaoProv.unitec.BACnet", "",
        "MyIP = 192.168.100.5, MyPort = 47808,
        Broadcast = 192.168.100.255, InstanceNo = 5,
        NetworkNo = 0");
}

```

```

}

Public Main()
{
    // Entry
    This.Entry();

    // Adding Files (ControllerFile Hierarchy)
    m_ctrlFile = m_caoCtrl.AddFile("Dev-100", "InstanceNo = 100");

    // Adding Files (ObjectTypeFile Hierarchy)
    m_objectTypeFile = m_ctrlFile.AddFile("AnalogInput", "");

    // Adding Files (ObjectFile Hierarchy)
    m_objectFile = m_objectTypeFile.AddFile("AI0", "ObjectInstanceNo=0");

    // Add Variable
    m_fileVariable = m_objectFile.AddVariable("@STATUSFLAGS");
    // Get value
    Ulong value = m_fileVariable.value;
}

```

3.3.2.3. PRIORITYVALUE<??>

Gets/sets the value stored in the priority order specified by the option. However, only objects that have PriorityArray properties (precedence table) are supported. See Table 3-4 for the supported objects. In addition, the following describes the properties of the priority table. Tabular 3-4 PRIORITYVALUE Supported Objects34

Tabular 3-4 PRIORITYVALUE Supported Objects34

Object type
AnalogOutput object-object
AnalogValue object-object
BinaryOutput object-object
BinaryValue object-object
MultistateOutput object-object
MultistateValue object-object

PriorityArray Properties (Priority Table)

PriorityArray property is used to set the instruction precedence. You can set a priority from 1 to 16, with the lowest number being the highest priority.

If PriorityArray property has several values with varying priorities, the setting with the highest priority takes effect and is reflected in the present value.

There are two operation methods for setting the instruction priority: 16-level instruction priority method and fixed-level instruction priority method. The only rule for AnalogOutput objects is the fixed-level instruction precedence method. For detailed information on the operation method, refer to BACnet publication spec of Azbil Corporation.

When the value of the instruction priority is empty, the value can be empty by setting NULL to the priority to be set.

Option

The following options are specified during AddVariable:

Option	Required	Description	Value Range	Default value
Priority=	--	Specify the priority. The lower the value, the higher the priority.	1 - 16	8

Data type

Type	Description
VT_UI4	Gets/sets the present value of BACnet object.. The data type differs depending on BACnet object type to be acquired or set. See Table 3-3 for the supported data types for each object. If no value is stored in the specified priority, VT_NULL is set as the data at acquisition, and when VT_NULL is set, the value stored in the specified priority is initialized by NULL. Table 3-3: List of acquired data types for each object type33
Or	
VT_R4	
Or	
VT_NULL	

Examples of use

```

Private CaoEngine m_caoEng; // Engine object-object
Private CaoWorkspace m_caoWs; // Workspace object-object
Private CaoController m_caoCtrl; // Controller object-object
Private CaoFile m_ctrlFile; // File Objects (ControllerFile Hierarchy)
Private CaoFile m_objectTypeFile; // File Objects (ObjectTypeFile Hierarchy)
Private CaoFile m_objectFile; // File Objects (ObjectFile Hierarchy)
Private CaoVariable m_fileVariable; // Variable object-object

Enter // BACnet
Private void Entry()
{
    m_caoEng = new CaoEngine();
    m_caoWs = m_caoEng.AddWorkspace("NewWrks", "");
    m_caoCtrl = m_caoWs.AddController("BACnet", "CaoProv.unitec.BACnet", "",
        "MyIP = 192.168.100.5, MyPort = 47808,
        Broadcast = 192.168.100.255, InstanceNo = 5,
        NetworkNo = 0");
}

Public Main()
{
    // Entry
    This.Entry();

    // Adding Files (ControllerFile Hierarchy)
    m_ctrlFile = m_caoCtrl.AddFile("Dev-100", "InstanceNo = 100");

    // Adding Files (ObjectTypeFile Hierarchy)
    m_objectTypeFile = m_ctrlFile.AddFile("AnalogInput", "");

    // Adding Files (ObjectFile Hierarchy)
    m_objectFile = m_objectTypeFile.AddFile("AI0", "ObjectInstanceNo=0");

    // Add priority variable
    m_fileVariable = this.m_objectFile.AddVariable("PRIORITYVALUE_8", null);

    // Get value
    Float value = m_fileVariable.Value;
}

```

3.4. Event List

Controller error notifications and status changes can be received as OnMessage events.

Number	Description
0	An error occurred during message processing. Holds the error code as data.
1	The status of BACnet controllers in BACnet system has changed. Refer to 3.4.1.1 for the data type.3.4.1.1
2	An COV notification was received. Refer to 3.4.1.2 for the data type.3.4.1.2
3	An Event notification was received. Refer to 3.4.1.3 for the data type.3.4.1.3

3.4.1. Details of the data of the message that occurred

The details of the message data that occurred are described below.

3.4.1.1. BACnet Controllers Status Change

The present state of BACnet controllers whose status has changed and the previous state are treated as VARIANT types.

Data type

Type Description		
VT_ARRAY VT_VARIANT		
0	VT_UI4	Device instance number of BACnet controller whose status has changed
1	VT_I4	Previous state 0: Operational capability 1: Operational read-only 2: Download required 3: Downloading 4: Non-operation 5: Backup in progress 6: Undefined
2	VT_I4	Current status 0: Operational capability 1: Operational read-only 2: Download required 3: Downloading 4: Non-operation 5: Backup in progress 6: Undefined

3.4.1.2. COV notification information

VARIANT type handles the data received in COV notification.

Data type

Type Description		
VT_ARRAY VT_VARIANT		
0	VT_UI4	Process ID
1	VT_UI4	Device instance number of the notifier BACnet controller
2	VT_UI4	Object Type Number
3	VT_UI4	Object Instance Number
4	VT_UI4	Time Remaining
5	VT_UI4	Number of properties
6	VT_ARRAY VT_VARIANT	
	i	VT_VARIANT 3.3.2.1

i: Number of properties

3.4.1.3. Event notification information

VARIANT type handles the data received in Event notification.

Data type

Type Description				
VT_ARRAY VT_VARIANT				
0	VT_UI4	Process ID		
1	VT_UI4	Device instance number of the notifier BACnet controller		
2	VT_UI4	Object Type Number		
3	VT_UI4	Object Instance Number		
4	VT_ARRAY VT_VARIANT		Time stamp	
	0	VT_I4	Type 0. Time 1. Unsigned 2. Datetime	
	1	VT_ARRAY VT_I4		Time (if type is 0)
		0	VT_I4	Hour
		1	VT_I4	Minute
		2	VT_I4	Seconds
		3	VT_I4	1/100 second
	1	VT_UI4	Sequence number (when the type is 1)	
	1	VT_ARRAY VT_VARIANT		
		0	VT_ARRAY VT_I4	Date (if type is 2)

Type Description						
			0	VT_I4	Year	
			1	VT_I4	Month	
			2	VT_I4	Day	
			3	VT_I4	Week 1. Monday 2. Tuesday 3. Wednesday 4. Thursday 5. Friday 6. Saturday 7. Sunday	
		1	VT_ARRAY VT_I4		Time	
			0	VT_I4	Hour	
			1	VT_I4	Minute	
			2	VT_I4	Seconds	
			3	VT_I4	1/100 second	
		5	VT_UI4		NotificationClass instance number	
6	VT_UI4		Priority			
7	VT_I4		Event type			
8	VT_VARIANT VT_ARRAY		Message text			
	0	VT_I4		Character code 0. ANSIX3.4 1. DBCS 2. JISC6226 3. ISO10646(UCS-4) 4. ISO10646(UCS-2) 5. ISO-8859-1		
		1	VT_I4		Code page ※EMPTY if the character code is not 1.	
		2	VT_I4		String byte length	
		3	VT_BSTR		Message	
9	VT_I4		Notification Type 0. Alarm 1. Event 2. Request for acceptance			

Type Description		
10	VT_I4	Request for acceptance 0. FALSE 1. TURE ※A negative value means not received.
11	VT_I4	Previous state 0. Normal 1. Equipment failure 2. Error status 3. Upper limit error 4. Lower limit error 5. Life safety alarm ※A negative value means not received.
12	VT_I4	Current status 0. Normal 1. Equipment failure 2. Error status 3. Upper limit error 4. Lower limit error 5. Life safety alarm
13	VT_I4	Notification Parameter Type 1. State change 2. Change in the value 3. Command failure 4. Floating limit 5. Out of range 10. Buffer ready 11. Unsigned range ※A negative value means not received. If the notification parameter type is negative, [14] is set to VT_EMPTY.
14	VT_ARRAY VT_VARIANT	When the notification parameter type is 1

Type Description			
0	VT_UI4	New status type 0. Boolean values 1. Binary 2. Event type 3. Polarity 4. Program change 5. Program status 6. Reason for the failure 7. Reliability 8. State 9. Equipment status 10. Unit 11. Unsigned value 12. Life safety mode 13. Life safety condition 14. Reason for restart	
	1	VT_UI4 Status data corresponding to the above status types	
	2	VT_ARRAY VT_I4	
	0	VT_I4 Alarm status	
	1	VT_I4 Equipment status	
	2	VT_I4 Invalid state	
	3	VT_I4 Maintenance status	
	14	VT_ARRAY VT_VARIANT	When the notification parameter type is 2
	0	VT_UI4	Type 0. BitString 1. Real number
	1	VT_VARIANT	Data ※Depend on the type 0. VT_UI4 1. VT_R4
	2	VT_ARRAY VT_I4	
	0	VT_I4 Alarm status	
	1	VT_I4 Equipment status	
	2	VT_I4 Invalid state	
3	VT_I4 Maintenance status		

Type Description		
14	VT_ARRAY VT_VARIANT	When the notification parameter type is 3
	0	VT_VARIANT Command value ※The data type changes depending on the data type that is received.
	1	VT_ARRAY VT_I4
	0	VT_I4 Alarm status
	1	VT_I4 Equipment status
	2	VT_I4 Invalid state
	3	VT_I4 Maintenance status
	2	VT_VARIANT Equipment status ※The data type changes depending on the data type that is received.
14	VT_ARRAY VT_VARIANT	When the notification parameter type is 4
	0	VT_R4 Reference value
	1	VT_R4 Set value
	2	VT_R4 Error Limit
	3	VT_ARRAY VT_I4
	0	VT_I4 Alarm status
	1	VT_I4 Equipment status
	2	VT_I4 Invalid state
	3	VT_I4 Maintenance status
14	VT_ARRAY VT_VARIANT	When the notification parameter type is 5
	0	VT_R4 Excess value
	1	VT_R4 Threshold
	2	VT_R4 Overrun limit
	3	VT_ARRAY VT_I4
	0	VT_I4 Alarm status
	1	VT_I4 Equipment status
	2	VT_I4 Invalid state
	3	VT_I4 Maintenance status
14	VT_ARRAY VT_VARIANT	When the notification parameter type is 10
	0	VT_ARRAY VT_VARIANT
	0	VT_UI4 Device instance number
	1	VT_UI4 Object Type Number
	2	VT_UI4 Object Instance Number
	3	VT_I4 Property number

Type Description			
	4	VT_I4	Number of arrays ※ A negative value means no array count is used.
	1	VT_UI4	Last notification
	2	VT_UI4	This notice
14	VT_ARRAY VT_VARIANT		When the notification parameter type is 11
	0	VT_UI4	Excess value
	1	VT_UI4	Overrun limit
	2	VT_ARRAY VT_I4	
	0	VT_I4	Alarm status
	1	VT_I4	Equipment status
	2	VT_I4	Invalid state
	3	VT_I4	Maintenance status

4. Command Reference

4.1. CaoController class-

List of 4-1 CaoController Class Commands41

Command	Function	Page
SendTimeSync	Send time synchronization request by broadcast communication.	29
SendTimeSyncUnicast	Sends a time synchronization request to the specified device.	29
SendUTCTimeSync	UTC time synchronization is requested by broadcast communication.	30

4.1.1. SendTimeSync Commands

Send time synchronization request by broadcast communication.

The time to synchronize is sent in local time.

Format SendTimeSync()

Return value : None

Examples

of use

```
This.ctrl.Execute("SendTimeSync", ":START");
```

4.1.2. SendTimeSyncUnicast Commands

Sends a time synchronization request to the specified device.

Format SendTimeSyncUnicast(<Device ID>)

Device ID : [in] Destination device ID (VT_UI4)

Return value : None

Examples

of use

```
This.ctrl.Execute("SendTimeSyncUnicast", "1");
```

4.1.3. SendUTCTimeSync Commands

Send time synchronization request by broadcast communication.

The time to synchronize is sent in Coordinated Universal Time.

Format SendUTCTimeSync(<command>)

Return value : None

Examples

of use

```
This.ctrl.Execute("SendUTCTimeSync");
```

5. Programming by BACnet providers

With BACnet providers, you can enter PC into BACnet as follows:

- Creating a CaoEngine
- Creating a CaoWorkspace
- Creating a CaoController

After entering BACnet, CaoFile can be accessed by creating a 3-tier CaoVariable object.

5.1. Sample programming to get/set the value stored in object priority

This example shows a sample program that acquires and sets the values stored in AnalogOutput object-priority order. Table 5-1 describes the requirements of the sample program, and Figure 5-1 describes the flow of the sample program. Table 5-1. Sample Program Requirements

Flow of acquiring the present value of 5-1 AnalogInput object.

Table 5-1. Sample Program Requirements

Requirements	Description
Own settings	Its own IP is 192.168.100.5
	The port number is 47808.
	The broadcast address is 192.168.100.255
	Its instance number is 5
	The network number is 0.
Process Description	Reads the value stored in AnalogOutput object-precedence.
	Writes the retrieved value + 1 to the value stored in AnalogOutput object-precedence.

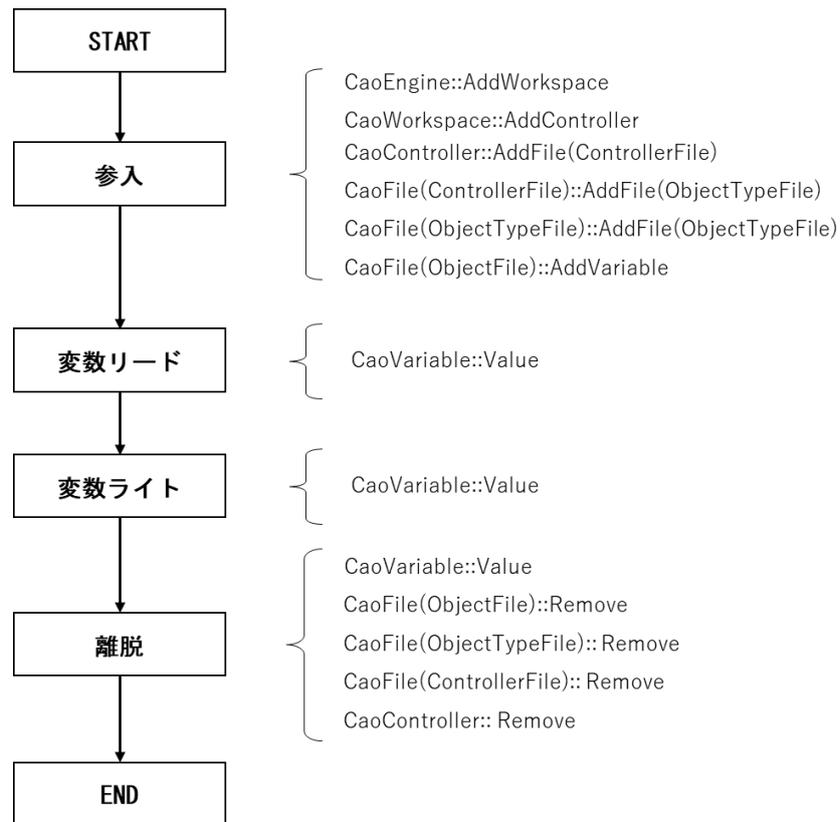


Fig. Flow of acquiring the present value of 5-1 AnalogInput object.51

The following sections show specific codes.

5.1.1.1. Sample program

An overview of the sample program is shown below.

Sample	GetPutPriorityValue.cs
--------	------------------------

```
// Object
```

```
Private CaoEngine m_caoEng;
```

```
Private CaoWorkspace m_caoWs;
```

```
Private CaoController m_caoCtrl;
```

```
Private CaoFile m_ctrlFile;
```

```
Private CaoFile m_objectTypeFile;
```

```
Private CaoFile m_objectFile;
```

```
Private CaoVariable m_priorityValue;
```

```
Public GetPutPriorityValue()
```

```
{
```

```
    // Entry
```

```
    This.Entry();
```

```
Add // CaoFile
This.AddCaoFiles();

// Add priority variable
This.m_priorityValue = this.m_objectFile.AddVariable("PRIORITYVALUE_8", null);

// Value
Float value;

// Get value
Value = this.m_priorityValue.Value;
// Add 1 to the acquired data.
Value += 1;

// Set the value
This.m_priorityValue.Value = value;

// Re-acquisition
Value = this.m_priorityValue.Value;

Remove CaoVariable from // CaoFile(ObjectFile layer)
This.m_objectFile.Variables.Remove(this.m_priorityValue.Index);

Clear // CaoVariable
System.Runtime.InteropServices.Marshal.ReleaseComObject(this.m_priorityValue);
This.m_priorityValue = null;

// CaoFile release
This.RemoveCaoFiles();

// Detachment
This.BreakAway();
}

Enter // BACnet
Private void Entry()
{
```

```

This.m_caoEng = new CaoEngine();
This.m_caoWs = this.m_caoEng.AddWorkspace("NewWrks", "");
This.m_caoCtrl = this.m_caoWs.AddController("BACnet",
                                           "CaoProv.unitec.BACnet",
                                           "",
                                           "MyIP=192.168.100.5,MyPort=47808,
                                           Broadcast=192.168.100.255,InstanceNo=5,
                                           NetworkNo=0");
}

```

Leave // BACnet

```
Protected void BreakAway()
```

```

{
    If (this.m_caoCtrl != null)
    {
        Remove CaoController from // CaoWorkspace
        This.m_caoWs.Controllers.Remove(this.m_caoCtrl.Name);
        Clearing // CaoController
        System.Runtime.InteropServices.Marshal.ReleaseComObject(this.m_caoCtrl);
        This.m_caoCtrl = null;
    }
    Remove CaoWorkspace from // CaoEngine
    This.m_caoEng.Workspaces.Remove(this.m_caoWs.Name);
    Clear // CaoWorkspace
    System.Runtime.InteropServices.Marshal.ReleaseComObject(this.m_caoWs);
    This.m_caoWs = null;
    Remove // CaoEngine
    System.Runtime.InteropServices.Marshal.ReleaseComObject(this.m_caoEng);
    This.m_caoEng = null;
}

```

Add // CaoFile

```
Private void AddCaoFiles()
```

```

{
    This.m_ctrlFile = this.m_caoCtrl.AddFile("DEV-100", "InstanceNo=100");
    This.m_objectTypeFile = this.m_ctrlFile.AddFile("AnalogOutput", "");
    This.m_objectFile = this.m_objectTypeFile.AddFile("AO0", "ObjectInstanceNo=0");
}

```

```

}

// CaoFile release
Private void RemoveCaoFiles()
{
    If (this.m_objectFile != null)
    {
        Remove CaoFile(ObjectFile Hierarchy from // CaoFile(ObjectTypeFile Hierarchy
        This.m_objectTypeFile.Files.Remove(this.m_objectFile.Index);
        Delete // CaoFile(ObjectFile layer)
        System.Runtime.InteropServices.Marshal.ReleaseComObject(this.m_objectFile);
        This.m_objectFile = null;
    }
    If (this.m_objectTypeFile != null)
    {
        Remove CaoFile(ObjectTypeFile Hierarchy from // CaoFile(ControllerFile Hierarchy
        This.m_ctrlFile.Files.Remove(this.m_objectTypeFile.Index);
        Erase // CaoFile(ObjectTypeFile layer)
        System.Runtime.InteropServices.Marshal.ReleaseComObject(this.m_objectTypeFile);
        This.m_objectTypeFile = null;
    }
    If (this.m_ctrlFile != null)
    {
        Remove CaoFile(ControllerFile Hierarchy from // CaoController
        This.m_caoCtrl.Files.Remove(this.m_ctrlFile.Index);
        Delete // CaoFile(ObjectFile layer)
        System.Runtime.InteropServices.Marshal.ReleaseComObject(this.m_ctrlFile);
        This.m_ctrlFile = null;
    }
}

```

5.1.1.1. Entry

To enter BACnet, take the following steps:

- (1) Prepare a variable to hold the object. The objects required for controller entry are CaoEngine object, CaoWorkspace object, and CaoController object..CaoWorkpace object does not need to be provided if CaoController object is retrieved from CaoWorkspaces.

The following code sample shows C#.

```
Private CaoEngine m_caoEng;           // Engine object-object
Private CaoWorkspace m_caoWs;        // Workspace object-object
Private CaoController m_caoCtrl;     // Controller object-object
```

- (2) Creates a CaoEngine object. A.CaoEngine object is generated using new keyword.

```
Generating // CaoEngine Objects
This.m_caoEng = new CaoEngine();
```

- (3) Get or Generate CaoWorkspace Objects When you create a.CaoEngine object, it defaults to creating one CaoWorkspaces object and one CaoWorkspace object. The following code sample shows a new CaoWorkspace object. The default CaoWorkspace is shown below.

```
Generating // CaoWorkspace Objects
This.m_caoWs = this.m_caoEng.AddWorkspace("NewWrks", "");
```

- (4) Create a CaoController Object To generate a.CaoController object, set the provider name to use and the parameters to use. For.BACnet provider, optionally specify its IP address, its port number, broadcast address, its instance number, and network number.

The example creates a CaoController with the following specifications:

- Your IP: 192.168.100.5,
- Its own port number: 47808
- Broadcast address: 192.168.100.255
- Your instance number: 5
- Network number: 0

```
Generating // CaoController Objects
This.m_caoCtrl = this.m_caoWs.AddController("BACnet",
                                           "CaoProv.unitec.BACnet",
                                           "",
                                           "MyIP=192.168.100.5,MyPort=47808,
                                           Broadcast=192.168.100.255,InstanceNo=5,
                                           NetworkNo=0");
```

5.1.1.2. Adding CaoFile Objects (3-tier)

To specify BACnet for individual BACnet controllers, proceed as follows:

- (1) Prepare variables to hold objects The objects required to specify.BACnet objects are CaoFile objects (ControllerFile hierarchy), CaoFile objects (ObjectTypeFile hierarchy), and ObjectFile objects (OOF hierarchy). You also need a CaoVariable object for accessing the variable. The following code sample shows C#.

```
Private CaoFile m_ctrlFile;           // File Objects (ControllerFile Hierarchy)
Private CaoFile m_objectTypeFile;     // File Objects (ObjectTypeFile Hierarchy)
Private CaoFile m_objectFile;         // File Objects (ObjectFile Hierarchy)
Private CaoVariable m_priorityValue;  // Variable object-object
```

- (2) Create a CaoFile Object (ControllerFile Hierarchy) To generate a.CaoFile Object (ControllerFile Hierarchy), configure any filename and the instance number of BACnet Controllers you want to connect to. The code example is shown below.

```
Generating // CaoFile Objects (ControllerFile Hierarchy)
```

```
This.m_ctrlFile = this.m_caoCtrl.AddFile("DEv-100", "InstanceNo=100");
```

- (3) Generate CaoFile Objects (ObjectTypeFile Hierarchy) To generate a.CaoFile Object (ObjectTypeFile Hierarchy), set the type of object to be accessed by filename. The code example is shown below.

```
Generating // CaoFile Objects (ObjectTypeFile Hierarchy)
This.m_objectTypeFile = this.m_ctrlFile.AddFile("AnalogOutput", "");
```

- (4) Create a CaoFile Object (ObjectFile Hierarchy). To create a.CaoFile Object (ObjectFile Hierarchy), specify an optional filename and set the instance number of the object to be accessed to the parameter. The code example is shown below.

```
Generating // CaoFile Objects (ObjectFile Hierarchy)
This.m_objectFile = this.m_objectTypeFile.AddFile("AO0", "ObjectInstanceNo=0");
```

- (5) CaoVariable objects must be added to CaoFile object (ObjectFile tier) in order to retrieve/set the values stored in BACnet object's precedence. PRESENTVALUE <??> is added to the sample program to acquire/set the values stored in the order of precedence. Enter any character string for <??>.

```
Creating // CaoVariable Objects
This.m_priorityValue = this.m_objectFile.AddVariable("PRIORITYVALUE_8", null);
```

5.1.1.3. Get/Set Value Stored in Priority

To get/set the value stored in the priority level, refer/set Value property of CaoVariable object. To get/set the value stored in the priority level of.AnalogOutput object, you need to prepare a 32-bit floating point type. The code example is shown below.

```
// Value
Float value;

// Get value
Value = this.m_priorityValue.Value;

// Add 1 to the acquired data.
Value += 1;

// Set the value
This.m_priorityValue.Value = value;

// Re-acquisition
Value = this.m_priorityValue.Value;
```

5.1.1.4. Deleting CaoFile Objects (3-tier)

Deletes the added 3-tier CaoFile object. The code example is shown below.

```
Remove CaoFile(ObjectFile Hierarchy from // CaoFile(ObjectTypeFile Hierarchy)
This.m_objectTypeFile.Files.Remove(this.m_objectFile.Index);
Delete // CaoFile(ObjectFile layer)
System.Runtime.InteropServices.Marshal.ReleaseComObject(this.m_objectFile);
This.m_objectFile = null;

Remove CaoFile(ObjectTypeFile Hierarchy from // CaoFile(ControllerFile Hierarchy)
This.m_ctrlFile.Files.Remove(this.m_objectTypeFile.Index);
Erase // CaoFile(ObjectTypeFile layer)
System.Runtime.InteropServices.Marshal.ReleaseComObject(this.m_objectTypeFile);
This.m_objectTypeFile = null;

Remove CaoFile(ControllerFile Hierarchy from // CaoController
This.m_caoCtrl.Files.Remove(this.m_ctrlFile.Index);
Delete // CaoFile(ObjectFile layer)
System.Runtime.InteropServices.Marshal.ReleaseComObject(this.m_ctrlFile);
```

```
This.m_ctrlFile = null;
```

5.1.1.5. Detachment

If you are disconnecting from the controller, erase the generated object and remove the object from the collection class that manages the object.

The code example is shown below.

```
Remove CaoController from // CaoWorkspace
This.m_caoWs.Controllers.Remove(this.m_caoCtrl.Name);
Clearing // CaoController
System.Runtime.InteropServices.Marshal.ReleaseComObject(this.m_caoCtrl);
This.m_caoCtrl = null;

Remove CaoWorkspace from // CaoEngine
This.m_caoEng.Workspaces.Remove(this.m_caoWs.Name);
Clear // CaoWorkspace
System.Runtime.InteropServices.Marshal.ReleaseComObject(this.m_caoWs);
This.m_caoWs = null;

Remove // CaoEngine
System.Runtime.InteropServices.Marshal.ReleaseComObject(this.m_caoEng);
This.m_caoEng = null;
```

5.2. Sample Programming to Automatically Receive BACnet Controllers Status Changes

This example shows a sample program that retrieves the present value of AnalogInput object.. Table 5-2 describes the requirements of the sample program, and Figure 5-2 describes the flow of the sample program. Table 5-2 Sample program requirements52 Fig. Flow of acquiring status change of 5-2 BACnet controllers52

Table 5-2 Sample program requirements52

Requirements	Description
Own settings	Its own IP is 192.168.100.5
	The port number is 47808.
	The broadcast address is 192.168.100.255
	Its instance number is 5
	The network number is 0.
Process Description	Receive status change from BACnet controllers in OnMessage

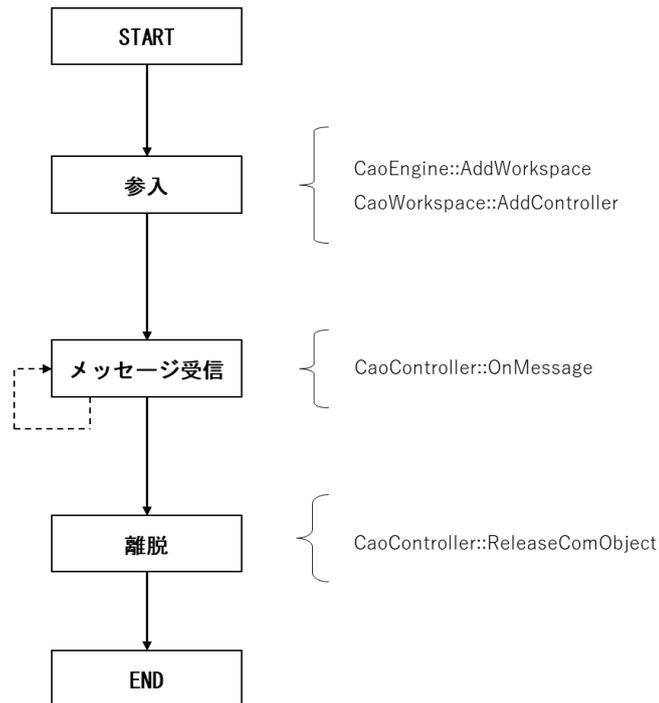


Fig. Flow of acquiring status change of 5-2 BACnet controllers52

The following sections show specific codes.

5.2.1. Sample program

An overview of the sample program is shown below.

Sample	ChangeSystemStatusFeature.cs
<pre> // Object Private CaoEngine m_caoEng; Private CaoWorkspace m_caoWs; Private CaoController m_caoCtrl; Public ChangeSystemStatusFeature() { // Entry This.Entry(); Registering // OnMessage event handlers This.m_caoCtrl.OnMessage += New _ICaoControllerEvents_OnMessageEventHandler(OnMessage); // Detachment This.BreakAway(); </pre>	

```

}

// Message receiving method
Private void OnMessage(CaoMessage pICaoMsg)
{
    If (pICaoMsg.Number == 1) // When a device status change notification message is received
    {
        Object[] msgValues = new object[3];
        MsgValues = pICaoMsg.Value;

        // Notifying source instance number
        Ulong deviceInstanceNo = Convert.ToUInt64(msgValues[0]);

        // Previous state value
        Int oldStatus = Convert.ToInt32(msgValues[1]);

        // Current state value
        Int nowStatus = Convert.ToInt32(msgValues[2]);
    }
}

Enter // BACnet
Private void Entry()
{
    This.m_caoEng = new CaoEngine();
    This.m_caoWs = this.m_caoEng.AddWorkspace("NewWrks", "");
    This.m_caoCtrl = this.m_caoWs.AddController("BACnet",
                                                "CaoProv.unitec.BACnet",
                                                "",
                                                "MyIP=192.168.100.5,MyPort=47808,
                                                Broadcast=192.168.100.255,
                                                InstanceNo=5,NetworkNo=0");
}

Leave // BACnet
Private void BreakAway()

```

```

{
  If (this.m_caoCtrl != null)
  {
    Remove CaoController from // CaoWorkspace
    This.m_caoWs.Controllers.Remove(this.m_caoCtrl.Index);
    Clearing // CaoController
    System.Runtime.InteropServices.Marshal.ReleaseComObject(this.m_caoCtrl);
    This.m_caoCtrl = null;
  }
  Remove CaoWorkspace from // CaoEngine
  This.m_caoEng.Workspaces.Remove(this.m_caoWs.Index);
  Clear // CaoWorkspace
  System.Runtime.InteropServices.Marshal.ReleaseComObject(this.m_caoWs);
  This.m_caoWs = null;
  Remove // CaoEngine
  System.Runtime.InteropServices.Marshal.ReleaseComObject(this.m_caoEng);
  This.m_caoEng = null;
}

```

5.2.1.1. Entry

To enter BACnet, take the following steps:

- (1) Prepare a variable to hold the object. The objects required for controller entry are CaoEngine object, CaoWorkspace object, and CaoController object..CaoWorkpace object does not need to be provided if CaoController object is retrieved from CaoWorkspaces.

The following code sample shows C#.

```

Private CaoEngine m_caoEng;           // Engine object-object
Private CaoWorkspace m_caoWs;        // Workspace object-object
Private CaoController m_caoCtrl;     // Controller object-object

```

- (2) Creates a CaoEngine object. A.CaoEngine object is generated using New keyword.

```

Generating // CaoEngine Objects
This.m_caoEng = new CaoEngine();

```

- (3) Get or Generate CaoWorkspace Objects When you create a.CaoEngine object, it defaults to creating one CaoWorkspaces object and one CaoWorkspace object. The following code sample shows a new CaoWorkspace object. The default CaoWorkspace is shown below.

```

Generating // CaoWorkspace Objects
This.m_caoWs = this.m_caoEng.AddWorkspace("NewWrks", "");

```

- (4) Create a CaoController Object To generate a.CaoController object, set the provider name to use and the parameters to use.

For.BACnet provider, optionally specify its IP address, its port number, broadcast address, its instance number, and network number.

The example creates a CaoController with the following specifications:

- Your IP: 192.168.100.5,
- Its own port number: 47808
- Broadcast address: 192.168.100.255
- Your instance number: 5
- Network number: 0

Generating ' CaoController Objects

```
This.m_caoCtrl = this.m_caoWs.AddController("BACnet",
                                         "CaoProv.unitec.BACnet",
                                         "",
                                         "MyIP=192.168.100.5,MyPort=47808,
                                         Broadcast=192.168.100.255,InstanceNo=5,
                                         NetworkNo=0");
```

5.2.1.2. Message reception

Notifications sent from other BACnet controllers can be retrieved by handling CaoController::OnMessage events. The example program provides

OnMessage event handlers to handle CaoController::OnMessage events. For the specifications of each message, refer to 3.4.3.4

```
// A message indicating a change in the device status is received.
If (pICaoMsg.Number == 1)
{
    Object[] msgValues = new object[3];
    MsgValues = pICaoMsg.Value;

    // Notifying source instance number
    ULONG deviceInstanceNo = Convert.ToUInt64(msgValues[0]);

    // Previous state value
    Int oldStatus = Convert.ToInt32(msgValues[1]);

    // Current state value
    Int nowStatus = Convert.ToInt32(msgValues[2]);
}
```

5.2.1.3. Disconnect

If you are disconnecting from the controller, erase the generated object and remove the object from the collection class that manages the object.

The code example is shown below.

```
Remove CaoController from // CaoWorkspace
This.m_caoWs.Controllers.Remove(this.m_caoCtrl.Name);
Clearing // CaoController
System.Runtime.InteropServices.Marshal.ReleaseComObject(this.m_caoCtrl);
This.m_caoCtrl = null;

Remove CaoWorkspace from // CaoEngine
This.m_caoEng.Workspaces.Remove(this.m_caoWs.Name);
Clear // CaoWorkspace
System.Runtime.InteropServices.Marshal.ReleaseComObject(this.m_caoWs);
This.m_caoWs = null;

Remove // CaoEngine
System.Runtime.InteropServices.Marshal.ReleaseComObject(this.m_caoEng);
This.m_caoEng = null;
```

6. BACnet provider error code

This provider has the following unique error code masked by 0x8011****. (Refer to Table 6-1 Unique Error Code Table.)

See the error codes section in ORiN2 Programming Guide [\(Link\)](#) for information on common errors in ORiN2. [C:\ORiN2\CAO\Doc\ORiN2_ProgrammersGuide_ja.pdf](#)

Table 6-1 Unique Error Code Table61

Error Number	Description
0x80110002	Exclusive control failed.
0x80110003	The object does not exist.

This provider returns API error code masked by "0x8010****". (Refer to the error code table from the table 6-2 API.)

Error codes from tabular 6-2 API62

Error Number	Description
0x8010FFFF	Open function has already been called in Open status.
0x8010FFFE	Close function has already been called in Close status.
0x8010FFFD	Failed to read/write the backup file.
0x8010FFFC	The information to be added has already been registered.
0x8010FFFB	The specified parameter is invalid.
0x8010FFFA	No information was found for the specified parameter.
0x8010FFF9	Starting BACnet communication failed because IP address set in ACnet.ini is not registered in PC or is already in use.
0x8010FFF8	Failed to execute due to insufficient resources.
0x8010FFF7	BACnet communication has not started.
0x8010FC19	The unlock key for unprotecting was not found. ※ Only when using a protected DLL.
0x8010D8F1	An unknown error occurred.
0x80100001	An Error reply was received from the remote device.
0x80100002	An Reject reply was received from the remote device.
0x80100003	An Abort reply was received from the remote device.
0x80100004	The request to the remote device exceeded the number of retries.
0x80100005	The remote device is disconnected from BACnet.
0x8010000D	The remote device does not exist.

Appendix A. API map

CaoWorkspace::AddController

API function name
BD_OpenDirect
BD_SetDeviceStatusChangeCallback
BD_SetCOVCallback
BD_SetEventCallback
BD_SetSystemStatus
BD_SendWhoisWithRange
BD_SendTimeSync
BD_SendIam
BD_Close

CaoController

API function name
BD_SendTimeSyncUnicast
BD_SendWhoisWithRange

CaoVariable

Variable Name	Get_Value	Set_Value
@PRESENTVALUE	BD_ReadPropertyMultiple	---
@SYSTEMSTATUS	BD_ReadPropertyMultiple	---
PRIORITYVALUE<??>	BD_ReadPropertyMultiple	BD_WritePropertyMultiple

Appendix B. Initial Procedure

When entering/leaving BACnet, this provider enters/leaves based on the procedure described in BACnet Publishing Specification Initial Procedure. Entry is based on the entry sequence, and leaving is based on the leaving sequence.

Appendix B.1. Entry sequence

The following describes the steps of this provider and other BACnet controllers when entering BACnet.

Procedure

This provider	Other BACnet controllers
1) Entry (DOWNLOAD_REQUIRED) Service = UnconfirmedCOVNotification 'Subscriber Process Identifier'= 0 'Initiating Device Identifier'= (Device, Instance N) 'Monitored Object Identifier'= (Device, Instance N) 'Time Remaining'= 0 'List of Values' = ((System_Status, DOWNLOAD_REQUIRED), (Time_Of_Device_Restart, (YYYYMMDD(W), HH:MM:SS.XX)), (Last_Restart_Reason, XXXXX))	→

This provider		Other BACnet controllers
<p>2) Send Who-Is (broadcast)</p> <p>Service=Who-Is</p> <p>Device Instance Low Limit = XX</p> <p>Device Instance Hi Limit = ZZ</p>	<p>→←</p>	<p>1) Send I-Am (broadcast)</p> <p>Service = I-Am</p> <p>'I-Am Device Identifier'=(Device, Instance N)</p> <p>'MAX APDU Length Accepted'= 1024</p> <p>'Segmentation Supported'= SEGMENTED_BOTH</p> <p>'Vendor Identifier'= 85</p>
<p>3) Determining the system time</p> <p>Service = UnconfirmedCOVNotification</p> <p>'Subscriber Process Identifier'= 0</p> <p>'Initiating Device Identifier'=(Device, Instance N)</p> <p>'Monitored Object Identifier'=(Device, Instance N)</p> <p>'Time Remaining'= 0</p> <p>'List of Values'=((System_Status, DOWNLOAD_IN_PROGLESS), (Time_Of_Device_Restart, (YYYYMMDD(W), HH:MM:SS.XX)), (Last_Restart_Reason, XXXXX))</p>	<p>→</p>	
<p>4) End of entry</p> <p>Service = UnconfirmedCOVNotification</p> <p>'Subscriber Process Identifier'= 0</p>	<p>→</p>	

This provider		Other BACnet controllers
'Initiating Device Identifier'=(Device, <div style="text-align: right;">Instance N)</div> 'Monitored Object Identifier'=(Device, Instance N) 'Time Remaining'= 0 'List of Values'= ((System_Status, OPERATIONAL), (Time_Of_Device_Restart, (YYYYMMDD(W), HH:MM:SS.XX)), (Last_Restart_Reason, XXXXX))		

Appendix B.2. Detachment sequence

The following describes the steps when the ISP leaves BACnet.

Procedure

This provider		Other BACnet controllers
1) Detachment Service = UnconfirmedCOVNotification 'Subscriber Process Identifier'= 0 'Initiating Device Identifier'=(Device, Instance N) 'Monitored Object Identifier'=(Device, Instance N) 'Time Remaining'= 0 (System Status = Non_OPERATIONAL)	→	

Appendix B.3. When BACnet controllers enter

The following describes the steps when other BACnet controllers enter BACnet.

Procedure

This provider	Other BACnet controllers
	<p>1) Entry (DOWNLOAD_REQUIRED)</p> <p>Service = UnconfirmedCOVNotification</p> <p>'Subscriber Process Identifier'= 0</p> <p>'Initiating Device Identifier'= (Device, Instance N)</p> <p>'Monitored Object Identifier'= (Device, Instance N)</p> <p>← 'Time Remaining'= 0</p> <p>'List of Values' =</p> <p>((System_Status, DOWNLOAD_REQUIRED),</p> <p>(Time_Of_Device_Restart, (YYYYMMDD(W), HH:MM:SS.XX)),</p> <p>(Last_Restart_Reason, XXXXX))</p>
<p>3) Send I-Am (broadcast)</p> <p>Service = I-Am</p> <p>'I-Am Device Identifier'= (Device,</p> <p style="padding-left: 100px;">Instance N)</p> <p>'MAX APDU Length Accepted'= 1024</p> <p>'Segmentation Supported'=</p> <p style="padding-left: 100px;">SEGMENTED__BOTH</p> <p>'Vendor Identifier'= 85</p>	<p>2) Send Who-Is (broadcast)</p> <p>Service=Who-Is</p> <p>Device Instance Low Limit = XX</p> <p>Device Instance Hi Limit = ZZ</p> <p>←</p> <p>→</p>

This provider	Other BACnet controllers
	<p>4) Terminate entry (OPERATIONAL)</p> <p>Service = UnconfirmedCOVNotification</p> <p>'Subscriber Process Identifier'= 0</p> <p>'Initiating Device Identifier'= (Device, Instance N)</p> <p>'Monitored Object Identifier'= (Device, Instance N)</p> <p>←</p> <p>'Time Remaining'= 0</p> <p>'List of Values'= ((System_Status, OPERATIONAL), (Time_Of_Device_Restart,(YYYYMMDD(W),HH:MM:SS.XX)), (Last_Restart_Reason, XXXXX))</p>