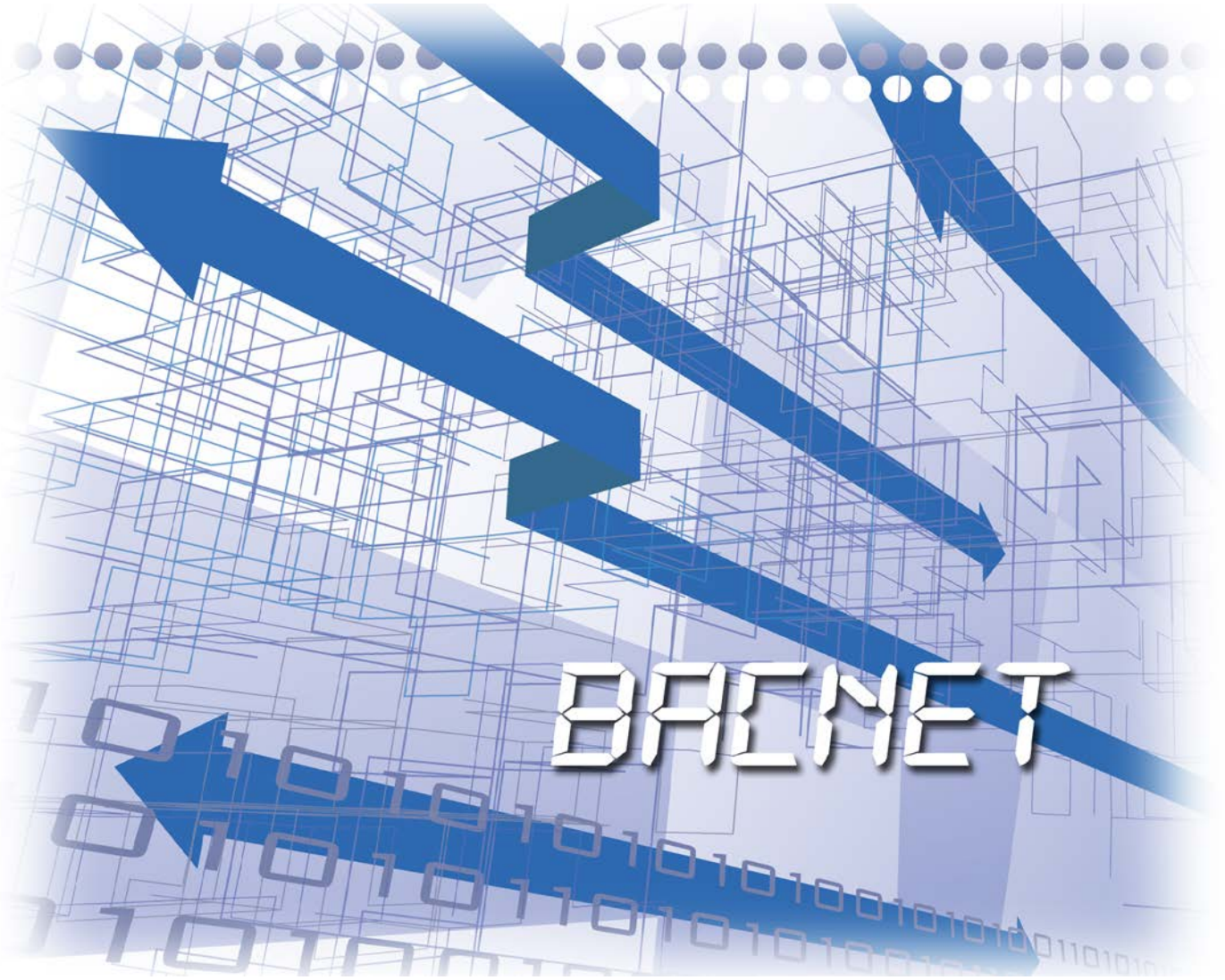
# BACnet Basics
# User's Guide





A member of the United Technologies Corporation family · Stock symbol UTX · Catalog No. 11-808-417-01 · 9/23/2013

# Table of Contents

# What is BACnet?

BACnet (**B**uilding **A**utomation and **C**ontrols **net**work) was developed by the American Society of Heating, Refrigerating, and Air-Conditioning Engineers (ASHRAE). BACnet is an ISO global standard, American national standard, a European pre-standard, and is used in more than 30 countries.

BACnet is a data communication protocol, or set of communication rules, that ASHRAE created in order to standardize communication between building automation system components. BACnet allows systems from various vendors, such as HVAC, lighting, security and fire systems, to communicate with each other by providing standardized methods for presenting, requesting, interpreting, and transporting information.

The BACnet specification defines all aspects of the BACnet protocol. This document focuses on the following:

- **Objects** – A BACnet object is a logical representation used in the BACnet protocol. It can be used to represent many different aspects of a control system. Examples are:
  - A physical device (device objects)
  - A temperature input (analog input)
  - A relay control (binary output)
- **Services** – Information exchange between objects is provided by services. Services are used to perform reads, writes, and I/O. The object that provides the service is a server and the object that requests the service is the client. Most objects can be both a server and a client, depending on the system's needs.
- **Properties** – A property contains information about an object. Objects may contain a large collection of properties, some of which may be required for the specific object type. Every object in BACnet must have at least the following three properties:
  - object_identifier
  - object_name
  - object_type

## BACnet resources

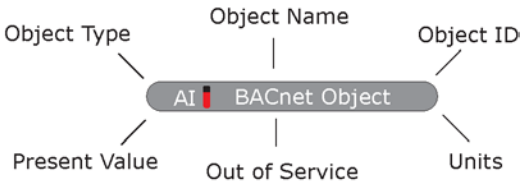
*www.bacnet.org* (*http://www.bacnet.org*)

*www.ashrae.org* (*http://www.ashrae.org*)

# BACnet objects

BACnet is an object-oriented protocol. I/O points, schedules, and devices are examples of objects. An object may be a single point or a group of points that perform a specific function. Many i-Vu® microblocks are BACnet objects.

A BACnet object packages all the information another system would need to understand and work with the object.



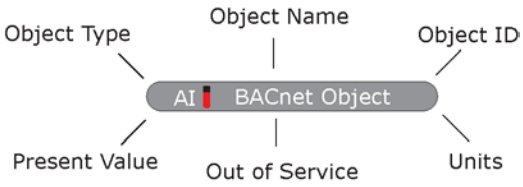

Each object has a set of properties, such as the object's name, type, and present value, which describe its behavior or govern its operation. Every property has a name and a value. The following are properties that might exist in a BACnet analog input object:

| Name | Value |
|---|---|
| Object_Name | Space Temperature |
| Object_Type | Analog Input |
| Present_Value | 74.1 |
| High_Limit | 78 |
| Low_Limit | 68 |

# BACnet objects

BACnet is an object-oriented protocol. I/O points, schedules, and devices are examples of objects. An object may be a single point or a group of points that perform a specific function. Many i-Vu® microblocks are BACnet objects.

A BACnet object packages all the information another system would need to understand and work with the object.




Each object has a set of properties, such as the object's name, type, and present value, that describe its behavior or govern its operation. Every property has a name and a value. Listed below are examples of properties that might exist in a BACnet analog input object:

| Name | Value |
|---|---|
| Object_Name | Space Temperature |
| Object_Type | Analog Input |
| Present_Value | 74.1 |
| High_Limit | 78 |
| Low_Limit | 68 |

## BACnet standard objects

BACnet defines the following object types as standard objects and defines each object's required minimum behavior (the properties it contains and services it provides).

| | | |
|---|---|---|
| Binary Input | Multi-state Input | File |
| Binary Output | Multi-state Output | Program |
| Binary Value | Multi-state Value | Schedule |
| Analog Input | Loop | Trend Log |
| Analog Output | Calendar | Group |
| Analog Value | Notification Class | Event Enrollment |
| Averaging | Command | Device |
| LifeSafetyZone | LifeSafetyPoint | |

## BACnet vendor objects

The BACnet Standard allows custom objects to be created by vendors. Carrier uses custom objects for Linkage support and other proprietary functions.

# BACnet device objects

A BACnet device object is a logical representation of a piece of control hardware, such as a controller. Device objects are one of the most important object types in the BACnet protocol, since they represent controllers and contain a list of the point objects related to the device, such as the I/O of the device. When dealing with the i-Vu® application, every entry in the navigation pane on the left side of the screen is a device object.

Device objects contain properties and perform services such as responding to read/write requests to provide information about the hardware they represent.

A BACnet device can also contain a program object, which is the control logic that the device executes.

# BACnet properties

Every object must have a collection of properties describing the object. Examples are:

- present value of an input object (space temperature)
- units of the input object (degrees Fahrenheit)
- communication status of the input object ("Good")

## BACnet optional properties

Standard objects may have optional properties and services that can be implemented if needed, but are not required. An example of an optional property is a description of the object itself, such as Space Temperature in Conference room.

## BACnet proprietary properties

Vendors may implement proprietary properties for some objects, or even proprietary object types. For other BACnet systems to read or write to the proprietary properties or objects, the vendor must document the properties or objects in the same style as BACnet standard objects and use only BACnet-defined data types to represent values.

The i-Vu® product line uses many proprietary objects and messages during normal communication. A proprietary property in the i-Vu® product line indicates whether or not it has been auto-addressed.

# BACnet routers

A BACnet router transmits BACnet messages between two BACnet networks. The networks can be different (IP to MS/TP) or the same (IP to IP). The router sends appropriate messages between the networks in both directions.

All i-Vu® routers, such as the i-Vu® Open Router and i-Vu® Open Link, can function as a BACnet router.

# BACnet networks

Different networking technologies provide varying levels of transport speed. The i-Vu® BACnet network types include:

- High speed network for high level communications (BACnet/IP)
- Economical MS/TP subnets for application-specific controllers (BACnet MS/TP)

## BACnet MS/TP

MS/TP means Master-Slave/Token-Passing.

**Master-Slave** means that any device on the MS/TP bus has to be a master or a slave. A Master can initiate communication when it has the token. Slaves can only respond to requests and can never initiate communication or hold the token.

**Token-Passing** indicates that the entire MS/TP bus is controlled by a single token that is passed around from master to master in order of MS/TP MAC address. Only the device with the token can initiate communication, but any device can respond to a communication request.

MS/TP (Master-Slave/Token-Passing) is a datalink that is unique to BACnet. The MS/TP datalink sits on top of an EIA-485 physical layer and uses common electronic components to provide a low cost implementation.

i-Vu® systems use MS/TP at the lowest level of communication in the system. BACnet MS/TP is the sole communication protocol used by i-Vu® Open controllers. Communications on an MS/TP bus solely use MAC addresses to direct messages.

Here are a few terms commonly used with BACnet MS/TP:

**Polling for Masters**: BACnet MS/TP network segments use a single token that is passed from master node to master node in ascending address order. When a network is initialized, the lowest addressed node creates a token, initiates any necessary communication, then incrementally polls the MS/TP bus until the next master node is found. This process is called polling for masters.

Once the next master node is found, the token is passed to it and the process incrementally repeats until the polling token is passed back to the original node (i.e. the node with the lowest MAC address).

**NOTE** All i-Vu® Open devices are MS/TP Masters.

**Max Masters**: This is the MAC address in which the node that is polling for masters rolls over to MAC address 0. For example, if the Max Masters setting is set to 60 (and the lowest address node is MAC: 1), the polling for masters' sequence would execute as such:

- Polling for master at 59
- Polling for master at 60
- Polling for master at 0
- Polling for master at 1
- Master found at 1

This sequence would continue until the first master device responds to the master polling. All i-Vu® Open devices are masters and would respond accordingly.

**Max Info Frames**: This is the maximum number of information frames that a node may send before it has to pass the token. For example, a setting of 10 will allow the node with the token to initiate communication 10 times before it has to retire the token to the next master node on the network segment.

**Token Count**: The number of tokens received before polling for masters' sequence is initiated. This number is set at 50 and is not configurable in i-Vu® Open systems.

### Network Overview

Below is a simplified sequence of operations for each node on the network:

1 Receive Token.

2 Initiate communication (up to the number of Max Info Frames) as needed.

3 Increment the node's token count, if the token count equals the Token Count parameter (as defined above) then initiate the polling for masters sequence, or pass the token to the next node.

## BACnet/IP

The BACnet/IP network allows BACnet messages to be communicated over IP local and wide area networks.

**NOTE** BACnet uses IP broadcasts to locate and communicate with other BACnet devices. These broadcasts are normally blocked by IP routers. The BACnet specification outlines a method of using a BACnet Broadcast Management Device (BBMD) that allows BACnet/IP communication across IP routers.

### BACnet BBMD

BACnet/IP relies heavily on IP broadcasts. But in i-Vu® systems deployed across multiple IP subnets, the IP routers may block IP broadcasts. To address this issue, BACnet Broadcast Management Devices (BBMD's) must be used.

Every subnet with a BACnet/IP router must have a BBMD configured in order for broadcasts from controllers on that subnet to reach the rest of the routers on the network.

Every BBMD has a Broadcast Distribution Table (BDT). A BDT lists all of the IP addresses of other BBMD's on the network. When a BBMD receives a BACnet broadcast on its local network segment, it forwards the message to the other BBMD's through IP routers via unicasts. Once received, the receiving BBMD retransmits the message on their subnet as an BACnet/IP broadcast.

BACnet routing with BBMD's is also referred to as "Annex J" routing because it is defined in that portion of the BACnet specification.

A more thorough explanation of BBMD's is available in the BACnet/IP Tutorial at *http://www.bacnet.org/Tutorial/BACnetIP/default.html http://www.bacnet.org/Tutorial/BACnetIP/default.html*

Configuring BBMD's on i-Vu Open routers requires the BBMD Configuration Tool. See the *BBMD Utility User Guide* for more information.

The following can function as a BBMD:

- i-Vu® Open Router
- i-Vu® Open Link
- i-Vu® integrated router

# BACnet addressing

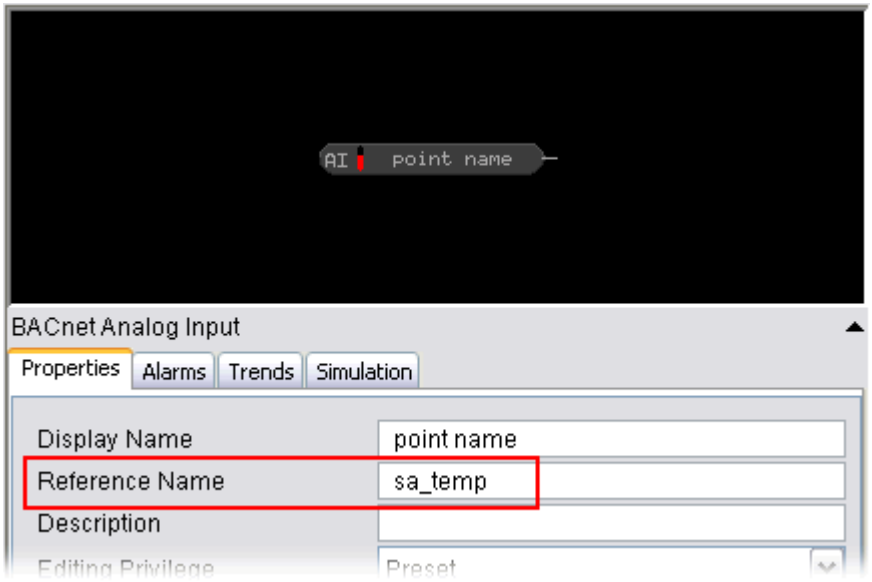
This section provides an overview of generic BACnet addressing and a detailed explanation of automatic addressing in the i-Vu® Open product line.

When using the i-Vu® application, set unique addresses on every controller's rotary switches, prior to installation. BACnet addresses are automatically generated, based on the rotary switches. Manual addressing is necessary only for specific installation requirements or in case of a conflict with other BACnet systems.

A complete BACnet address must identify the object and the device that references it. An object ID is the combination of the object type and its instance number. See *BACnet standard objects* (page 3) for common object types.

## Reference Names in the Snap application

The Snap application automatically gives each microblock a unique alphanumeric **Reference Name** such as "M001". You should change it to something intuitive such as "sa-temp", but the name must be unique within the system.




## Networks

The network number uniquely identifies a network in a BACnet system. The i-Vu® application automatically generates all network numbers.

Typical i-Vu® Control Systems have at least one MS/TP network with an IP backbone. All i-Vu® Open systems must have at least one MS/TP network. When using more than one MS/TP network, BACnet/IP must be used with i-Vu® Open routers to connect the MS/TP networks together with a single BACnet/IP backbone.

- **Network Numbers** - Network numbers must be unique on a BACnet system regardless of the protocol. BACnet MS/TP and BACnet/IP network numbers cannot be duplicated. A network number can range from 1 – 65,534 (with 65,535 reserved for broadcasts).
- **BACnet/IP** – In an i-Vu® Open Control System, all BACnet/IP devices (i-Vu® web servers, i-Vu® Open routers, etc.) default to a network number of 1600. We recommend you do not change this setting.
- **MS/TP** – MS/TP network numbers are determined by the router to the MS/TP network. Auto-generated BACnet MS/TP network numbers take the following form:

  **161XX**

  16 = Carrier's BACnet Vendor ID
  1 = MS/TP network
  XX = value of the router's rotary switches

  **NOTE** In stand-alone systems, when no router is used, XX = 01

# Objects

## Point Objects

Point objects describe a point, such as analog or binary inputs and outputs. See *BACnet standard objects* (page 3) for examples. In the i-Vu® application, the default address of any point is initially configured at the factory.

- **Object name -** an alpha-numeric string that describes a point and is unique within each device. Every point name was carefully selected to briefly describe the point. We recommend that you do **not** modify these names.
  Example: The object name that represents zone temperature in a VVT Open Zone controller is zone_temp.
- **Object type** - an identifier that describes the specific type of point, such as an analog input, binary output, etc. In Thei-Vu® application, an object type can be represented in many ways, numerically or with reserved words.
  Example: An object type representing an analog input in thei-Vu® application could be ai, analog-input, or 0.
- **Object instance** - a numeric value (1 to 4194302) that must be unique to each object type in each device. This value is set by the manufacturer and cannot be changed in the i-Vu® application. Common points, such as Outside Air Temperature, are usually found at the same instance number for consistency between controllers.
  Example: The object instance for zone temperature in a VVT Open Zone controller is 1.
- **Object ID** - a combination of the object type and a unique instance number. For example, in the i-Vu® application, AO:2 would be the second instance in a device of an analog output. The object ID is always unique because object instances must be unique within the device.
  Example: An object identifier that represents zone temperature in a VVT Open Zone controller is AI:0.

## Device Objects

The i-Vu® application automatically generates the address of the device object, based on the device's rotary switch setting.

- **MAC (Media Access Code) addresses** are the lowest level addressing scheme in BACnet. In i-Vu® Open Control systems, MS/TP MAC addresses range from 1 – 99 and must be unique for each device within an MS/TP network.

  The i-Vu® application uses the following conventions when auto-generating addresses:

  - **MS/TP Controllers** - the MS/TP MAC address is the two digit number defined by the rotary switches®
    **NOTE** The MAC address **cannot** be zero.

- **Routers** - have a separate MAC address for each port. On MS/TP to IP routers, there is an IP and an MS/TP MAC address. The IP MAC address is the IP address and BACnet port of the router. (Example: 192.168.1.100:0xBAC0 or 192.168.1.100.186.192). The MS/TP MAC address is always **0** on i-Vu® Open Routers and i-Vu® Open Links. See the *i-Vu® Open Router* or the *i-Vu® Open Link Installation Guide* for details on setting the IP address.

- **Device Instance** must be unique within the entire system. It is the Instance number of the Device object (Object type 8) in a BACnet system.
  The i-Vu® application uses the following conventions when auto-generating addresses:

  - **Controllers** - the device instance of controllers is a combination of the MS/TP network address and MAC address for all controllers.
    Example: A controller on network 16101 with a MAC Address set to 11 on the rotary switches has a Device Instance number of 1610111.

  - **Routers** – the Device Instance of routers is a combination of the BACnet/IP network number (default: 1600) and the value of the router's rotary switches
    Example: The router with rotary switches set for **01** will have a device instance of 160001.

  - **Device name -** an alpha-numeric string that is unique within the system, automatically generated as **device** followed by the device instance.

    Example: The device with instance 1610111 has a device name of **device1610111**.

# Addressing summary

**Legend**

16 = Carrier's BACnet Vendor ID

xx = Router's rotary address (Router Address)

yy = Open controller's rotary address (MS/TP MAC Address)

**Router:**

BACnet IP network number = 1600

BACnet device instance number = 1600xx

BACnet device instance name = device1600xx

BACnet MS/TP network number = 161xx

MS/TP MAC address = 0 (fixed)

**Open controllers that are connected to the router:**

BACnet device instance number = 161xxyy

BACnet device instance name = device161xxyy

MS/TP MAC address = yy

BACnet MS/TP network number = 161xx (supplied by the router, defaults to 16101 if there is no router.)

**If a router's MS/TP address is set to 200 and the connected Open controllers are auto-generated, their settings will be:**

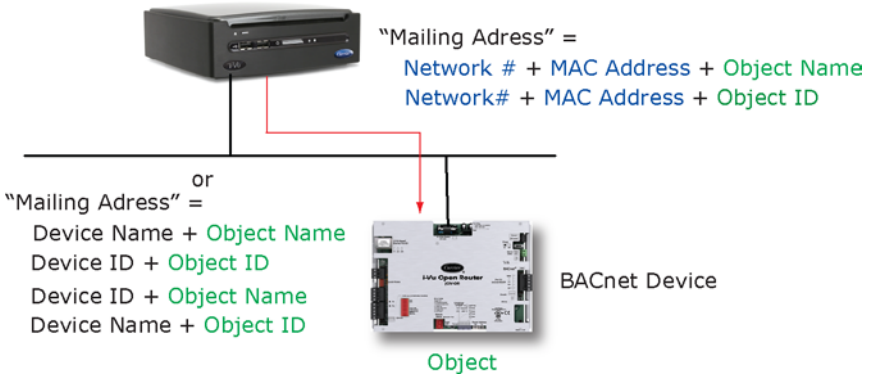
BACnet MS/TP network number = 200

BACnet device instance number = 200xxyy

BACnet device instance name = device200xxyy

MS/TP MAC address = yy

## Complete BACnet address formats




## To format a BACnet address

Use the information below to format a valid BACnet address for the microblock you are using to read or write to a third-party or Carrier BACnet object.

Address format: **bacnet://device/object/property@priority**

1 2 3 4

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

| | **EXAMPLES** |
|---|---|
| 1 **Device** - Use one of the following: | |
| Device instance number | `bacnet://1610101/...` |
| BACnet device name | `bacnet://device1610101/...` |
| Network number: MAC address (of third-party device) | `bacnet://16101:0x1/...` |

| | |
|---|---|
| The word "this" if a network point requests a value from another control program in the same Carrier controller. Avoids network traffic. | `bacnet://this/…` |

**2** **Object** - Use one of the following:

| | **EXAMPLES** |
|---|---|
| Object type: Instance number (See NOTES below) | `bacnet://…/ai:2`<br>`bacnet://…/analog-input:2`<br>`bacnet://…/0:2` |
| BACnet object name | `bacnet://…/zone_temp` |

**NOTES**

- For object type, you may type the abbreviation, 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.

**3** **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.

| | **EXAMPLE** |
|---|---|
| Number (1–16) | `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://1610101/ai:2
bacnet://1234:0x23/analog-input:2/
```

# Special rules for i-Vu® systems

### Internal Router

#### Manually defining network numbers greater than 40,000

Special rules apply when manually defining MS/TP network numbers on routers with an auto-generated device ID and having network numbers set above 40,000. Device instance numbers for MS/TP devices are comprised of the MS/TP network number and the rotary switch address of the device.

The maximum value for a device instance number is 4,194,302. If an MS/TP network number of 50,000 is used, a controller on that network (assume rotary switches set to 11) has a device instance of 5000011, which exceeds the maximum value. To circumvent this issue, 40,000 is subtracted from any network number greater than, but not equal to, 40,000. In the example above, a controller on the network with rotary switches set to 11 has a device instance of 1000011.

Another example using 40001 as the network number results in a device instance of 111 for the controller with rotary switches set to 11.

#### Networks using i-Vu® routers

Whenever an i-Vu® Open Router or i-Vu® Open Link is used in a i-Vu® system, only one router per network must be addressed as 01. This router is the main router, known as the color-caching master. It broadcasts time and it controls other vital inter-router communications. The concept of color-caching is important to the i-Vu® application operating properly, but its function is outside the scope of this document.

#### Multiple i-Vu® systems on an IP network

Multiple i-Vu® systems on a single IP network is not recommended. There is no way to limit an i-Vu® Open Router or i-Vu® Open Link to a single server so all i-Vu® servers will be able to communicate with all i-Vu® Open Routers or Links on the logical network. This includes networks that are spanned with BBMD's.

This issue goes beyond the need for unique rotary switch settings on each router, since, as previously stated, every i-Vu® Open system must have a router using rotary address 01. This leads to one of two conflicts; either there are two routers acting as the color-caching master (as described above) or the color-caching master is communicating with routers outside of the BACnet system.



A member of the United Technologies Corporation family · Stock symbol UTX · Catalog No. 11-808-417-01 · 9/23/2013