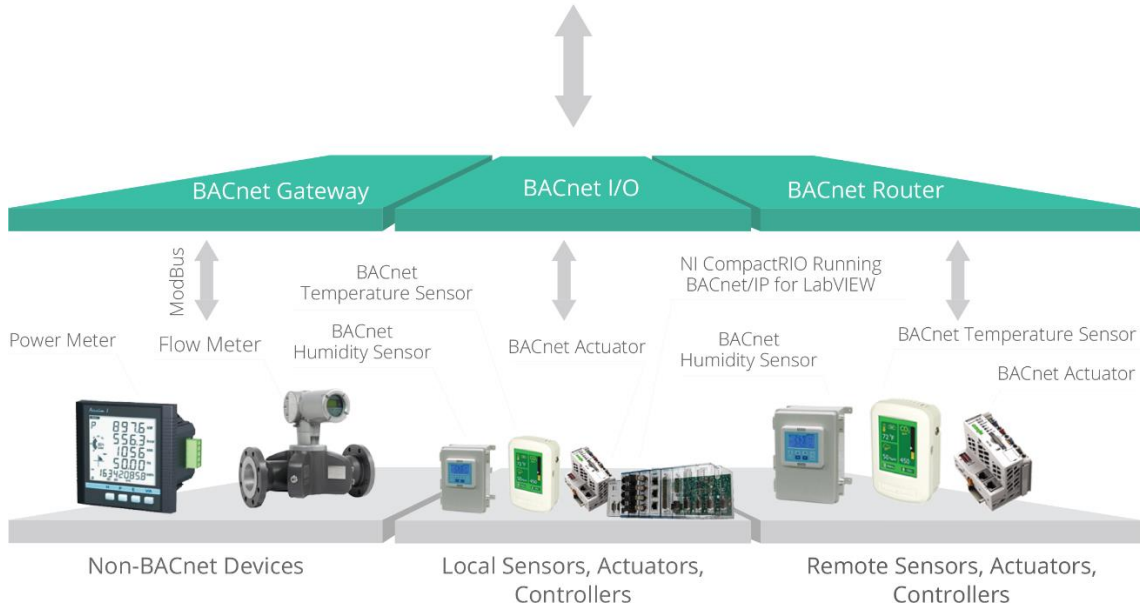


# BACnet/IP Protocol for LabVIEW

User manual



BACnet/IP Protocol for LabVIEW



## Contents

<b>1. Introduction</b> .....	3
1.1. Definitions and Acronyms .....	3
1.1. Purpose.....	4
1.2. Overview.....	4
<b>2. BACnet Background</b> .....	5
<b>3. BACnet/IP</b> .....	5
<b>4. BACnet Client and Server</b> .....	6
<b>5. BACnet Protocol Implementation Conformance Statement (PICS)</b> .....	6
<b>6. Transfer Library and License Files to NI Linux RT</b> .....	7
<b>7. BACnet/IP Protocol Palettes</b> .....	8
<b>8. Examples</b> .....	9
<b>9. System Requirements</b> .....	12
<b>10. References</b> .....	12
<b>11. Support Information</b> .....	12

## 1. Introduction

### 1.1. Definitions and Acronyms

**Device Instance** – this is the logical address that matters to BACnet. Whether on an MS/TP link or IP network, the device instance is unique across all subnets and routed links.

**Object** – the general reference to sensors, actuators, and other functional elements that make up a BACnet device. The objects fall into categories specified by BACnet protocol. Analog input object and analog output object are a couple of the most commonly used objects.

**Object Property** – each object has several properties required by BACnet protocol. The most commonly used property is *Present Value*.

*BACnet* – Building Automation and Control Networks

*HVAC&R* – Heating, Ventilating, and Air Conditioning & Refrigeration

*ASHRAE* – American Society of Heating, Refrigerating, and Air Conditioning Engineers

*ANSI* – American National Standards Institute

*OSI model* – Open Systems Interconnection model

*PICS* – Protocol Implementation Conformance Statement

*AI* – Analog Input

*AO* – Analog Output

*AV* – Analog Value

*BI* – Binary Input

*BO* – Binary Output

*BV* – Binary Value

*RP* – ReadProperty

*WP* – WriteProperty

*RF* – ReadFile

*WF* – WriteFile

*IP* – Internet Protocol

*UDP* – User Datagram Protocol

*PICS* – Protocol Implementation Conformance Statement

*DVR* – Data Value Reference

*OSI* – Open Systems Interconnection

SO – Shared Object

RT – Real Time

### 1.1. Purpose

This document is designed to guide the LabVIEW developers in using “*BACnet/IP protocol for LabVIEW*” toolkit. In addition, there are PICS documents included for both BACnet client and server.

### 1.2. Overview

The BACnet is a communication protocol that defines the services used to communicate between building automation end-devices and building control systems. The BACnet protocol specification like all protocol specification defines both how details represented on the network and the services that are used to move data from one BACnet node to another. There are no proprietary chip sets or special electronics required to implement. Thus, BACnet meets the needs of users, integrators, and equipment vendors.

Historically, building automation and control systems have used proprietary communication networks. In this kind of closed system, building automation equipment supplied from different manufacturers cannot communicate with each other. Building owners and facility managers were forced to rely on products from a single vendor. Modern building automation and control systems provide a variety of building services such as heating, ventilating, and air conditioning (HVAC), lighting, fire and life safety systems, security, and vertical transportation. There can be significant safety and operational advantages to combine these building services through integrated control networks. Closed network systems provide a major barrier to integrated building facilities with the kind of flexibility and expandability that building owners want. In order to solve these problems, ASHRAE developed BACnet, the only consensus developed communication protocol standard in the world specifically designed to meet the needs of building automation and control networks. BACnet defines a set of standard objects whose properties represent the information that is exchanged between components of the building automation system and an application layer protocol that is used to access and manipulate this information. It also provides a way to convey the information across a variety of local and wide-area networks that may be interconnected to form an internetwork.

## 2. BACnet Background

BACnet is a four layer collapsed architecture that corresponds to the physical, data link, network and application layers of the OSI model.

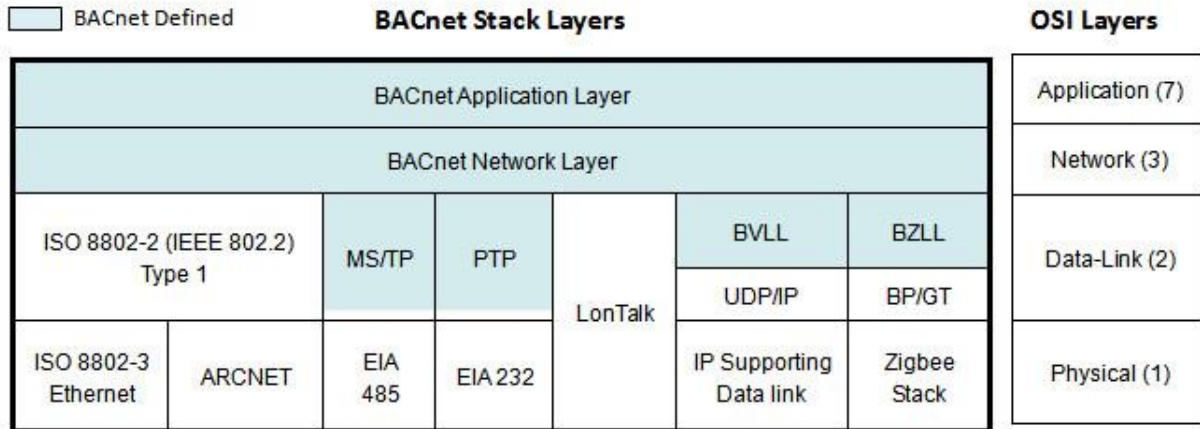


Figure 1 BACnet architecture

1. The *application layer* and *simple network layer* are defined in the BACnet standard. BACnet of version 1, revision 14 (ANSI/ASHRAE 135-2012) is implemented in *BACnet/IP Protocol for LabVIEW* toolkit.
2. The *physical layer* provides a means of connecting the devices and transmitting the electronic signals that convey the data. There may be a virtual physical layer that supports IP.
3. The *data link layer* organizes the data into frames or packets, regulates access to the medium, provides addressing, and handles some error recovery and flow control. These are all functions that are required in a BACnet protocol.

The functions provided by the network layer include translation of global addresses to local addresses, accommodating differences in network types and in the maximum message size permitted by those networks, sequencing, flow control, error control.

4. The *application layer* of the protocol provides the communication services required by the applications to perform their functions, in this case monitoring and control of *HVAC&R* and other building systems. It is clear that the application layer is needed in the protocol.

## 3. BACnet/IP

*BACnet/IP* network is a collection of one or more IP sub networks (IP domains) that are assigned to a single *BACnet* network number. *BACnet* internetwork consists of two or more *BACnet* networks. In the case of *B/IP* networks, there are six octets consisting of the four-octet IP address followed by two-octet UDP port number (both of which shall be transmitted most significant octet first). This address shall be referred to as a *B/IP* address. The default UDP port for both directed messages and broadcasts shall be *X'BAC0'* and *B/IP* devices shall support it.

#### 4. BACnet Client and Server

As BACnet is based on a "client-server" model, BACnet messages are called "service requests". A client machine sends a service request to a server machine that performs the service and reports the result to the client. BACnet defines 35 message types that are divided into 5 groups or classes. For example, one class contains messages for accessing and manipulating the properties of the objects described above. A common one is the "ReadProperty" service request. This message causes the server machine to locate the requested property of the requested object and sends its value back to the client.

#### 5. BACnet Protocol Implementation Conformance Statement (PICS)

The server in "BACnet/IP protocol for LabVIEW" toolkit supports the following methods:

- Broadcasting "I-Am" on start;
- Receiving "Who-Is" and sending "I-Am" unconfirmed request;
- Receiving "ReadProperty" and sending "ReadPropertyACK" complex ACK or error if there is any;
- Receiving "WriteProperty" and sending "WritePropertyACK" complex ACK or error if there is any;
- Receiving "CreateObject" and sending "CreateObjectACK" complex ACK or error if there is any;
- Receiving "DeleteObject" and sending "DeleteObjectACK" simple ACK or error if there is any.

Please find below the link for PICS document for *Server*:

[http://ovaktechnologies.com/wp-content/uploads/pdf/BACnet\\_IP%20PICS%20server.pdf](http://ovaktechnologies.com/wp-content/uploads/pdf/BACnet_IP%20PICS%20server.pdf)

The client in "BACnet/IP protocol for LabVIEW" toolkit supports the following methods:

- RP
- WP
- Who is
- Who has

Please find below the link for PICS document for *Client*:

[http://ovaktechnologies.com/wp-content/uploads/pdf/BACnet\\_IP%20PICS%20client.pdf](http://ovaktechnologies.com/wp-content/uploads/pdf/BACnet_IP%20PICS%20client.pdf)

## 6. Transfer Library and License Files to NI Linux RT

There are many ways of transferring .so and license files to NI Linux RT. For example *SSH*, *WebDAV* and *FileZilla*.

The NI Linux Real-Time OS offers full Linux shell support so you can perform complex administrative tasks to manage real-time targets more easily. To access the shell, you need a terminal client. If you use Windows 7 and later, then *PuTTY* is a popular and free client.

In addition to a terminal client, you must select the *Enable Secure Shell Server* (sshd) option on the real-time target running the NI Linux Real-Time OS. You can use *NI Measurement & Automation Explorer (MAX)*.

NI real-time targets support *WebDAV* file transfer. *WebDAV* is an industry-standard protocol that is based on top of *HTTP*. You can easily secure it to transfer *libBACnetIP\_Client.so*, *libBacnetIP.so* and *BacnetIP\_license.ini* files. *WebDAV* is the default file transfer mechanism for real-time targets running NI Linux Real-Time.

After installing package the *libBACnetIP\_Client.so*, *libBacnetIP.so* libraries you can find in *C:\Program Files (x86)\National Instruments\LabVIEW 2013\vi.lib\Ovak Technologies\BACnet\_IP Protocol for LabVIEW\BACnet\_IP Client\_LinuxRT\Supporting files* and *C:\Program Files (x86)\National Instruments\LabVIEW 2013\vi.lib\Ovak Technologies\BACnet\_IP Protocol for LabVIEW\BACnet\_IP\_Server\_LinuxRT\Supporting files* path. We recommend to transfer these libraries and *BacnetIP\_license.ini* files to */usr/local/natinst/lib/* then reboot the controller. You need to initialize in your program server and client side library and license file paths. For deployment license you need target's Serial Number.

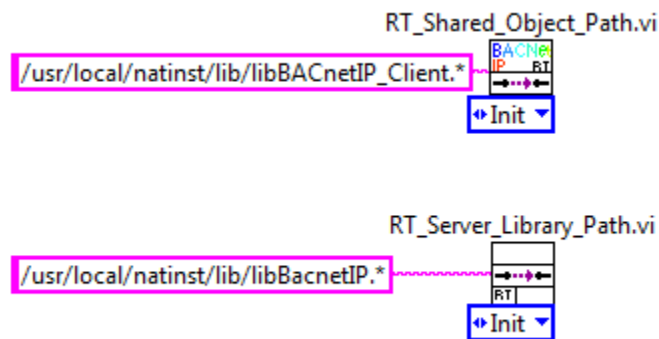


Figure 2 Initialize Library and License file path

## 7. BACnet/IP Protocol Palettes

“BACnet/IP protocol for LabVIEW” toolkit palette consists of client and server.

The client contains one VI for each service: Who-Is, Who-Has, ReadProperty, WriteProperty.

The server palette enables to set **APDU timeout (ms)**, **Vendor\_ID**, **Device\_ID**, the **Stack**, functions for stopping all stack loops and for initializing BACnet objects. Either when running “Stack.vi” it can broadcast “**I-Am**” request by local network or not, it depends on “**Send I-Am**” parameter. The next input for the stack is the data value reference on array of *BACnet\_Object objects*.

Read Property Value.vi and Write Property Value.vi are polymorphic VIs for getting and setting properties of *BACnet objects*.

The *BACnet Array palette* is included to enable the developer create, read and write data of *BACnetArray* type.

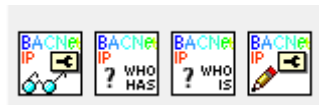


Figure 3 Client palette



Figure 4 Server palette



Figure 5 BACnetARRAY palette



## 8. Examples

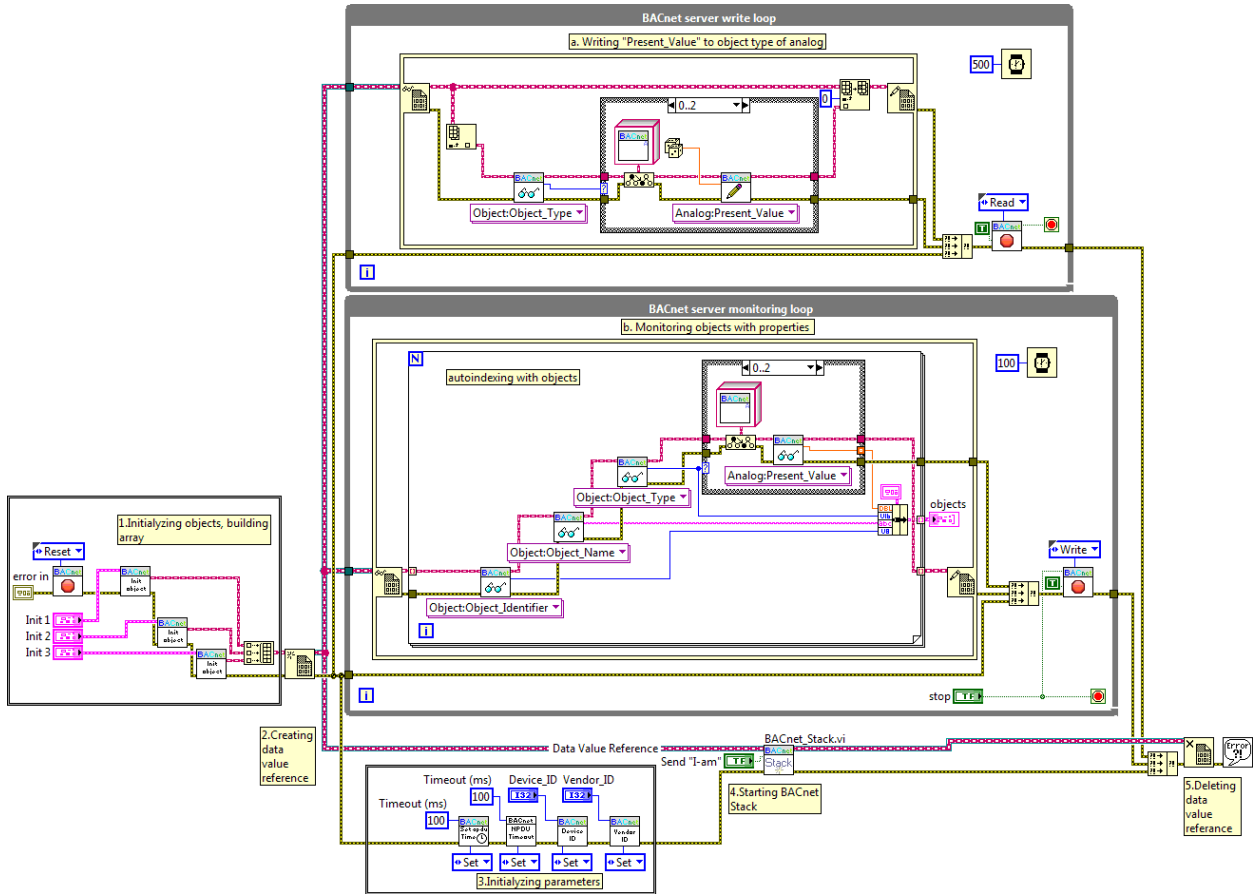


Figure 6 Example Block Diagram

**Set the initialization parameters.**  
 Init 1, Init 2 and Init 3 each one creates a BACnet object with specified parameters.  
 Set Device ID and Vendor ID for device recognition.  
 Set whether broadcast "I-am" request on start or not.

**Run the example.**  
 Follow the List of existing objects with some properties. Here you can see the following properties: object type, object name, object identifier and present value. Explore the code and try to understand the main usage principles of the read and write functions.  
 Write properties to an object and read it from a client.  
 Write properties, create and delete objects with client services and follow them on the front panel.

**Initializing server**

Init 1	Init 2	Init 3	Send "I-am"
Object_Identifier 11	Object_Identifier 1	Object_Identifier 8	<input type="checkbox"/>
Object_Name Analog In	Object_Name Analog value	Object_Name Device one	Device_ID 262626
Object_Type Analog Input	Object_Type Analog Value	Object_Type Device	Vendor_ID 1

**STOP**

**List of existing objects with some properties**

Present_Value	Object_Type	Object_Name	Object_Identifier
0	Analog Input		0
0	Analog Input		0
0	Analog Input		0
0	Analog Input		0

Figure 7 Server Example Front Panel

The “*Server*” example functions as a BACnet server. First, there are three objects initialized, there is a data value reference created for array of that objects, then the values of *apdu* timeout, vendor ID, Device ID are set, BACnet loops function FGV is reset. Before Stack is run, “Send “I am”” parameter is decided. The monitoring loop continuously reads data from *DVR*, for each object reads its object name, object type, object identifier, and present value if its type of AI of AO or AV. The write loop shows using the functionality of writing data on property of a BACnet object. In this case there is a random number written on the AI, 11 *Present\_Value* every half a second. Similarly, any property can be read and written for any BACnet object. The developer should clearly understand the hierarchy of BACnet objects to avoid unwanted errors while reading and writing properties (watch the figure “BACnet class hierarchy”).

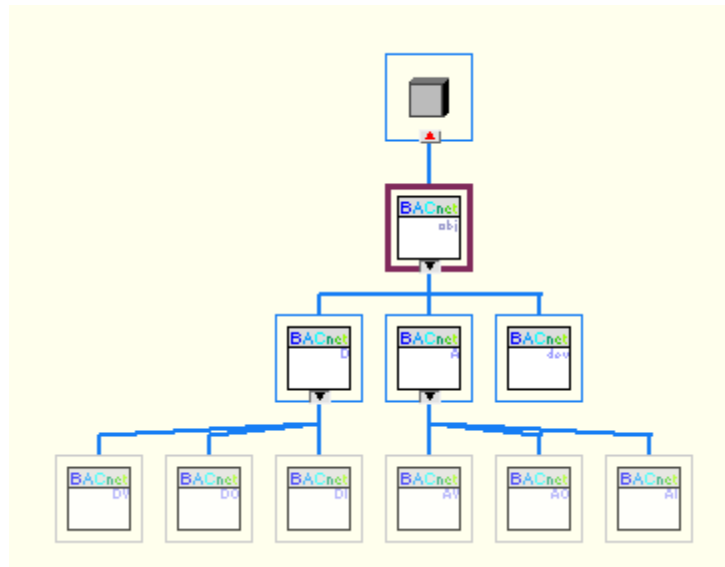


Figure 8 BACnet class hierarchy

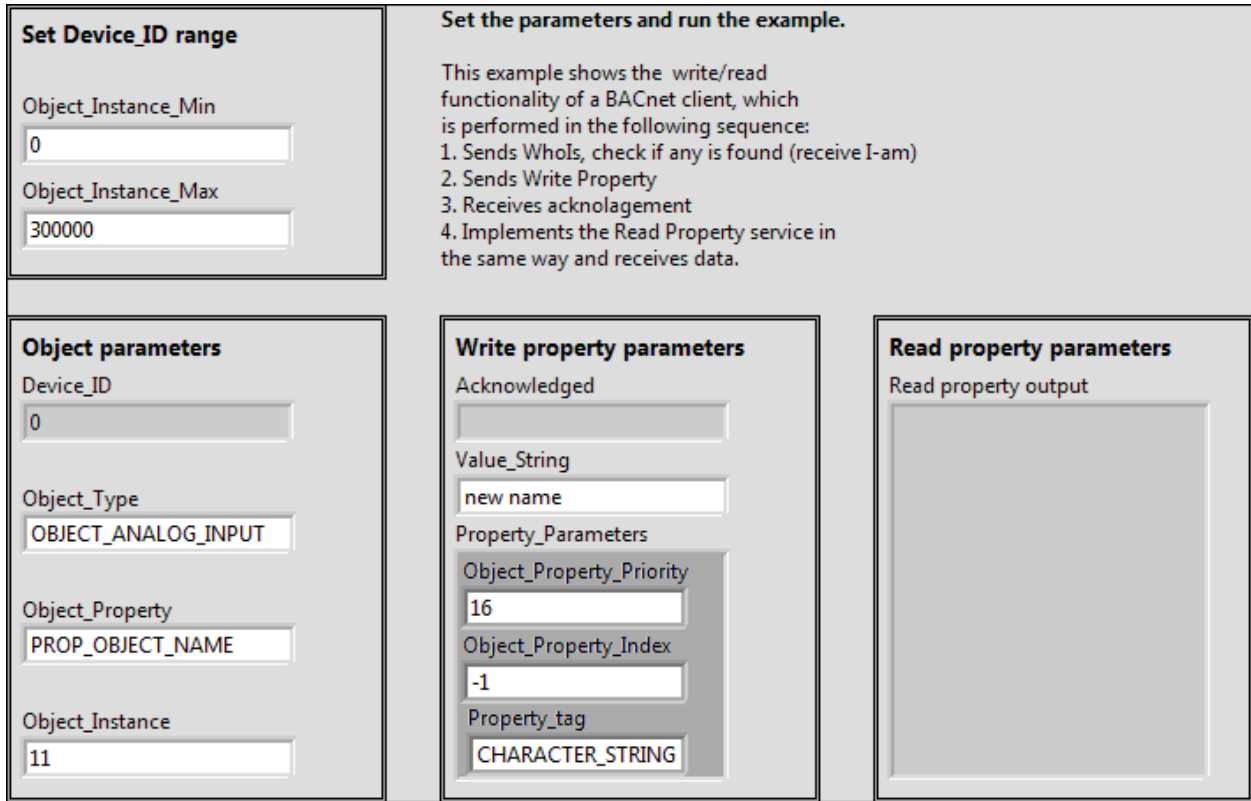


Figure 9 Client Example Front Panel

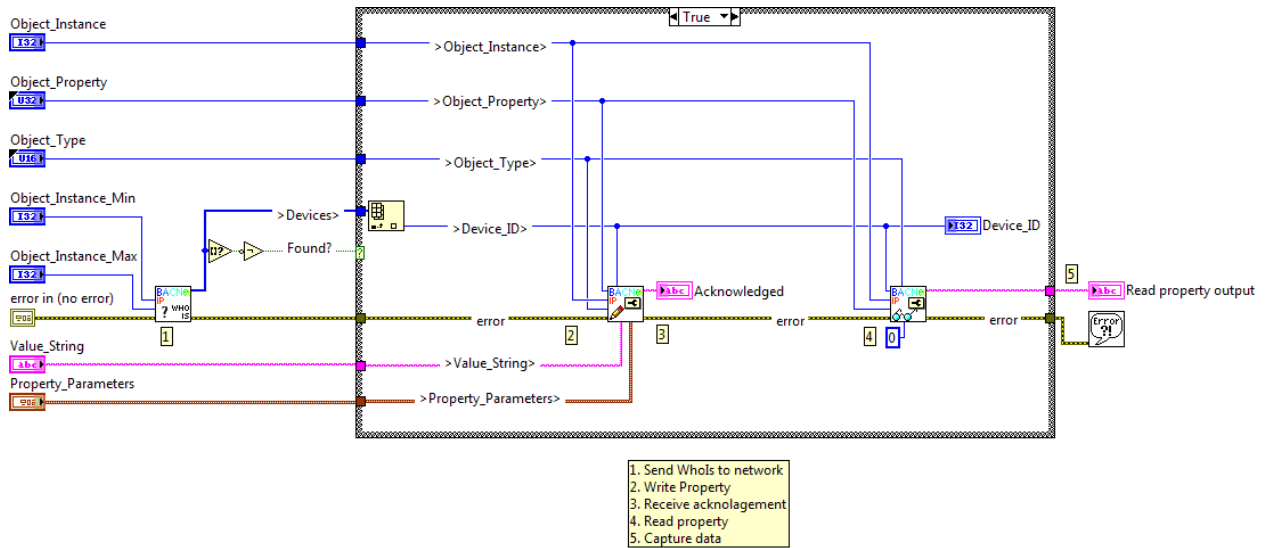


Figure 10 Example Client Block Diagram

The “Client” example functions as a BACnet client. Three BACnet services are used in this example: *Who-Is*, *WriteProperty* and *ReadProperty*. First, the example VI discovers all BACnet devices in the network. If any is found, then the device with the first *Device\_ID* is chosen, the specified property is written, acknowledged, then the same property of the same device is read using *ReadProperty.vi*.

Pay attention, that the client and the server of this toolkit don't work locally, that means that you cannot successfully use both example VIs on the same device.

## 9. System Requirements

- LabVIEW Base, Full, or Professional Development System.
- Windows XP, vista, 7, 8, 10.
- NI Linux RT

## 10. References

1. <http://www.bacnet.org/>
2. "Standard 135-2012-- BACnet--A Data Communication Protocol for Building Automation and Control Networks (ANSI Approved)". 2012.

## 11. Support Information

For technical support, please, contact Ovak Technologies at:

Phone: + 374 10 21-97-68

Email: [support@ovaktechnologies.com](mailto:support@ovaktechnologies.com)

Web: [www.ovaktechnologies.com](http://www.ovaktechnologies.com)