BACnet Integration Guide

For Automated Logic LGR, ME-LGR, or ME812u-LGR control modules

Contents

Basic Hardware/Point allowance Overview

ALC Side

ALC Control modules required for a Third party Integration	LGR25 (supports up to 25 Third Party points) LGR250 (supports up to 250 Third Party points) LGR-1000 (supports up to 1000 Third Party points) ME-LGR25 (supports up to 25 Third Party points, and has I/O points) ME-LGR200 (supports up to 200 Third Party points and has I/O points) ME812u-LGR (supports up to 200 Third Party points and has I/O points)
Read/write capability	*Can read from and write to the third-party equipment*

Third Party Side

Supported equipment	Any device that supports the BACnet protocol
Supported media	IP, (Bacnet/IP) Ethernet, (Bacnet/Ethernet) ARCNET, (Bacnet/Arcnet) MS/TP (9600, 19200, 38400, or 76800 kbps) (Bacnet/MSTP) PTP, (Bacnet/PTP)

• The point allowance of a control module that provides third-party points applies only to itself. For example, if you purchase an LGR1000 that provides 1000 third-party points and you download control programs that use 500 third-party Network I/O points, you cannot apply the unused 500 points to a different piece of hardware.

EXAMPLES of third-party BACnet integration hardware

With this hardware...	Use...
LGR25 SE6104 UNI M4106	• The LGR25 for the first 25 third-party points. • You can add third-party points with: Additional control modules that provide third-party points

With this hardware...	Use...
ME-LGR200	• The ME-LGR200 for the first 200 third-party points. • You can add third-party points with additional control modules that provide third-party points.
LGR1000	• The LGR1000 for the first 1000 third-party points. • You can add third-party points with: ○ Additional control modules that provide third-party points.

Before You Begin Checklist

- ✓ The ALC control module's *Technical Instructions*. (Attached CD has documents included)

- ✓ The system's SiteBuilder database.

- ✓ The third-party device's:
  - BACnet network number
  - MAC address
  - Device Instance
  - Points List including BACnet object ID's (object type plus point instance)
  - *If this information is not supplied by the third-party vendor, use WebCTRL's auto discovery tool. See pg27*

- ✓ The baud rate of the third-party device if running MS/TP

- ✓ Verification that all communication settings have been set on the third-party device

- ✓ Experience creating control programs with EIKON LogicBuilder

- ✓ Experience installing, wiring, setting up, and downloading memory to the ALC control module

The Prefered Method of Integration

This highlights the preferred method of integration to be used when interfacing to a third party device via BACnet

Method :

You can use a Network I/O microblock to read from or write to any single property in a third-party BACnet object.

You must use Network I/O microblocks if any of the following apply:

- An ALC control program must read from or write to third-party BACnet points for automatic control

- You want to trend values from a third-party device that does not support BACnet trends

- You want to display third-party values that require unit conversion or other math processing

****NOTE** Each Network I/O microblock that is addressed to a third-party object property uses one third-party point provided by a control module (see: Basic Hardware/Point allowance Overview beginning on pg 3)

Create a control program that will be your third party interface in EIKON LogicBuilder

- A control program must be created in EIKON LogicBuilder.
- This control program is graphical user interface that is to be used to monitor and control the Third party device
- This control program will contain all required control that is to be used on you WebCTRL system
- This control program must be downloaded into a third party capable Automated Logic control module (see: Basic Hardware/Point allowance Overview beginning on pg 3)
- When you create your control program, use a Network I/O microblock for each third-party point.
- Each third party point must follow proper formatting to allow communication between WebCTRL and the third party device.
- Although most parameters within microblocks are editable within WebCTRL, the control program itself can only be edited in EIKON LogicBuilder.

Formatting a Bacnet Address

Use the following information to format a valid address in each microblock that you use to read or write a third-party point.

CAUTION!!!

- When integrating third-party devices into a WebCTRL system, most communication problems are caused by incorrect data or typing errors in the microblock's Address field.

Typical BACnet Address (URL)

Items:

(third party identifier) Required for communication

(Device) Required for communication

(Object) Required for communication

(Property) Optional

(Priority) Optional

Typical BACnet Address (URL)

A This is the third party identifier, some examples are:

bacnet:///, Modbus:///, N2:///

There are several others but for this manual we will only be using: **bacnet://**

1 **Device:**

Use one of the following:	EXAMPLES
Device instance number	bacnet://2010/…
BACnet device name	bacnet://MyDevice/…
Network number: MAC address (of third-party device)	bacnet://1234:35/… bacnet://1234:0x23/…
The word "this" if a network point requests a value from another control program in the same ALC control module. Avoids network traffic. Requires v2.05 or later control module driver.	bacnet://this/

Typical BACnet Address (URL)

Object:

Use one of the following:	EXAMPLES
Object type: Instance number (See NOTES below) BACnet object name	bacnet://…/ai:2 or bacnet://…/MyObject

NOTES on Object above:

For object type, you may type the abbreviation (not case sensitive), the full name, or the object type number. Some standard BACnet object type numbers are listed below. See the BACnet standard for a complete list. For proprietary BACnet objects, see the object's manufacturer

USE:	Or:	Or:
ai	analog-input	0
ao	analog-output	1
av	analog-value	2
bi	binary-input	3
bo	binary-output	4
bv	binary-value	5
dev	device	8
msi	multistate-input	13
mso	multistate-output	14
msv	multistate-value	19

• Every object in a control module has a unique instance number, regardless of its control program.

Typical BACnet Address (URL)

Property (optional):

If you want to read or write a property other than present_value, type one of the following:	EXAMPLES
BACnet property identifier BACnet property identifier # Property identifier (with index)	bacnet://…/…/cov_increment or bacnet://…/…/22 or bacnet://…/…/priority-array(12)
Property identifier # (with index)	bacnet://…/…/87(12)

Note: TIP For standard BACnet objects, see the BACnet standard for property identifiers and property identifier numbers. For proprietary BACnet objects, see the object's manufacturer.

Typical BACnet Address (URL)

4 Priority (optional):

By default(if no priority is entered into the bacnet url string) the priority will be 16. If you want to write at a priority other than 16, type @ followed by a priority number between 1 and 16 (bacnet://…/…/…@9). See below table for specific priority levels

NOTE 1: Priority levels 1 and 2 are reserved for manual and automatic life safety commands.

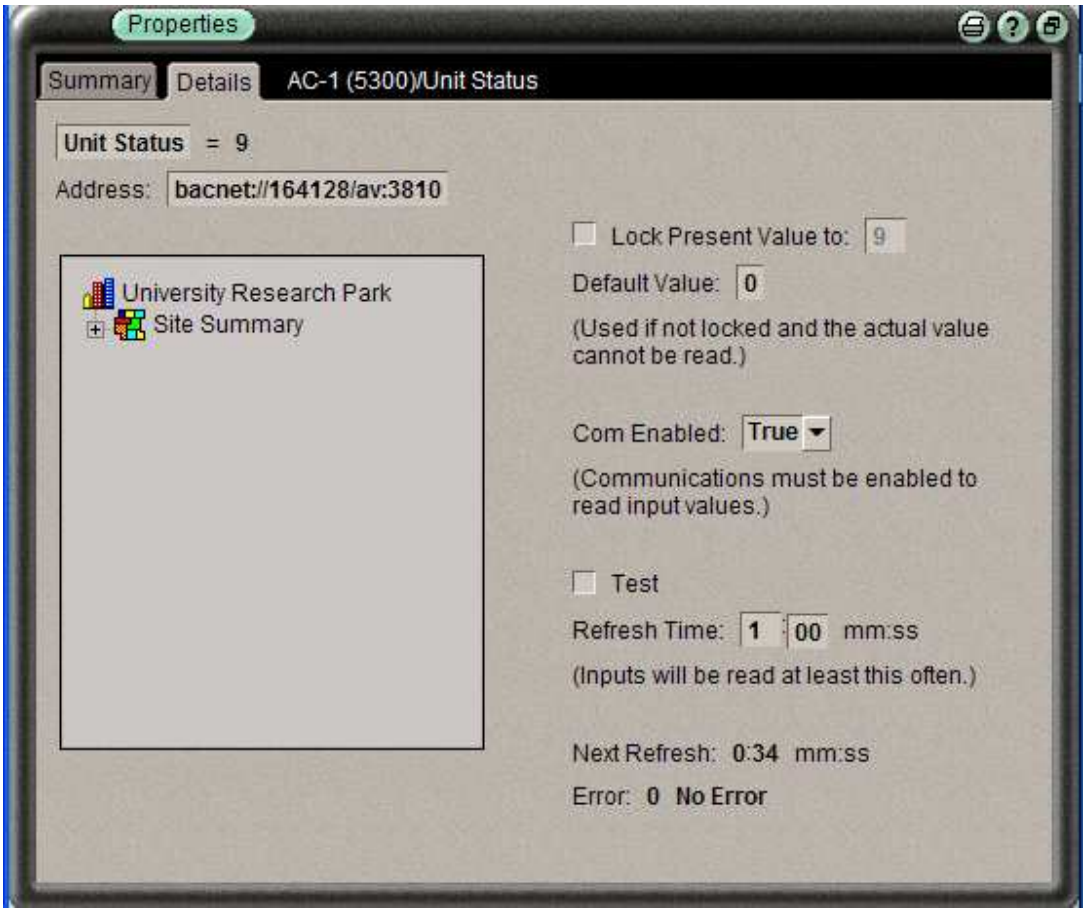
Prioity Level	Priority Type
1	Manual Life Safety (Highest Priority)
2	Automatic Life Safety
3	Unspecified
4	Unspecified
5	Critical Equipment Control
6	Minimum On/Off
7	Unspecified
8	Manual Operator
9	Unspecified
10	Unspecified
11	Unspecified
12	Unspecified
13	Unspecified
14	Unspecified
15	Unspecified
16	Lowest (default)

Typical BACnet Address (URL)

Examples of Valid BACnet addresses:

- bacnet://MyDevice/ai:2
- bacnet://1234:0x23/analog-input:2/priority-array(12)@8
- bacnet://2499:0x00E0C90047CA/bi:3
- bacnet://2436:192.168.47.36:47806/0:2
- bacnet://192.168.168.5/ai:s231
- bacnet://164128/av:3810

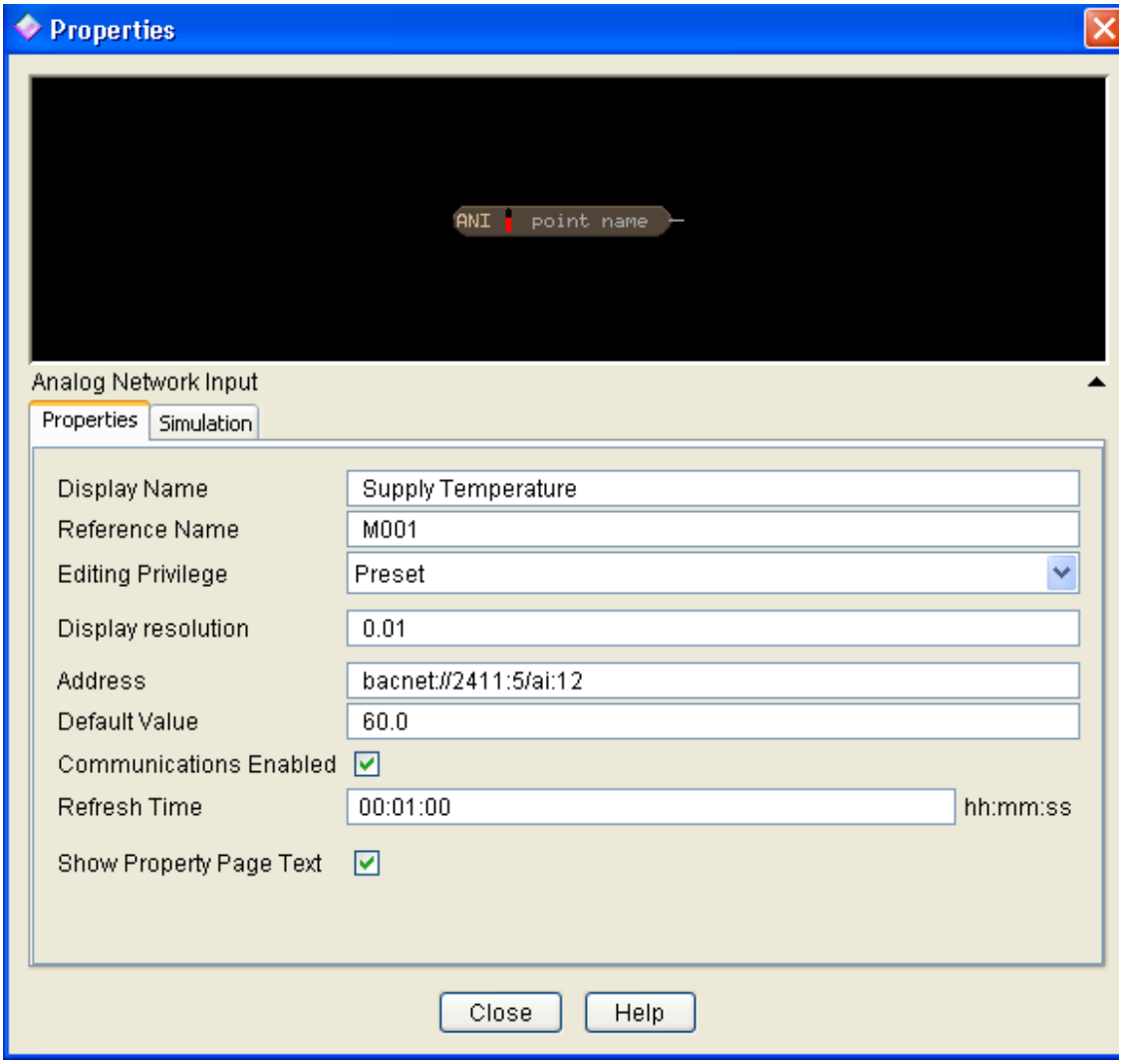
Below is an example of an actual properly bound BACnet point



Editing a Microblock Address:

You can edit a microblock address (BACnet URL Address string) in any of the following places.

- In EIKON LogicBuilder in the Property Editor(double click on a microblock In EIKON LogicBuilder to bring up the Property Editor)(see image below)
- In WebCTRL on the microblock's **Properties** page > **Details** tab
- In WebCTRL on the control program's Properties page > **Network Points** tab (see above image for actual bound network point)



Setting up Network Inputs

WebCTRL uses two different methods to communicate to third part network points; **Polling** and **BACnet COV (**change of value)

If a Network Input or Total Analog microblock's Address field references a BACnet object property, the microblock reads the property's value using one of the following methods.

• **Polling**—The microblock reads the property at the Refresh Time interval using the BACnet ReadProperty or ReadPropertyMultiple service (see "Method 1: Polling" below).

• **BACnet COV** (Change of Value) subscription—The microblock subscribes with the target BACnet object. An analog target notifies the microblock if the target's value changes by more than the target's BACnet COV_Increment. A binary target notifies the microblock when it changes state (see "Method 2: BACnet COV subscriptions" below).

Method 1:

Polling Benefits

- Allows rapid detection of a dead device or of network problems
- Does not require additional memory

Drawbacks

- Generates unnecessary network traffic if a value does not change frequently
- Misses value changes that occur between pollings
- Can overwhelm the target's control module if many microblocks request the same property value (such as outside air temperature). The BACnet object must send the value to each microblock that polls for that data.

To set up — Set the microblock's Refresh Time to 30 seconds or less.

NOTE: The ALC microblock will not poll at a Refresh Time interval smaller than 1 second.

Method 2:

BACnet COV subscriptions Benefits	• Can decrease network traffic by preventing unnecessary updates if the target's COV_Increment is set appropriately. See step 2 in "To set up" below.
Drawbacks	• Can generate excessive network traffic if the target's COV_Increment property is too small. See step 2 in "To set up" below. • Can delay detection of a dead device or of network problems

To Set Up:

1) Set the microblock's Refresh Time to 31 seconds or more.

2) If the microblock's Address field references an analog property, set the target's COV_Increment property to the smallest amount by which the value must change for the target to notify its subscribers. The optimal COV_Increment is large enough to prevent unnecessary updates but small enough to be useful to the control program(s) receiving the updates.

NOTE: If COV subscription fails, the microblock reads the value at the Refresh Time interval using the BACnet ReadProperty or ReadPropertyMultiple service. See "Method 1: Polling" above.

COV subscription details

When an input (Network Input or Total Analog microblock) subscribes with a BACnet target (object property), the input sets a 21-minute subscription Lifetime in the target. The target responds with a COV notification that includes the target's value and time remaining from the original subscription Lifetime (TimeRemaining). The input resubscribes with the target every 10 minutes to

keep the target's BACnet subscription service active. The Next Subscription field on the input's Properties page shows the time remaining until the input's next subscription.
The target also sends a COV notification that includes the target's value and subscription Lifetime TimeRemaining when the target's value changes by more than the target's COV_Increment.
If the ALC target has one subscriber, the target sends COV notifications directly to that subscriber. If the ALC target has more than one subscriber, it broadcasts its COV notifications to optimize network traffic. A third-party subscriber can participate in this broadcast scheme by subscribing for Unconfirmed COV notifications with a Process ID of 0. Otherwise, the ALC target maintains and responds to the third-party subscription separately with its own Lifetime timer.
The ALC input compares the TimeRemaining value in each COV notification broadcast the target sends to its (Next Subscription time + 11) to determine whether another input has subscribed since it did. If another input has subscribed more recently, the input adds 10 minutes to its Next Subscription time. This allows the COV Subscription request from the last subscribing input to keep the subscription service active for all subscribers to the same data.

See examples on next page

EXAMPLES and explanation of COVs within WebCTRL

Elapsed time (minutes)	**Action**	**Target** Lifetime TimeRemaining (minutes)	**Input 1** Next Subscription (minutes)	**Input 2** Next Subscription (minutes)
0	Input 1 subscribes to target	21 (Input 1)	10	
0	Target broadcasts COV notification because Input 1 subscribed	21	21 ≤ 10 + 11, so keep current value of 10	
2	Input 2 subscribes to target	21 (Input 2)	10 - 2 = 8	10
0	Target broadcasts COV notification because Input 2 subscribed	21	21 > 8 + 11, so add 10 to current value of 8 8 + 10 = 18	21 ≤ 10 + 11, so keep current value of 10
3		21 - 3 = 18	18 - 3 = 15	10 - 3 = 7
0	Target broadcasts COV notification because value changed	18	18 ≤ 15 + 11 so keep current value of 15	18 ≤ 7 + 11 so keep current value of 7
7		18 - 7 = 11	15 - 7 = 8	7 - 7 = 0 resubscribe
0	Input 2 resubscribes	21 (Input 2)	8	10
0	Target broadcasts COV notification because Input 2 subscribed	21	21 > 8 + 11, so add 10 to current value 8 + 10 = 18	21 ≤ 10 + 11, so keep current value of 10

Input 2 keeps the subscription service active at the target with a minimum of network traffic.

NOTE: If an input receives COV notification with a target time remaining < 11, which could happen if the last subscribing input loses communication with the target, the input re-subscribes immediately.

COV notification rate:

COV notifications from a BACnet object property are controlled by that property's BACnet COV_Increment. When the absolute value of the difference between the property's Present_Value and the value sent in the last COV notification is greater than the COV_Increment, the object broadcasts a COV notification. For ALC control modules, the rate of notifications is further limited by two internal processes:

1.) The control program's execution rate determines how often the check against COV_Increment is performed.

2.) The control module's pending COV Notification task has built-in delays to prevent COV notifications from consuming the control module's CPU processing time.

The built-in delays are as follows:

If more than 15 COV notifications are pending delivery, the control module inserts a 50 millisecond delay after each set of 15 notifications. Once the entire list of pending notifications is serviced, the control module inserts another 50 millisecond delay. This results in a maximum COV notification rate of 300 COV notifications per second per ALC control module.

To Speed up detection of an unresponsive BACnet Device

If a BACnet object's device loses network communication, a network input reading the object's value does not detect the failure until :

• the network input's next subscription (up to 10 minutes) if using BACnet COV subscription,

or

• the **Refresh Time** expires, if polling

You can use a small **Refresh Time** to poll more often, but this can generate unnecessary network traffic under normal conditions.

WARNING: Exxcessive network traffic can slowdown a system and can create realtime control issues especially if network points are directly linked to critical control points, an example could be…a PID loop control based on a network read value compared to a network read setpoint.

To use the benefits of BACnet COV subscription, but overcome the potential delay in detection of a dead device, send a constantly changing value from the BACnet object's control program to a network input using BACnet COV subscription. If the value stops changing, the network input's control program generates an alarm.

Connect the third party device to the ALC control module

Note: there is supplemental material(both documents and video) about bacnet wiring and communication standards on the included CD on the back cover of this document.

To wire a BACnet/ARC156 device:

1) Turn off the control module's power.

2) Check the communications wiring for shorts and grounds.

3) Connect the third-party device's communications wiring to the control module's screw terminals labeled Net +, Net -, and Shield.

4) Use the same polarity throughout the network segment.

5) If connecting to an ME812u-LGR, set the BACnet Mode jumper to ARC156.

6) Turn on the control module's power.

To wire a BACnet MS/TP device:

1) Turn off the control module's power.

2) Check the communications wiring for shorts and grounds.

3) Connect the third-party device's communications wiring to the control module (see below table)

For a...	Connect wiring to...	Set...
LGR25 LGR250 LGR-1000 ME-LGR25 ME-LGR200	**Port S1** terminals labeled **Net +** and **Net -**	• **MSTP on S1** DIP switch to **Enable** (**ON**) • Port S1 jumpers to **EIA-485** and **485-2w**
ME812u-LGR	**BACnet** port terminals labeled **Net +**, **Net -**, and **Shield**	**BACnet Mode** jumper to **MSTP**

4) NOTE Use the same polarity throughout the network segment.

5) Turn on the control module's power.

6) To change the port's baud rate, see "To set a port's baud rate using HyperTerminal" in the control module's *Technical Instructions*.

NOTE: Use the same baud rate for all devices on the network segment.

To wire a BACnet PTP device:

1) Turn off the control module's power.

2) Check the communications wiring for shorts and grounds.

3) Connect the third-party device's communications wiring to the control module. (see below table)

For a...	Connect wiring to...	Set...
LGR25 LGR250 LGR-1000 ME-LGR25 ME-LGR200	**Port S2** terminals labeled **Tx**, **Rx**, and **Signal Ground**	• **PTP on S2** DIP switch to **Enable** (**ON**) • Port S2 jumper to **EIA-232**
ME812u-LGR	**Port S1** terminals labeled **Tx**, **Rx**, and **Signal Ground**	• **PTP on S1** DIP switch to **Enable** (**ON**) • **Enhanced Access Port** DIP switch to **Off** • Port S1 jumpers to **EIA-232** and **Full Duplex**

NOTES :
○ Jumper the DTR and DCD terminals.
○ Use the same polarity throughout the network segment.

4) Turn on the control module's power.

5) To change the port's baud rate, see "To set a port's baud rate using HyperTerminal" in the control module's *Technical Instructions*

NOTE :Use the same baud rate for all devices on the network segment

Download the BACnet driver

Get the latest BACnet driver

If you do not have the latest version of your ALC control module's driver, follow the steps below.

NOTE If your ALC control module is using a driver for another protocol, that driver will also allow you to integrate with BACnet points.

1) Get the latest module driver drv_melgr_vanilla_<latest version>.driver from the *an ALC Dealer*

2) Download the driver, saving it in WebCTRLx.x\webroot\<system_name>\drivers.

3) On SiteBuilder's Network tree, open the Device Properties dialog box for the ALC control module.

4) Select the ALC control module in the Device Definition drop-down list, then click OK.

5) On the Network tree, expand the control module, then double click on Driver

6) Select drv_melgr_vanilla_<latest version> in the driver definition drop down list, then click OK.

Download the driver and control programs :

1) In SiteBuilder, assign the equipment to the control module by dragging the equipment from the Geographic tree and dropping it on the control module in the Network tree.
2) In WebCTRL, download memory to the ALC control module.

Verify the control module is set up correctly

To verify correct setup of network points:

1) On WebCTRL's GEO tree, select the control program for the ALC control module.

2) Select the Properties page > Network Points tab.

If...	Then...
You see the point value you expect with no errors in the **Error** column	You have successfully established communication with the third-party device.
All points show question marks instead of values	WebCTRL is not communicating with the ALC control module or the control program. Troubleshoot the control module's communications. See the control module's *Technical Instructions*.
Some points show question marks instead of values	You may have exceeded the third-party points available in the control module. If so, do one of the following: • Use a control module that provides more third-party points. • Split the points between two control programs used in separate control modules. To determine the number of third-party points used in a control module: On WebCTRL's **NET** tree, click on the controller's **Driver**, then scroll to the bottom of the page. **Number of integration points requested** and **Number of integration points active** show how many third-party Network I/O microblocks the control module is using. These counts will differ if you exceed the control module's integration point limit. For example, if your LGR25's control program includes 27 third-party points, your **Integration points requested** will be 27 and your **Integration points active** will be 25.

If...	Then...
The point name is red	Look in the **Error** column for one of the following error codes and descriptions. • **1 - Communications Disabled for this Microblock** Enable the microblock's **Communications Enabled** field on WebCTRL's **Network Points** tab, on the microblock's **Properties** page > **Details** tab, or in EIKON LogicBuilder. • **3 - Address Error - Unknown Protocol Specified** Select the correct port on the module driver page in WebCTRL, set the DIP switches correctly on the control module, or correct the **Address** field in the microblock.
A value is incorrect	Verify that: • The **Address** in the microblock is correct. • The retrieved value is scaled properly, if necessary. For example, scaled from Celsius to Fahrenheit. Refer to the third-party manufacturer's documentation or the control module's *Technical Instructions* for scaling information.

If the above solutions do not resolve the problem, gather the following information for technical support prior to calling your support representative:

- A screenshot of the driver Properties, IP Addressing, and Protocol pages
- A screenshot of the Properties page > Network Points tab showing addresses and errors
- All information from a Modstat copied into a text file. Right-click the Modstat, then select Select All. Press Ctrl+C to copy the information, then open Notepad and paste the information into a text file.

To discover BACnet networks, devices and objects using WebCTRL's built in Discovery tool

WebCTRL's Discovery tool locates all accessible BACnet networks, BACnet devices, and BACnet objects (including devices in your WebCTRL system) on a BACnet network. The information gathered in this process is typically used to incorporate third-party BACnet devices and their BACnet objects into the WebCTRL database.

To use the Discovery tool:

1) On the WebCTRL **CFG** tree, select **System Setting**s.

2) On the **Communications** tab, clear the **Use Static BACnet Bindings** checkbox.

3) On the WebCTRL **CFG** tree, select **Connections**.

4) On the **Configure** tab, enter or verify the server's IP Address and Subnet Mask for the BACnet/IP connection.

5) Restart the connection or the WebCTRL server.

6) On the **NET** tree, select the system level item.

7) Click **Discovery**.

8) Click **Go** to discover BACnet sites for the system. An item called **Discovered Networks** appears in the tree. When all sites are found, close the status dialog box.

9) To discover BACnet networks, select **Discovered Networks**, then click Go. A list of all BACnet networks appears in the **NET** tree. When all networks are found, close the status dialog box.

TIP: Use the *commstat* manual command to determine which device routes to each network.

10) To discover BACnet devices on a network, select the network on the **NET** tree, then click **Go**. Click the plus sign beside an item to expand the list of devices. When all devices are found, close the status dialog box.

11) To discover BACnet objects on a device, select the device in the **NET** tree, then click **Go**. A list of all BACnet objects in this device appears on the **NET** tree. When all objects are found, close the status dialog box.

TIP: Make sure you are discovering objects in the correct device. It may take some time to discover objects in devices with more than 100 objects.

12) Open SiteBuilder. If SiteBuilder was open during discovery, close, then reopen SiteBuilder to view the discovered items under the **Discovered Site**. Use the information you discovered to establish communication with the desired third-party objects using Network I/O or Display microblocks, then delete Discovered Site.

13) In WebCTRL, on the **Communications** tab, select the **Use Static BACnet Bindings checkbox**.

14) Restart the connection or the WebCTRL server.

NOTES :

- Some third-party BACnet devices may not be discovered because they do not support the BACnet methods required for auto discovery.
- If the discovery process returns ambiguous information, such as multiple points with similar names, contact the third-party manufacturer's representative for clarification.

End Of Document