EquipmentBuilder v8.0 Help

 Verify that you have the most current version of this document from **www.hvacpartners.com** or **www.accounts.ivusystems.com** or your local Carrier office.

Important changes are listed in **Document revision history** at the end of this document.

Contents

What is EquipmentBuilder?

You can generate files in EquipmentBuilder for both i-Vu® Open and CCN controllers.

i-Vu® Open controllers

You can:

- Create control programs for programmable controllers from a library of pre-defined applications. These include specific equipment programs (Single Zone AHU, Water Source Heat Pump, etc.), as well as, a selection of Carrier standard algorithms.
- Create an associated graphic, in most instances, which can then be edited in ViewBuilder, or uploaded to your equipment in the i-Vu®/Field Assistant interface. You can also create your own view using ViewBuilder, if one is not automatically generated.
- Change default setpoints on the **Sequence of Operation** tab.
- Generate the following files, based on your selected options:
    - Control program (.equipment)…editable in Snap
    - Graphic (.view)…editable in ViewBuilder
    - BAC*view*® file (.bacview), if applicable to the equipment….editable in ViewBuilder
    - Touch file (.touch)…editable in ViewBuilder
    - Sequence of operation (.odt)….editable in OpenOffice or Microsoft® Word
    - .driver

CCN equipment

You can:

- Build custom equipment files for your equipment, such as Universal Controllers, Comfort Controllers, Terminal System Managers, and EMEA Hydronic System Managers
- Create an associated graphic, in most instances, which can be edited in ViewBuilder or uploaded to your equipment in the i-Vu®/Field Assistant interface. You can also create your own view using ViewBuilder, if one is not automatically generated.
- Create **CCN Values Only, Read Integration**, or **Link Integration** custom equipments for UC/CC that you tailor for your specific application by selecting type and points. Once the equipment files are created and uploaded to i-Vu®/Field Assistant, these points can be assigned to custom graphic elements, which are created in ViewBuilder.
- Create some pre-engineered, stand-alone equipment applications, with view, for:
    - **Consumable Reports for Energy Meters** (Electric, Gas, Water, Steam, Generic)
    - **Non-linear flow meters**
    - **Energy (BTU) consumption**
    - **Equipment Runtime**.

NOTE You can also add these options to **CCN Values Only** and **Link Integration** custom equipments, which only require mapping of the points to be monitored.

- Add one meter and one **Runtime** to equipment.
- Add an application for **CCN Vertical Pack unit**. This is a CC with special software and BEST++. While this controller will discover automatically as a Vertical Pack, none of the possible selected options can be auto-discovered at this time, so only a base view will be generated.
- Generate the following files, based on your selected options:
    - Control program (.equipment)…editable in Snap
    - Graphic (.view)…editable in ViewBuilder

i-Vu® programs

Create or edit equipment control programs for i-Vu® Open, i-Vu® XT, and TruVu™ programmable controllers

EquipmentBuilder allows you to choose your equipment file from a list of currently available equipment products from the factory. If the factory equipment is not available in EquipmentBuilder, you must build a control program in Snap. See Snap Help for detailed information on using Snap.

After creating a control program, download it to the appropriate programmable controller in the i-Vu® or Field Assistant application.

To create your control program in EquipmentBuilder

1 Start EquipmentBuilder. (Windows **Start** > **All Programs** > **i-Vu Tools x.x**)

2 Click **Create Equipment** and then click **Next**.

3 Select the .sal library from the **Library:** drop-down list.

4 Select the equipment type from the list and click **Next**.

5 In the **Equipment Name** field, edit the name, if desired.

NOTE Your name must not exceed 21 characters.

6 Select options on the **Summary** tab and, if applicable, edit setpoints on the **Sequence** tab.

NOTE The **Points** and **Sequence** tabs change based on your choices on the **Summary** tab.

7 Click **Next**.

8 Check the files you wish to generate from the list.

9 Check to select saving your control program files to a folder of your choice or a folder linked to a system.

10 Click **Next**.

To edit an existing control program in EquipmentBuilder

1 Start **EquipmentBuilder**. (Windows **Start** > **All Programs** > **i-Vu Tools x.x**.

2 Click **Open Equipment** and browse to webroot/<system>/programs folder > .equipment file that you want to edit and click **Open**.

3 In the **Equipment Name** field, edit the name, if desired.

NOTE Your name must not exceed 21 characters.

4 Select options on the **Summary** tab and ,if applicable, edit setpoints on the **Sequence** tab.

NOTE The **Points** and **Sequence** tabs change based on your choices on the **Summary** tab.

5 Click **Next**.

6 Check the files you wish to generate from the list.

7 Check to select saving your control program files to a folder of your choice or a folder linked to a system.

8 Click **Next**.

See i-Vu® Help for instructions on how to apply your new or edited control programs to controllers in your system.

Programmable controller application restrictions

A programmable controller supports multiple applications in the same controller, but with the following restrictions:

- Certain applications need fixed BACnet addresses, channel numbers, and other programming data. Do not place more than one of these applications in the same controller.
- Certain very complex applications (especially PIC-based), have so many fixed addresses that you must not use more than one type of application.
- Certain applications are generally unrestricted, but contain restricted options. The **Unit Heater**, for example, is unrestricted as long as you do not select the **Hot Water Source Linkage** option. Only one application with this option is allowed per controller. This restriction is clearly identified in EquipmentBuilder configuration options.

Restricted application codes

- **R - Restricted** - Do not mix with other Restricted or Special Type applications in the same controller.
- **CR** - **Conditionally Restricted** - The basic application is general, but certain options make this a **Restricted** application.
- **S-XX** - **Special Type** – The application contains a special type of addressing, which usually serves as a target for single point linkage. Use **type code** to identify the type of addressing, such as **S-OA** for outside air addresses, or **S-EM** for electric meter addresses. These can be mixed with other applications, including a different **Special Type** application, but only one copy of any specific **Special Type** application can reside in a Universal Controller.

  For example, an **S-OA** and a **S-EM** can both be used within the same Universal Controller, but do not use two **S-EM** applications.

Applications created in Snap can be used freely with factory-supplied EquipmentBuilder applications.

CCN programs

Create CCN custom equipment files

1. Click **Create Equipment** to start a new custom equipment file. Click **Open Equipment** to edit an existing equipment file.
2. Select equipment type:
   - **CCN Values Only** - to read and write values on the CCN network and to display those points on a graphic
   - **Integration Values Only -** to read and write values from the third party network and to display those points on a graphic
   - **CCN Link Integration -** to share values from the third party network with the application on the CCN network and to display those points on a graphic
3. Click **Next**.
4. Type a name for the custom equipment (i.e., Hot Water System).

   NOTE The name must not exceed 21 characters.
5. Check **English** or **Metric** units.
6. Select appropriate engineering options for your application (**Schedule and Setpoint, Runtime,** etc..).
7. Click **Accept**.

Add Elements

1 Add **Elements** to your application.

NOTE Elements are a collection of input/output points that perform a specific operation. The input/output point that is reading or writing to the network is called an integration point. Integration points may be used in conjunction with CCN points to share data between the network and the CCN network.

The available **Elements** that you can add to your custom equipment in EquipmentBuilder are:

Point type	Used for
Read CCN Point	Reading an analog or binary value from the CCN network In: CCN Values, Link Integration
Carrier Text Point	Reading Text value from a CCN device In: CCN Values, Link Integration
Setpoint Write	Allows CCN setpoint value to be “edited” directly from graphic In: CCN Values, Link Integration
Link Integration Point to CCN Point	Reading an analog or binary value from the third party device and then writing that value to the CCN network In: Link Integration
Link CCN Passive Point to Integration Point	Exposing an analog or binary value to the CCN network so that it can be written to the third party network In: Link Integration
Read Integration Point	Reading an analog or binary value from the third party device In: Read Integration, Link Integration
Link CCN Point to Integration Point	Reading an analog or binary value from a CCN device and then writing it to the third party network In: Link Integration
Link Integration Point to CCN Passive Point	Reading an analog or binary value from the third party device and then exposing that value to the CCN network In: Link Integration
Link BACnet variable to CCN Point	Allows an analog or binary value from BACnet to write that value to the CCN network In: Link Integration

2 As you add **Elements**, enter the requested information for the integration or CCN points:

- **Display Text -** the name of the point as it will appear in the i-Vu®/Field Assistant interface (i.e., Frequency)
- **Reference Base -** the name that will be added to each point that makes up the Element, so that all points have a unique identifier (i.e.,input_Freq, trendFreq, output_Freq) - this name must be unique (do not copy and paste)
- **Input Address** - enter the or CCN address

  Define the or CCN address string using the syntax for each point in the list, as described below:

- **Input Scaling -** enter variables

  NOTE Use Scaling when the value you are reading from the or CCN device needs to be scaled before showing the value on a graphic or trend. Scaling information can be found in the third party points list.

  Example: You have an integration point set up to read the motor temperature of a variable speed drive. The third party points list shows that this value will be given in degrees C, but you want to display it in degrees F on a graphic. Therefore, using the formula °F = 9/5(°C) + 32, the scaling/unit conversion fields for the integration point would be filled out as follows: ([value read] + 0) x 1.8 + 32
- **Output Address -** enter the or CCN address, as described below.
- **Output Scaling -** enter variables

  Continuing the example above, if you wanted the graphic to display values in °F, but then you wanted to share that value with the CCN network in °C, the scaling for the CCN Passive Point would be:

  ([value read] + 0) x1 + 0

3 Click **Next.**

4 **Equipment Name** - type a new equipment name if desired.

5 **Save Location** - browse to a location where you would like to save the new custom equipment.

6 Click **Save**.

To assign and download a custom CCN equipment file

To add a custom control program to the list of available programs:

1 In the i-Vu®/Field Assistant navigation tree, right-click the controller you want to associate the equipment or control program with.

2 Either double-click the controller in the navigation tree, or right-click and select **Configure** in the list.

3 Enter the **Display Name**.

4 To add a control program to the list of possible .equipment files in i-Vu®/Field Assistant, do one of the following:

If the control program is...	
In the **Control Program** drop-down list	a. Select the control program that you generated in EquipmentBuilder or Snap. b. Click **Accept**.
Not in the **Control Program** drop-down list	a. Click **Add New**. b. Browse to select the equipment file. c. Click **Open**. d. Click **Continue**. e. Click **Close**. f. Click **Close** again.

5 To add a graphic, do one of the following:

If the graphic is...	
In the **Views Available** list	a. Select the graphic, then click **Attach**. b. Click **Accept**.
Not in the **Views Available** list	a. Click **Add New**. b. Browse to select the view file. c. Click **Open**. d. Click **Continue**. e. Click **Close**. f. Click **Close** again.

6 When finished, select the Gateway in the navigation tree and select the **Devices** page.

7 Click **Download CCN** to finalize your changes.

For additional pieces of equipment controlled by your Universal Controller/Comfort Controller (expansion controllers)

1 In the navigation tree, select the device manager that the controller is associated with.

2 Select the **Devices** page and click **Add Control Program**.

3 Enter the **Display Name**.

4 Select the controller or Gateway that you want to associate the new equipment with. If you select **CCN Controller**, enter the Bus and Element number of the controller.

5 Do one of the following:

If the control program is...	
In the **Control Program** drop-down list	a. Select the control program that you generated in EquipmentBuilder. b. Click **Accept**.
Not in the **Control Program** drop-down list	a. Click **Add New**. b. Browse to select the view file. c. Click **Open**. d. Click **Continue**. e. Click **Close**. f. Click **Close** again.

6 Add a graphic or, if finished, select the Gateway in the navigation tree and, on the **Devices** page, click **Download CCN** to finalize your changes.

7 To add a graphic, do one of the following:

If the graphic is...	
In the **Views Available** list	a. Select the graphic. b. Click **Accept**.
Not in the **Views Available** list	a. Click **Add New**. b. Browse to select the view file. c. Click **Open**. d. Click **Continue**. e. Click **Close**. f. Click **Close** again.

8 When finished, select the Gateway in the navigation tree and, on the **Devices** page, click **Download CCN** to finalize your changes.

Format input and output addresses

To format a CCN address

There are 3 different methods for defining the CCN address strings.

1. They can be manually typed in **EquipmentBuilder**.
2. You can use **Copy table point** in i-Vu®/Field Assistant's table interface to copy CCN point information directly from a CCN table to **EquipmentBuilder**'s "ccn://" address field.
3. You can use **Map to Point** in i-Vu®/Field Assistant's table interface to map the CCN points from your custom equipment file directly to CCN table data.

   NOTE Your custom equipment must already be downloaded in the application to use this method. (Proceed to *Assign and download custom equipment in i-Vu CCN* (page 8) first).

Method 1: Type the address manually in EquipmentBuilder

1. If you are actively reading or writing a point on a CCN device, then manually type in the CCN device's address, **Table Name**, and **Point Name** that you wish to read or write.

   ccn://link/<Table Name>/<Point Name> ("link" indicates the CCN device that the custom equipment has been mapped to)

   or

   ccn://<bus, element>/<Table Name>/<Point Name>

   Examples:

   ccn://link/HWP01-32/TEMP

   ccn://0,2/HWP01-32/TEMP

2. If the point is a CCN passive point (i.e, it's just being exposed to the CCN network), then type:

   ccn://passive/<point name>, where <point name> is the name that you have chosen for this CCN point.

   Example: ccn://passive/freq

Method 2: Copy table point in i-Vu®/Field Assistant

1. Launch i-Vu®/Field Assistant.
2. Select the desired equipment in the navigation tree.
3. Expand the tables underneath that equipment.
4. Find the specific table and point that you want to read or write.
5. Click **Copy** in the table interface.
6. From the EquipmentBuilder interface, hit CTRL-V to copy the CCN address from the table to the "ccn://" address field.

Method 3: Map to point in i-Vu®/Field Assistant

1 Launch i-Vu®/Field Assistant.

2 Select the desired equipment in the navigation tree.

3 Expand the tables underneath that equipment.

4 Find the specific table and point that you want to read or write.

5 In the table interface, navigate to the **Map to Point** column.

6 From the drop-down list, select the point in the custom equipment that should be mapped.

7 Click **OK**.

To format a BACnet address

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

CAUTION
When integrating third-party devices into a i-Vu®/Field Assistant system, most communication problems are caused by incorrect data or typing errors in the microblock's **Address** field.

NOTE Numeric values in a BACnet address can be entered using decimal or hexadecimal notation. Type `0x` before a hexadecimal value.

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 Carrier controller. Avoids network traffic. Requires v2.05 or later controller driver.	`bacnet://this/...`

Object - Use one of the following:

	EXAMPLES
Object type: Instance number (See NOTES below)	`bacnet://…/ai:2`
BACnet object name	`bacnet://…/MyObject`

NOTES

- 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 controller has a unique instance number, regardless of its control program.

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://…/…/cov_increment`
BACnet property identifier #	`bacnet://…/…/22`
Property identifier (with index)	`bacnet://…/…/priority-array(12)`
Property identifier # (with index)	`bacnet://…/…/87(12)`

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.

4 **Priority** (optional) If you want to write at a priority other than 16, type @ followed by a priority number.

Number (1–16)

EXAMPLE

```
bacnet://…/…/…@9
```

NOTE Priority levels 1 and 2 are reserved for manual and automatic life safety commands. For more information on reserved priority levels see the BACnet standard.

Examples of 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
```

To read BACnet points

Polling or BACnet COV

If an integration point is an input, and its **Address** field references a BACnet object property, the integration point reads the property's value using one of the following methods.

- Polling—The integration point 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 integration point subscribes with the target BACnet object. An analog target notifies the integration point if the target's value changes by more than the target's BACnet COV_Increment. A binary target notifies the integration point 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 controller if many integration points request the same property value (such as outside air temperature). The BACnet object must send the value to each integration point that polls for that data.
To set up	• Set the integration point's **Refresh Time** to 30 seconds or less.

NOTE A Carrier integration point will not poll at a **Refresh Time** interval smaller than 1 second.

BACnet ReadProperty and ReadPropertyMultiple services

See the BACnet specification for details on the ReadProperty and ReadPropertyMultiple services.

ReadPropertyMultiple occurs if:

- two or more integration points in an application read more than one target in the same device
- the **Refresh Time** in two or more integration points expires at the same time, and the remote device supports the service.

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.
	•	Can delay detection of a dead device or of network problems
To set up	**1**	Set the integration point's **Refresh Time** to 31 seconds or more.
	2	If the integration point'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 integration point 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 integration point 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. i-Vu®/Field Assistant shows the time remaining until the input's next subscription in the **Next Subscription** field on the input's i-Vu®/Field Assistant **Properties** page.

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 a Carrier target has one subscriber, the target sends COV notifications directly to that subscriber. If an Carrier 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 Carrier target maintains and responds to the third-party subscription separately with its own Lifetime timer.

A Carrier 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.

EXAMPLE

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 TimeRemaining < 11, which could happen if the last subscribing input loses communication with the target, the input resubscribes 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 Carrier controllers, the rate of notifications is further limited by two internal processes.

1. The custom equipment's execution rate determines how often the check against COV_Increment is performed.
2. The 's pending COV Notification task has built-in delays to prevent COV notifications from consuming the controller's CPU processing time.

   The built-in delays are as follows:

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

To speed detection of a dead device

If a BACnet object's device loses network communication, an integration point reading the object's value does not detect the failure until

- The integration point's next subscription (up to 10 minutes) if using BACnet COV subscription
- 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.

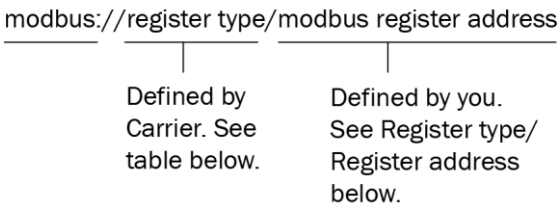
To format a Modbus address

Formatting a Modbus EIA-232/485 Slave address

Use the information below to format a valid address in each microblock that you want to expose to the Modbus master.

CAUTION
When integrating third-party devices into a i-Vu®/Field Assistant system, most communication problems are caused by incorrect data or typing errors in the microblock's **Address** field.

Address format:



Example: modbus://UINT/40128

Register type/Register address

When the Carrier controller is a slave device, you must expose its points so that they can be read or written to by a third-party master device.

For a third-party device to...	this kind of exposed value in the Carrier controller...	use this microblock...	with this register type...	and a Modbus register address in this range...
Write	0 to 65,535 Unsigned 16-bit integer Holding register (**4**XXXXX)	ANI	Uint (not Unit)	**4**00001–**4**65535
	–32,768 to +32,767 Signed 16-bit integer Holding register (**4**XXXXX)	ANI	Sint	**4**00001–**4**65535
	Value with decimal point Holding register (**4**XXXXX)	ANI	Float	**4**00001–**4**65535
	0 or 1 Coil Discrete (binary) output	BNI	Do	1–65535 [3]
Read	0 to 65,535 Unsigned 16-bit integer Input register (**3**XXXXX)	ANO	Uint (not Unit)	**3**00001–**3**65535
	0 to 65,535 Unsigned 16-bit integer Holding register (**4**XXXXX)	ANO	Uint (not Unit)	**4**00001–**4**65535
	–32,768 to +32,767 Signed 16-bit integer Input register (**3**XXXXX)	ANO	Sint	**3**00001–**3**65535
	–32,768 to +32,767 Signed 16-bit integer Holding register (**4**XXXXX)	ANO	Sint	**4**00001–**4**65535
	Value with decimal point Input register (**3**XXXXX)	ANO	Float	**3**00001–**3**65535
	Value with decimal point Holding register (**4**XXXXX)	ANO	Float	**4**00001–**4**65535
	0 or 1 Coil Discrete (binary) output	BNO	Do	1–65535 [3]
	0 or 1 Discrete (binary) input (**1**XXXXX)	BNO	Di	**1**00001–**1**65535

[1] If the register addresses in the points list are 5-digit numbers in the range **#**0001-**#**9999 (where **#** = the prefix **1**, **3**, or **4**), you may use the addresses as they are, or change them to fit the formats in the table above by shifting the prefix one digit to the left. For example, if you see the address **3**8129, you can also use **3**08129.

[2] Each Float uses 2 consecutive Modbus register addresses. Use the lower number in the microblock address.

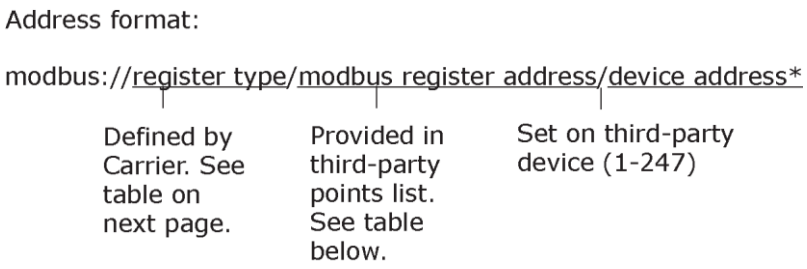
[3] Enter the number without any leading zeros. For example, enter 125, not 000125.

See Modbus Troubleshooting for more information on register addresses.

Formatting a Modbus EIA-232/485 Master address

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

CAUTION
When integrating third-party devices into a i-Vu®/Field Assistant system, most communication problems are caused by incorrect data or typing errors in the microblock's **Address** field.



Example: modbus://UINT/40128/26

NOTE To send a broadcast to all Modbus devices on the port specified by the protocol signature, use 0 for the device address in a network output microblock.

Register type/Register address

To...	this kind of value...	use this microblock...	with this register type...	and a Modbus register address in this range... [1]
Read	0 to 65,535 Unsigned 16-bit integer Input register (**3**XXXXX)	ANI	Uint (not Unit)	**3**00001–**3**65535
	0 to 65,535 Unsigned 16-bit integer Holding register (**4**XXXXX)	ANI	Uint (not Unit)	**4**00001–**4**65535
	0 to 4,294,967,296 Unsigned, 32-bit (long) integer Input register (**3**XXXXX)	ANI	Uint32	**3**00001–**3**65535
	0 to 4,294,967,296 Unsigned, 32-bit (long) integer Holding register (**4**XXXXX)	ANI	Uint32	**4**00001–**4**65535
	–32,768 to +32,767 Signed 16-bit integer Input register (**3**XXXXX)	ANI	Sint	**3**00001–**3**65535
	–32,768 to +32,767 Signed 16-bit integer Holding register (**4**XXXXX)	ANI	Sint	**4**00001–**4**65535
	-2,147,483,648 to 2,147,483,647 Signed, 32-bit (long) integer Input register (**3**XXXXX)	ANI	Sint32	**3**00001–**3**65535
	-2,147,483,648 to 2,147,483,647 Signed, 32-bit (long) integer Holding register (**4**XXXXX)	ANI	Sint32	**4**00001–**4**65535
	Value with decimal point Input register (**3**XXXXX)	ANI	Float	**3**00001–**3**65535 [2]
	Value with decimal point Holding register (**4**XXXXX)	ANI	Float	**4**00001–**4**65535 [2]
	0 or 1 Coil / Discrete (binary) output	BNI (or ANI)	Do	1–65535 [3]
	0 or 1 Discrete (binary) input (**1**XXXXX)	BNI (or ANI)	Di	**1**00001–**1**65535
	0 or 1 Input register (**3**XXXXX)	BNI	BIT*n* (where *n* is a value 0-15 defined in points list)	**3**00001–**3**65535
	0 or 1 Holding register (**4**XXXXX)	BNI	BIT*n* (where *n* is a value 0-15 defined in points list)	**4**00001–**4**65535

To...	this kind of value...	use this microblock...	with this register type...	and a Modbus register address in this range... [1]
Write	0 to 65,535 Unsigned 16-bit integer Holding register (**4**XXXXX)	ANO	Uint (not Unit)	**4**00001–**4**65535
	–32,768 to +32,767 Signed 16-bit integer Holding register (**4**XXXXX)	ANO	Sint	**4**00001–**4**65535
	Value with decimal point Holding register (**4**XXXXX)	ANO	Float	**4**00001–**4**65535 [2]
	0 or 1 Coil Discrete (binary) output	BNO (or ANO)	Do	1–65535 [3]
	0 or 1 Holding register (**4**XXXXX)	BNO (or ANO)	BIT*n* (where *n* is a value 0-15 defined in points list)	**4**00001–**4**65535

[1] The Modbus register address (sometimes called register, address, code, or parameter) must be a decimal value. If the register addresses in the points list include the letters A-F, use a scientific calculator to convert these hexadecimal values to decimal values.

If the register addresses in the points list are 5-digit numbers in the range **#**0001-**#**9999 (where **#** = the prefix **1**, **3**, or **4**), you may use the addresses as they are, or change them to fit the formats in the table above by shifting the prefix one digit to the left. For example, if you see the address **3**8129, you can also use **3**08129.

[2] Each Float uses 2 consecutive Modbus register addresses. Use the lower number in the microblock address.

[3] Enter the number without any leading zeros. For example, enter 125, not 000125.

See Modbus Troubleshooting for more information on register addresses.

Formatting a Modbus Ethernet Server address

Use the information below to format a valid address for each value that you want to expose to the Modbus client.

CAUTION

When integrating third-party devices into a i-Vu®/Field Assistant system, most communication problems are caused by incorrect data or typing errors in the microblock's **Address** field.

Address format:

mtcpip://register type/modbus register address

- register type: Defined by Carrier. See Register type/ Register address table below.
- modbus register address: Provided in third-party points list. See Register type/ Register address below.

Example: mtcpip://UINT/40128

Register type/Register address

When the Carrier controller is a server device, you must expose its points so that they can be read or written to by a third-party client device.

For a third-party device to...	this kind of exposed value in the Carrier controller...	use this microblock...	with this register type...	and a Modbus register address in this range...
Write	0 to 65,535 Unsigned 16-bit integer Holding register (**4**XXXXX)	ANI	Uint (not Unit)	**4**00001–**4**65535
	–32,768 to +32,767 Signed 16-bit integer Holding register (**4**XXXXX)	ANI	Sint	**4**00001–**4**65535
	Value with decimal point Holding register (**4**XXXXX)	ANI	Float	**4**00001–**4**65535
	0 or 1 Coil Discrete (binary) output	BNI	Do	1–65535 [3]
Read	0 to 65,535 Unsigned 16-bit integer Input register (**3**XXXXX)	ANO	Uint (not Unit)	**3**00001–**3**65535
	0 to 65,535 Unsigned 16-bit integer Holding register (**4**XXXXX)	ANO	Uint (not Unit)	**4**00001–**4**65535
	–32,768 to +32,767 Signed 16-bit integer Input register (**3**XXXXX)	ANO	Sint	**3**00001–**3**65535
	–32,768 to +32,767 Signed 16-bit integer Holding register (**4**XXXXX)	ANO	Sint	**4**00001–**4**65535
	Value with decimal point Input register (**3**XXXXX)	ANO	Float	**3**00001–**3**65535
	Value with decimal point Holding register (**4**XXXXX)	ANO	Float	**4**00001–**4**65535
	0 or 1 Coil Discrete (binary) output	BNO	Do	1–65535 [3]
	0 or 1 Discrete (binary) input (**1**XXXXX)	BNO	Di	**1**00001–**1**65535

[1] If the register addresses in the points list are 5-digit numbers in the range **#**0001-**#**9999 (where **#** = the prefix **1**, **3**, or **4**), you may use the addresses as they are, or change them to fit the formats in the table above by shifting the prefix one digit to the left. For example, if you see the address **3**8129, you can also use **3**08129.

[2] Each Float uses 2 consecutive Modbus register addresses. Use the lower number in the microblock address.

[3] Enter the number without any leading zeros. For example, enter 125, not 000125.

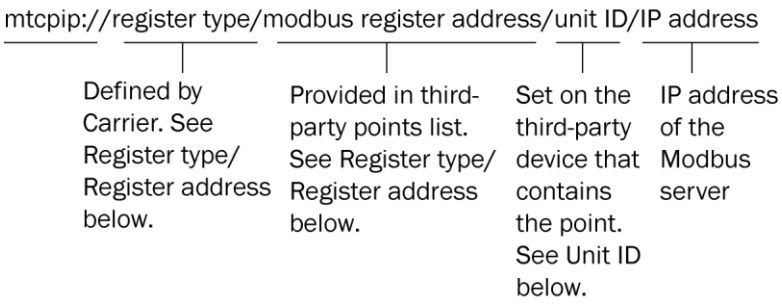
See Modbus Troubleshooting for more information on register addresses.

Formatting a Modbus Ethernet Client address

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

CAUTION
When integrating third-party devices into a i-Vu®/Field Assistant system, most communication problems are caused by incorrect data or typing errors in the microblock's **Address** field.

Address format:



Example: mtcpip://UINT/40128/3/192.168.168.1

Unit ID

If the register address is in...	the Unit ID is...
The Modbus server	0 or 255
A device on the Modbus server's serial network	1–254

Register type/Register address

To...	this kind of value...	use this microblock...	with this register type...	and a Modbus register address in this range... [1]
Read	0 to 65,535 Unsigned 16-bit integer Input register (**3**XXXXX)	ANI	Uint (not Unit)	**3**00001–**3**65535
	0 to 65,535 Unsigned 16-bit integer Holding register (**4**XXXXX)	ANI	Uint (not Unit)	**4**00001–**4**65535
	0 to 4,294,967,296 Unsigned, 32-bit (long) integer Input register (**3**XXXXX)	ANI	Uint32	**3**00001–**3**65535
	0 to 4,294,967,296 Unsigned, 32-bit (long) integer Holding register (**4**XXXXX)	ANI	Uint32	**4**00001–**4**65535
	–32,768 to +32,767 Signed 16-bit integer Input register (**3**XXXXX)	ANI	Sint	**3**00001–**3**65535
	–32,768 to +32,767 Signed 16-bit integer Holding register (**4**XXXXX)	ANI	Sint	**4**00001–**4**65535
	-2,147,483,648 to 2,147,483,647 Signed, 32-bit (long) integer Input register (**3**XXXXX)	ANI	Sint32	**3**00001–**3**65535
	-2,147,483,648 to 2,147,483,647 Signed, 32-bit (long) integer Holding register (**4**XXXXX)	ANI	Sint32	**4**00001–**4**65535
	Value with decimal point Input register (**3**XXXXX)	ANI	Float	**3**00001–**3**65535 [2]
	Value with decimal point Holding register (**4**XXXXX)	ANI	Float	**4**00001–**4**65535 [2]
	0 or 1 Coil / Discrete (binary) output	BNI (or ANI)	Do	1–65535 [3]
	0 or 1 Discrete (binary) input (**1**XXXXX)	BNI (or ANI)	Di	**1**00001–**1**65535
	0 or 1 Input register (**3**XXXXX)	BNI	BIT*n* (where *n* is a value 0-15 defined in points list)	**3**00001–**3**65535
	0 or 1 Holding register (**4**XXXXX)	BNI	BIT*n* (where *n* is a value 0-15 defined in points list)	**4**00001–**4**65535

To...	this kind of value...	use this microblock...	with this register type...	and a Modbus register address in this range... [1]
Write	0 to 65,535 Unsigned 16-bit integer Holding register (**4**XXXXX)	ANO	Uint (not Unit)	**4**00001–**4**65535
	–32,768 to +32,767 Signed 16-bit integer Holding register (**4**XXXXX)	ANO	Sint	**4**00001–**4**65535
	Value with decimal point Holding register (**4**XXXXX)	ANO	Float	**4**00001–**4**65535 [2]
	0 or 1 Coil Discrete (binary) output	BNO (or ANO)	Do	1–65535 [3]
	0 or 1 Holding register (**4**XXXXX)	BNO (or ANO)	BIT*n* (where *n* is a value 0-15 defined in points list)	**4**00001–**4**65535

[1] The Modbus register address (sometimes called register, address, code, or parameter) must be a decimal value. If the register addresses in the points list include the letters A-F, use a scientific calculator to convert these hexadecimal values to decimal values.

If the register addresses in the points list are 5-digit numbers in the range **#**0001-**#**9999 (where **#** = the prefix **1**, **3**, or **4**), you may use the addresses as they are, or change them to fit the formats in the table above by shifting the prefix one digit to the left. For example, if you see the address **3**8129, you can also use **3**08129.

[2] Each Float uses 2 consecutive Modbus register addresses. Use the lower number in the microblock address.

[3] Enter the number without any leading zeros. For example, enter 125, not 000125.

See Modbus Troubleshooting for more information on register addresses.

To format a LonWorks address

All Lonworks addresses should be assigned to "lonworks://" only when setting up address strings in EquipmentBuilder. The **LonWorks Integration Tool** is used to define these addresses.

An example of a LonWorks address:

```
lonworks://0/1/2/3ff5/108/0/0/0
```

Details:

lonworks://domain_index/subnet/node/nv_Number(Selector in HEX)/SNVT Type/NV Element/ Property/group

- group is always `0` (zero)
- Property = `1` for ANI/BNI
- Property = `0` for ANO/BNO

NOTE Carrier does not support a Domain Length of `0` (zero)

To edit an integration or CCN point address

You can edit an integration or CCN point address in the following places:

- In EquipmentBuilder
- In the i-Vu®/Field Assistant interface on the custom equipment's **Properties** page > **Control Program** tab
- In the i-Vu®/Field Assistant interface on the custom equipment's **Properties** page> **Network Points** tab
- In the **Lon Integration Tool**

To update your SAL library

The i-Vu®/Field Assistant SAL files update your i-Vu® controllers. The SAL libraries contain control programs, graphics, drivers, screen files, and other important controller data.

You can view your current SAL files in EquipmentBuilder by clicking **Help** > **Library** > **Library** tab.

NOTES

- The last digits in the SAL library name are the release date of the library.
- All of the SAL files will not necessarily have the same <date> revision.
- To ensure that your installation is running the latest software, we recommend that you check *Control Systems Support http://www.hvacpartners.com/*, *https://accounts.ivusystems.com/* for updates. Download the latest SAL files and apply them to all new installations.

To update your library, save the file in the **I-Vu_Tools_x.x\sals** folder.

NOTE Keep copies of the latest libraries in a safe place. In the event of a system restore, the updated .sal file must be reapplied.

To apply a patch

Carrier may provide product patches, which include enhancements and bug fixes.

If you receive notification from Carrier that a patch is available, follow the instructions below to update your software. Obtain the patch from *Carrier Control Systems Support Site http://www.hvacpartners.com/*, *https://accounts.ivusystems.com/*.

1 In EquipmentBuilder, select **Help** > **Apply Update**.

2 Browse to your file and click **Select**.

Document revision history

Important changes to this document are listed below. Minor changes such as typographical or formatting errors are not listed.

Date	Topic	Change description	Code*
		No changes yet	

* For internal use only