KMC Connect™ reference

Includes installation instructions

Revision F

Printed in U.S.A.

KMC Controls, Inc.

19476 Industrial Drive

New Paris, IN 46553

U.S.A.

TEL: 1.574.831.5250

FAX: 1.574.831.5252

E-mail: info@kmccontrols.com

Contents 3

About this manual 11
- **What you need to know** 11
- **Conventions used** 11
- **If you encounter difficulty** 12

Part I: Setting up KMC Connect 13

Section 1: About KMC Connect 15
- **Choose your computers** 16
- **Setting up computers** 18
  - **Computer names** 18
  - **Passwords and privileges** 18
  - **Sleep and hibernation modes** 18
- **Installing KMC Connect** 19
- **Uninstalling KMC Connect** 20

Section 2: Getting started 21
- **Starting KMC Connect** 22
- **Adding a new site** 23
- **Open an existing site** 25
- **Closing a site** 26
- **Working offline** 26

Part II: Working with KMC Connect 27

Section 3: The KMC Connect workspace 29
- **Customizing the workspace** 30
- **Options for KMC Connect** 34

Section 4: Using Network Manager 37
- **Opening and closing the Network Manager pane** 38
- **Network Manager toolbar commands** 38
- **Discovering devices** 39
- **Configuring an existing service** 40
- **Changing the Network Manager options** 41
- **Naming BACnet networks** 42
- **Restarting BACstac from the Network Manager** 42

Section 5: Output Window 43
- **Opening and closing the Output window** 43
- **Sorting the list in Output Window** 44
- **Output window options** 44

Section 6: About Resource Manager 45
- **Opening and closing the Resource Manager pane** 47
- **Editing items in the Resource Manager list** 47
- **Restoring a configuration** 48
- **Creating backup files** 51

Managing the Resource Manager list ... 51
Adding files to the Resource Manager folder ... 52

Section 7: Viewing objects with Table View ... 55
Viewing groups of similar objects ... 56
Custom Table Views ... 59

Section 8: Backing up and restoring a site ... 63
Backing up a site ... 63
Restoring site data ... 64

Section 9: Security and passwords ... 65
KMC Connect basic security ... 66
About the Security Manager ... 68
User tab ... 69
Groups tab ... 70
Adding a new user to a group ... 71
Changing a password ... 72
Deleting a user ... 72
Adding and modifying groups ... 73

Section 10: The KMC Connect Audit Log ... 75

Section 11: Updating firmware ... 77

Part III: Trend logs ... 79

Section 12: About trend logs ... 81

Section 13: Viewing trend logs in KMC Connect ... 83
Viewing with the Trend Log viewer ... 84
Starting a live trend log ... 84
Starting from the Network Manager ... 84
Viewing data from the Trend Manager ... 85
Parts of the Trend Viewer ... 85
Viewing controller trend logs ... 87

Section 14: Configuring BACnet trend logs and groups ... 89
Trend configuration wizard ... 90
Start the wizard ... 90
Add objects by searching ... 91
Configuring BACnet controller trend logs ... 92
Configuring BACnet device trend logs ... 94
Configuring a BACnet PC trend log ... 96
Adding BACnet trend logs to trend groups ... 98
Archiving trend logs ... 100

Part IV: Alarms, notifications, and events ... 101

Section 15: An overview of notifications, alarms, and events ... 103

Section 16: Configuring devices for BACnet alarms ... 105
Using the Alarm wizard for BACnet alarms ... 106

Start the wizard ... 106
Add by searching ... 107
About intrinsic BACnet alarms and events ... 108
An example of intrinsic notification ... 109
Notifications in analog objects ... 109
Notifications in binary objects ... 110
Configuring BACnet devices for intrinsic alarms ... 111
About algorithmic reporting in BACnet devices ... 113
Configuring event enrollment objects ... 114
Configuring a notification class object ... 116

Section 17: Change Of Value (COV) notifications ... 119
The parts of COV ... 119
COV server ... 119
COV client ... 120
Subscribing to an object on a configuration page ... 121
Subscribing to a property on a configuration page ... 122
Unsubscribing from COV subscriptions ... 123
Subscribing trend logs to a COV server device ... 124
Viewing all event enrollment subscriptions ... 124

Section 18: Viewing, acknowledging, and clearing alarms and events ... 127
Viewing alarms ... 128
Acknowledging alarms ... 130
Clearing alarms ... 130
Viewing a summary of BACnet notifications ... 130
Using the Alarm Monitor bar ... 131
Filtering alarms ... 133
Disabling alarms by network ... 134

Section 19: Emailing operator alarms ... 135
Setting up alarm emails ... 135
The components of the alarm routing manager ... 135
Changing the listing view ... 136
Configuring email notifications ... 136
Recipient operators ... 136
Alarm Message Delivery Details ... 136
Alarm Class ... 137

Section 20: Filtering alarms by security groups ... 139

Section 21: Configuring the Notification Service ... 141
General Properties ... 141
Alarm Printer ... 141
Email settings ... 142

Part V: Schedules ... 143

Section 22: About KMC Connect managed schedules ... 145
Configuring BACnet schedules ... 145
Setting the Weekly schedule time value pairs ... 146
Setting up the Exception schedule ... 147

Setting up a Calendar object 148

Part VI: Control Basic and the Code Editor 149

Section 23: Programming with the Code Editor 151
Writing Control Basic programs in controllers 152
Writing programs offline 154
Automatically numbering program lines 156
Keyword prompting 156
Using keyboard shortcuts 157
Finding and replacing text 158
Changing Code Editor options 162

Section 24: About Control Basic programs 165
About Control Basic scans 165
Statements 166
Multiple statements 166
Functions 167
Expressions 167
Labels and line numbers 167
Line numbers–Standard Control Basic only 168
Labels in Next Generation Control Basic 168
Programming format and notation 169
Real numbers 169
Hierarchy of operators 170
Relational operators 171
Using arithmetic operators 172
Using Boolean logic 172
Programming with variables 173
BACnet value objects as variables 173
Local variables 173
Transferring values between BACnet controllers 173
Reading properties from other BACnet devices 174
Writing values to other controllers 174
Programming with mnemonics 174
User defined procedures and functions 176
Data types 179

Section 25: Keywords for Control Basic 181
Using example programs from help 181
Syntax for commands and functions 181

Section 26: Writing block programs 233
Applicable controllers 233
Starting the Block Editor 234
A tour of the Block Editor 236
The parts of the editor 236
The Block Editor toolbar 237
Get and Set blocks 238
Get blocks 238
Set blocks 238
Defining local variables 239

Adding and editing routines 240
Macro blocks 241
Limitations and importing 245

Section 27: Reference to Control Basic blocks 247

Part VII: BACnet objects 289

Section 28: BACnet overview 291
BACnet objects and properties 291
An example BACnet object 291
BACnet properties 291
BACnet services 292
Priority arrays 292
An example priority array 292
Standard BACnet priority levels 294

Section 29: Configuring and connecting to BACnet devices and networks 297
Adding BACnet devices to the Network Manager list 298
Configuring BACnet devices and objects 300
Creating and deleting BACnet objects 302
Converting input and output objects 303
Regenerate networks and devices 304
Restart a BACnet device 306
Setting BACnet system time 307
Automatic MAC Addressing commands 308
Scanning BACnet configuration files 308
Backing up and restoring BACnet devices 309
BACnet Backup and restore 309
Importing and exporting .bnd files 310
Importing and exporting BACstage .bac files 310
Configuring a BACnet service 311
Configuring the IP connection 313

Section 30: Reference to BACnet objects 315
Access Door object 316
Averaging objects 320
Calendar object 321
Command object 322
Device objects 323
Event enrollment object 329
Event Log objects 333
File objects 336
Group objects 336
Input object–accumulator 336
Input object–analog 340
Input object–binary 345
Input objects–multistate 348
Life Safety Point object 351
Life Safety Zone object 354
Load Control objects 358
Loop objects 361

Notification object 365
Output objects–analog 366
Output objects–binary 370
Output object–multistate 373
Program object 377
Schedule object 379
Structured View objects 383
Trend Log object 384
Trend Log Multiple objects 388
Value objects–analog 392
Value objects–binary 396
Value objects–multistate 399

Section 31: Configuring NetSensors 403
KMD series NetSensors for BACnet 404
Configuring the STE-9000 series NetSensor 409
General tab 409
Home Display tab 410
User Setpoints tab 411
Advanced tab 411
Optional Menus tab 412

Section 32: BACnet tables 413
Input tables 413
CB tables 414

Section 33: Supported engineering units 417

Section 34: Reference to KMC BACnet controllers 419

Part VIII: Applications and wizards 423

Section 35: Application selection, configuration, and deployment 425

Section 36: Audit application 429

Section 37: VAV Balancing application 433
The VAV Balancing sequence 433
VAV flow monitoring 436
Commanding VAV airflow 437

Section 38: Custom Applications Wizard 439

Section 39: Input object wizards 441
Analog input object wizard 441
Binary input object wizard 445

Appendices 447

Appendix A: Configuring the Cimetrics BACstac driver 449
Before you begin 449
Configuring BACstac for BACnet IP 451
Configure the computer as a BACnet IP device 451

Register as a foreign device 453
Configure BACstac as a BBMD 454
Configure BACstac for Ethernet 8802.3 456
Remove previous versions of the BACstac driver 457

Appendix B: Versions of Control Basic 459
Control Basic versions in controllers 460
Deprecated keywords 461
Changes to IF THEN 462
File compatibility 462
TotalControl and KMC Connect compatibility 462
BACstage file compatibility 462
New keywords 463
Next Generation 463
Generation 5 464
Line numbers and labels 467
Line numbers 467
Labels 467
Local variables 468
References to objects in remote devices 468

Appendix C: Conquest compliant controllers 469

Appendix D: Glossary 471

Index 483

About this manual

This publication is an installation and operation manual for KMC Connect. Review this material in its entirety before installing the KMC Connect program.

This manual covers installation for KMC Connect. Sections in this manual include the following topics:

- Information you need to gather before you begin installing KMC Connect.
- Requirements and specifications for the computer to run KMC Connect.
- Installing KMC Connect.
- Customizing the installation of KMC Connect.
- Uninstalling KMC Connect.

What you need to know

This manual assumes your familiarity with the following:

- Your computer and the Windows operating system.
- The principles of the building automation systems that will be configured with the KMC Connect program.

In addition to the material presented in this user's guide, review and have available the following reference material.

- The user's guides for each device in the system.
- A sequence of operation for each piece of equipment in the site.
- Detailed plans and drawings for the building automation system.
- Information about the Local Area Network (LAN) including routers, switches, and network firewalls.

Conventions used

Some of the text in this publication uses special formatting to indicate emphasis or keystrokes. The text conventions are as follows:

Menu and dialog items	Highlights items in the KMC Connect interface, including buttons, dialog names, menus, and commands in menus.
`File names`	Highlights names of files and extensions.
Italics	Indicates a book or section title.
KEY NAMES	Indicates a specific key on the keyboard such as SHIFT or ENTER.

If you encounter difficulty

If you experience difficulty with KMC Connect, KMC Controls provides the following assistance.

Printed version of help An Adobe Acrobat version of KMC Connect help is included on the installation flash drive. The PDF document is identical to the on-line help, but it is formatted to print as a reference manual.

The KMC Controls web site Navigate to the support section on the KMC controls web site for the latest information for KMC Connect and other KMC Controls products.

www.kmccontrols.com

KMC technical support Our distribution partners have unlimited and free access to our team of Technical Support representatives. We provide coast-to-coast and toll-free support from 8:00 AM Eastern to 5:00 PM Pacific time.

Toll-Free Technical Support: (866) 303-4562

KMC Connect

Part I: Setting up KMC Connect

Section 1: About KMC Connect

KMC Connect is a master operator workstation with which you can program BACnet controllers for a building automation system. This section covers choosing and setting up computers and installing KMC Connect.

To install KMC Connect you will need the following items and information:

- Detailed plans and drawings for the building automation system.
- Information about the LAN including routers, switches and firewalls.
- Installation files from either a KMC Connect flash drive or an upgrade file available from the Software downloads area in the Partner Portal on KMC Controls web site.
- A hardware license key that includes a KMC Connect license.
- A password to establish for KMC Connect administrator for the computer on which KMC Connect will be installed.

Topics covered in this section

- *Choose your computers* on page 16
- *Setting up computers* on page 18
- *Installing KMC Connect* on page 19
- *Uninstalling KMC Connect* on page 20

Choose your computers

Computers running KMC Connect must meet minimum hardware and operating system requirements. Review the information in this section before installing KMC Connect.

Table 1-1 Windows 10 computer requirements

Component	Requirements
Operating system	Windows 10 Professional
Processor speed	2 GHz or faster
RAM memory	2 gigabyte (GB) for 32-bit or 64-bit
Graphics adapter	DirectX 9 graphics processor

Table 1-2 Windows 8 computer requirements

Component	Requirements
Operating system	Windows 8 and 8.1 Professional
Processor speed	2 GHz or faster
RAM memory	2 GB or greater
Graphics adapter	DirectX 9 graphics processor

Table 1-3 Windows 7 computer requirements for KMC Connect

Component	Requirements
Operating system	Windows 7 Professional Windows 7 Ultimate
Processor speed	2 GHz or faster
RAM memory	2 GB or greater
Graphics adapter	DirectX 9 graphics processor

Table 1-4 Windows Server computer requirements for KMC Connect

Component	Requirements
Operating system	Windows Server 2008 R2 Windows Server 2012 R2
Processor speed	2 GHz or faster
RAM memory	2 GB or greater

Table 1-5 Computer platform requirements for KMC Connect

Component	Recommended	Minimum
Hard disk space	160 GB 5 GB free	80 GB 5 GB free
Monitor	SVGA 1280 x 1024	SVGA 1024 x 768
Network connection	Ethernet 100BaseT	Ethernet 10BaseT
USB	USB port dedicated to hardware key	

Setting up computers

Computers hosting KMC Connect must meet the requirements listed in the section *Choose your computers* on page 16. Before installing KMC Connect, verify the computer name, passwords and privileges, and the sleep and hibernation modes.

Computer names

Assign a name to the computer that consists of only numbers and letters.

Passwords and privileges

Have available an administrator's user password for the computer on which KMC Connect will be installed. The installing technician must have Windows administrator privileges.

Sleep and hibernation modes

To avoid corrupting the hardware license key, change the power settings for the computer. To change the power settings for your version of Windows, do the following:

1. Navigate to the Windows desktop and press F1.
2. When help opens, search for “power” or “power management”.
3. Choose the help topic for changing or setting up a power scheme.
4. In the power scheme change the computer power settings to disable hibernation and set the sleep mode to “never”.

Installing KMC Connect

You must install KMC Connect from the USB flash drive onto a computer hard disk; KMC Connect will not run from the flash drive. To install KMC Connect, you will need the following:

- The installation USB flash drive or an installation file from the KMC Controls partners portal.
- A password for the site administrator.

Note: A hardware license key is not required to install KMC Connect. However, you will need a key to start KMC Connect after installation.

To start the KMC Connect installation wizard, do the following:

1. Insert the flash drive into any USB port.
2. Use Windows Explorer to locate and open the flash drive. The flash drive is labeled Removable Disk.
3. On the flash drive double-click the **SETUP** icon.
4. Follow the on-screen installation instructions.
5. When prompted, choose a location for the program. KMC Controls recommends the default location.
6. When installation is complete, remove the flash drive and store it in a safe location.

Once KMC Connect is installed, see the topic *Getting started* on page 21.

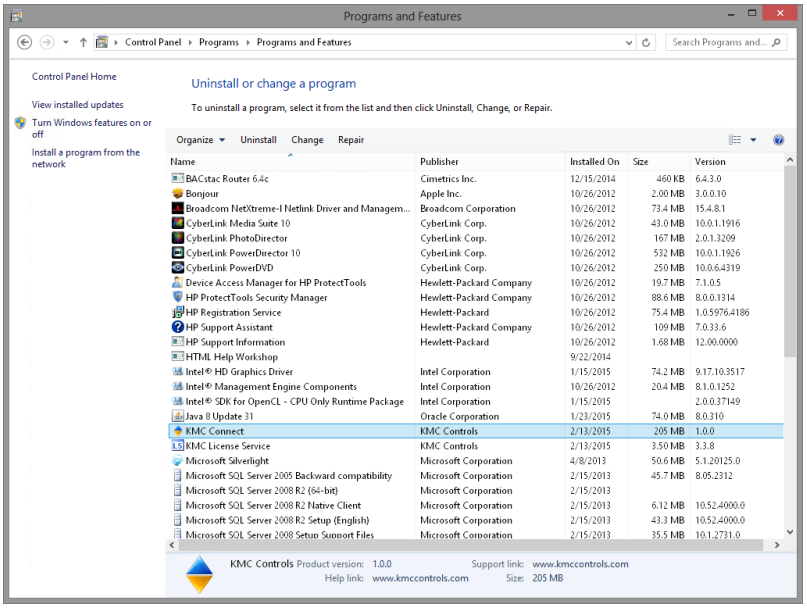
Uninstalling KMC Connect

This topic covers the removal of KMC Connect when it is no longer needed.

To uninstall KMC Connect do the following:

1. Click **Start**, **Settings**, **Control Panel** and then choose **Add or Remove Programs.**
2. Scroll to programs in the **Currently installed programs** list. See the illustration *Installed programs* on page 20 for a list of the programs.
3. Choose **KMC Connect** and then click **Remove**.
4. The following programs should also be uninstalled unless other programs, such as TotalControl, are installed on the same computer.
   - Choose **KMC License Service** and then click **Remove**.
   - Choose **BACstac Router** and then click **Remove**.
5. When the procedure is complete, restart the computer.

Illustration 1–1 Installed programs



Section 2: Getting started

KMC Connect is a master operator workstation with which you can program BACnet controllers for a building automation system. This section covers the first steps toward using the program.

With KMC Connect you can:

- View or change controller operation
- Set up alarms and notifications
- Set up and view historical trend data
- Set schedules
- Edit Control Basic programs

To use KMC Connect as a BACnet operator workstation you will need the following items:

- A hardware license key.
- For existing sites—the user name and password established for the site
- For new sites—a new password for the site administrator

The following topics are covered in this section.

- *Starting KMC Connect* on page 22
- *Adding a new site* on page 23
- *Open an existing site* on page 25
- *Closing a site* on page 26
- *Working offline* on page 26

Starting KMC Connect

KMC Connect starts the same as other Windows applications.

Note: The first time a hardware key is plugged into a computer, Windows will notify you that new hardware has been found. In the following procedure, step 2 may not be required after the key is inserted for the first time.

To start KMC Connect do the following:

1. Insert a valid hardware key or verify that a hardware key is inserted into a USB port. The key must remain in the USB port at all times KMC Connect is operating.
2. If the Found New Hardware Wizard opens, do the following.
   - Choose the **Install the software automatically** option. This may take several minutes to complete.
   - When the wizard finishes installing the software for the hardware key, installation is complete.
3. Choose **Start** > **Programs** > **KMC Controls** > **KMC Connect**.
4. Once KMC Connect starts, you can choose a site to open. If a site has not been set up on the computer, you can choose to run KMC Connect without connecting to a site or set up a new site.

Related topics

- *Adding a new site* on page 23
- *Open an existing site* on page 25
- *Closing a site* on page 26
- *Working offline* on page 26

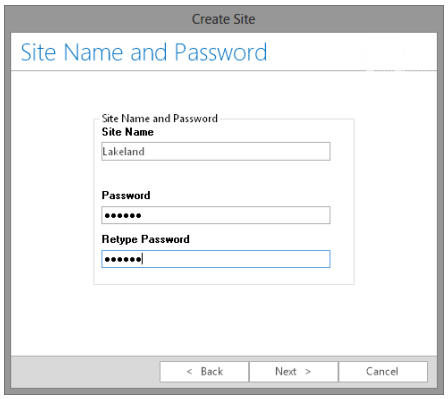
Adding a new site

Adding a new site sets up KMC Connect with all of the files, user names, and passwords for a specific building automation system. Typically every building automation system that a control technician manages will have its own site on the technician's computer.

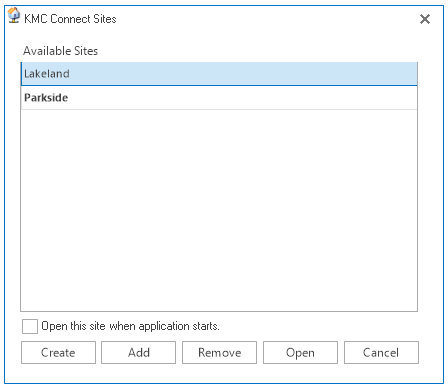
Note: The following procedure sets up a new site but does not configure BACstac for a connection to the BACnet internetwork. Verify or set Cimetrics BACstac to match the BACnet routing protocol, network, and port established by a BACnet router. See the topic *Configuring the Cimetrics BACstac driver* on page 449.

To start a new site in KMC Connect, do the following:

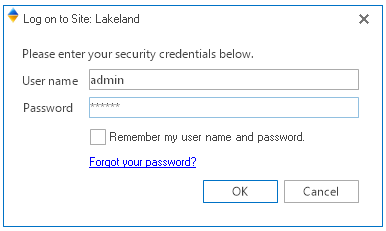
1. On the **Site** tab click **New**. The Site Name and Password dialog opens.
2. Enter the site name and a password for the site administrator.



3. Click **Next** to continue. When the Complete dialog opens, click **Finish**.
4. When the KMC Connect Sites dialog opens, select the site and then click **Open**.



5 In Log on to Site, enter the user name admin and the password entered in Step 2.



6 When the site opens, change the Device Instance number of the BACnet service if it does not match system plans or if there is a conflict with another device. To change the Device Instance, right click the BACnet service icon and choose Configure Service from the shortcut menu. See *Configuring a BACnet service* on page 311

Related topics

- *Open an existing site* on page 25
- *Closing a site* on page 26
- *Working offline* on page 26

Open an existing site

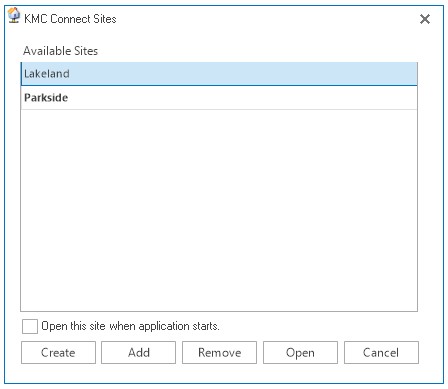
To open an existing site, you will need the following items:

- A hardware license key.
- The user name and password established for the site.

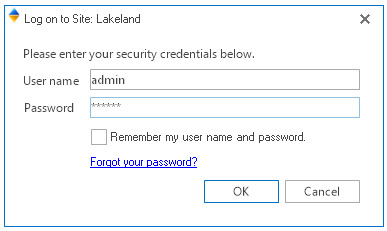
Note: The following procedure opens an existing site but does not configure BACstac for a connection to the BACnet internetwork. Verify or set Cimetrics BACstac to match the BACnet routing protocol, network, and port established by a BACnet router. See the topic *Configuring the Cimetrics BACstac driver* on page 449.

To open an existing KMC Connect, do the following:

1. On the **Site** tab click **Open**.
2. When the KMC Connect Sites dialog opens, select the site by name and then click **Open**.



3. In the **Log on to Site** text box, enter the user name and the password.



4. If necessary, configure BACstac as described in the topic *Configuring the Cimetrics BACstac driver* on page 449.

Related topics

- *Adding a new site* on page 23
- *Closing a site* on page 26
- *Working offline* on page 26

Closing a site

Close an open KMC Connect site from the ribbon. On the **Site** tab click **Close**. With the site is closed, you can continue to work offline.

- See *About Resource Manager* on page 45 to program and configure a device or object file in the Resource Manager.
- See *Application selection, configuration, and deployment* on page 425 to configure a job for Conquest compliant controller.

Related topics

- *Adding a new site* on page 23
- *Open an existing site* on page 25
- *Working offline* on page 26

Working offline

KMC Connect supports offline programming at any time even if a site is open.

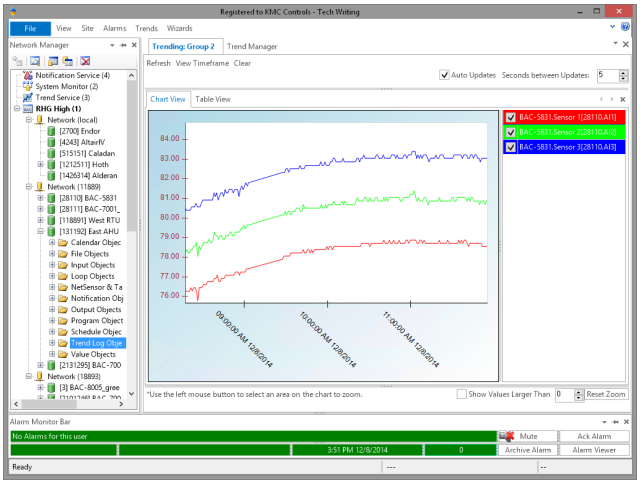
- See *About Resource Manager* on page 45 to program and configure a device or object file in the Resource Manager.
- See *Application selection, configuration, and deployment* on page 425 to configure a job for Conquest compliant controllers.

Related topics

- *Adding a new site* on page 23
- *Open an existing site* on page 25
- *Closing a site* on page 26

KMC Connect

Part II: Working with KMC Connect

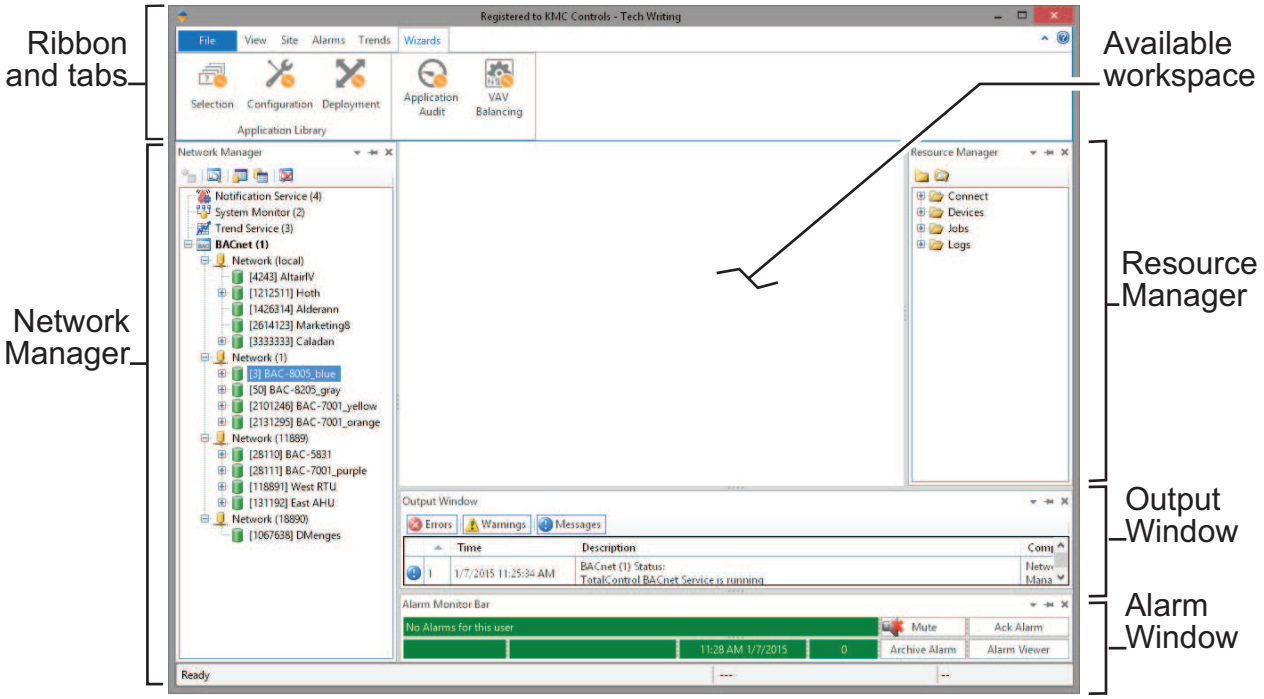


Section 3: The KMC Connect workspace

KMC Connect is a master operator workstation for building automation systems. Topics in this section explain the basic components of the KMC Connect.

KMC Connect is a collection of software modules with which you can configure controllers, and set up security, trends, schedules, and alarms.

Illustration 3–1 KMC Connect workspace



Site management tools The most often used components of KMC Connect are the dockable site management tools. Each of these tools open in a pane in the KMC Connect workspace.

- *Output Window* on page 43 displays recent actions and messages about things that may need the operators attention.
- *Using Network Manager* on page 37 is an expandable list of all controllers, networks, and services in a KMC Connect managed building automation system.
- *About Resource Manager* on page 45 is a tool for managing templates and working offline to create configuration files, and Control Basic programs.

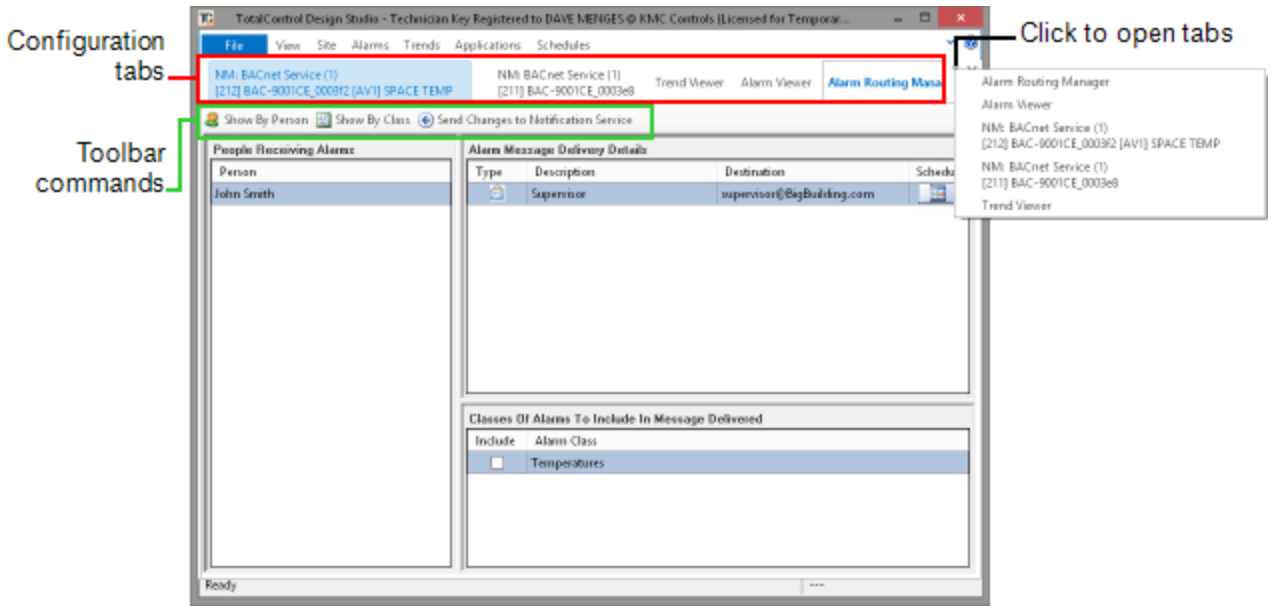
Alarm Monitor bar The alarm window is a scrolling list of alarms and notifications that have yet to be acknowledged and archived. See the topic *Using the Alarm Monitor bar* on page 131 for details.

Configuration tabs Components in a KMC Connect managed building automation site are set up and configured from configuration tabs. The tabs may be opened from KMC Connect ribbon or from one of the site management tools.

- To set up and view data trends, see *About trend logs* on page 81.
- To write Control Basic programs, see *Control Basic and the Code Editor* on page 149.
- To configure a service, network, device or object, see *Using Network Manager* on page 37.
- To set up alarms and notifications, see *Alarms, notifications, and events* on page 101.

Configuration tabs may also include toolbar commands that are specific to the tabs. The commands are explained in the sections for that configuration tab.

Illustration 3–2 Parts of a configuration tab



To locate a hidden tab

A tabbed page may be open but, because of space available, its tab may not be visible in the workspace. Click the down arrow ▾ to open the tabs menu.

To close the active configuration page

- Click ✕ in the upper right corner of the page.
- Right-click the tab and choose **Close**.

Related topics

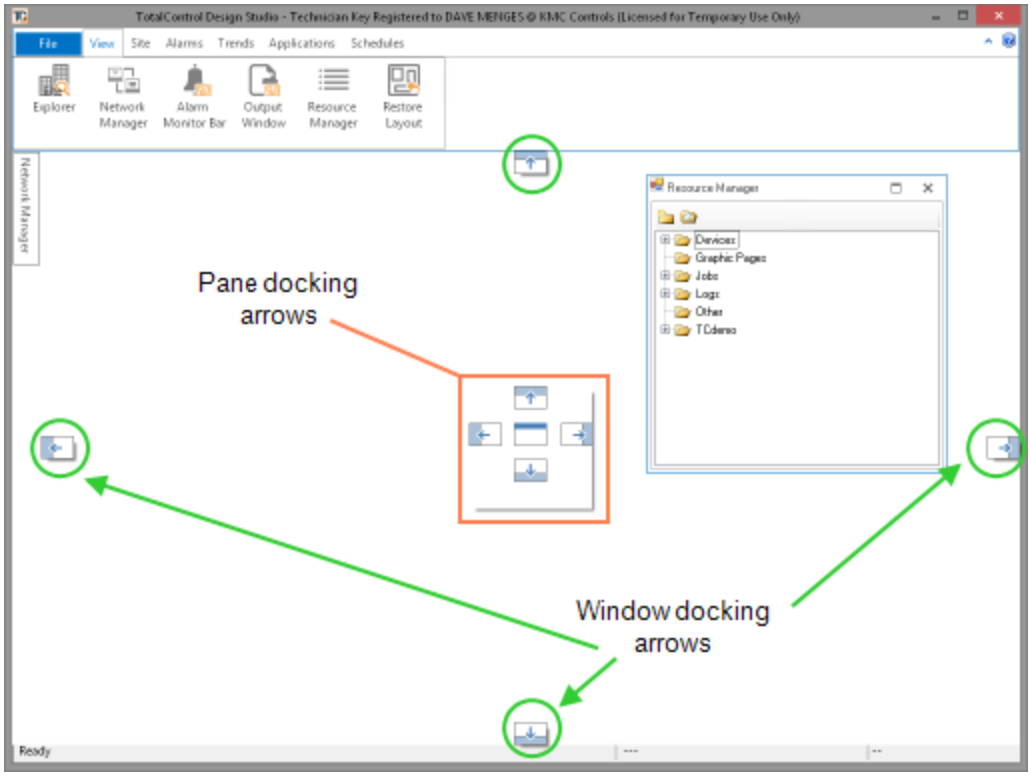
- *Customizing the workspace* on page 30
- *Options for KMC Connect* on page 34

Customizing the workspace

When first installed, KMC Connect opens with the panes for the site management tools—Network Manager, Output Window, and Resource Manager—docked and visible.

Configuration tabs are opened in the remaining workspace. The available workspace can be controlled by docking, floating, or hiding the panes for the site management tools.

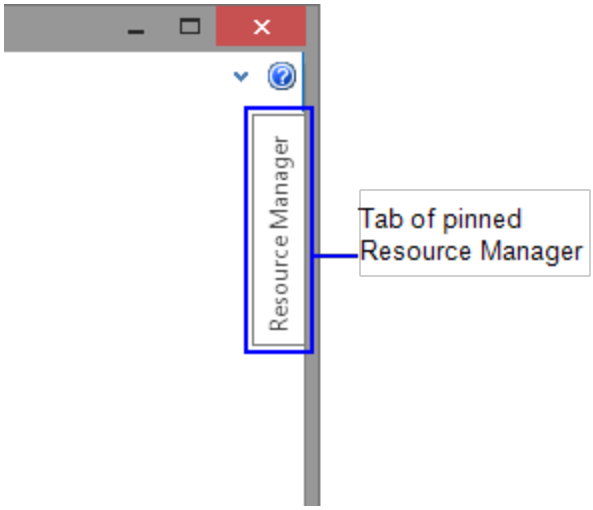
Illustration 3–3 Docking arrows



Docked panes—When a pane is docked, it is stationary and visible against a side of the KMC Connect window. Network Manager, Resource Manager, and Output Window are all docked when KMC Connect opens. When docked, any of these management tool panes can be hidden or changed to floating panes.

Hidden panes—Docked panes may be hidden from view and kept ready to use when needed. A hidden pane's location is indicated by a tab along the edge of the window.

Illustration 3–4 Tab for pinned management tool pane



Floating panes—A floating pane is detached from the KMC Connect window and can be moved to any location on the desktop.

To change a docked management tool pane to a floating pane.

- Drag the title bar of the pane to a new location.
- Double click the title bar. The pane moves into the workspace.

To set a management tool pane to Auto-Hide

Setting a site management tool pane to Auto-Hide pins the pane against the edge of the workspace. Only a docked pane may be set to Auto-Hide. A tab marks the location of the hidden tab.

1. If the pane is floating, dock the pane.
2. Do one of the following:
   - Click 📌 in the title bar. The pane is pinned against the edge of the workspace to which it is docked.
   - Right-click the title bar and choose **Auto-Hide**.

Viewing a management tool pane in Auto-Hide

To display a pinned pane, hover the pointer over the tab of the pinned pane. The pane remains visible until you click outside of the pane or reveal another pane.

To change a hidden pinned pane to a docked pane

1 Display the pane.

2 Do one of the following:

- Click 📌 in the title bar. The pane docks along the edge of the workspace.
- Right-click the title bar and choose **Dockable**.

To change a floating page to a docked pane

1 Click the title bar of the pane. The docking arrows appear in the KMC Connect window.

2 Drag the pane to one of the docking arrows.

- Pane docking arrows–When released over the pane arrows, KMC Connect splits the stationary pane and docks the floating pane in the position indicted by the arrow.
- Window arrow–KMC Connect docks the floating pane against the entire edge of the KMC Connect window.

To close a management tool pane

To close a pane that is floating, docked, or pinned, from the **View** tab on the ribbon, click the name of the management tool.

To open a management tool pane

To open a pane that is not docked, pinned or floating, do one of the following:

- On the KMC Connect toolbar, click the icon for the pane.
- From the **View** tab on the ribbon, choose the site management tool by name. Its pane will become active in the same location in which it was closed.

Options for KMC Connect

The General Application options set the way KMC Connect displays a notification pop up and plays the audible alarm when it opens.

To open the General Applications option dialog, do the following:

1. On the ribbon click the **File** tab and then click **Options**.
2. From the **Component** list choose **General Applications.**
3. Change settings or selections.
4. Click **Apply** or **Ok** when finished.

Save My Window Settings The Save My Windows Settings check box saves the configuring of the site management tools when KMC Connect opens.

- Select the **Save My Windows Settings Upon Exit** check box to set KMC Connect to open the next time with the present configuration of site management tools.
- Clear the **Save My Windows Settings Upon Exit** check box to set KMC Connect to always open with the present configuration of site management tools.

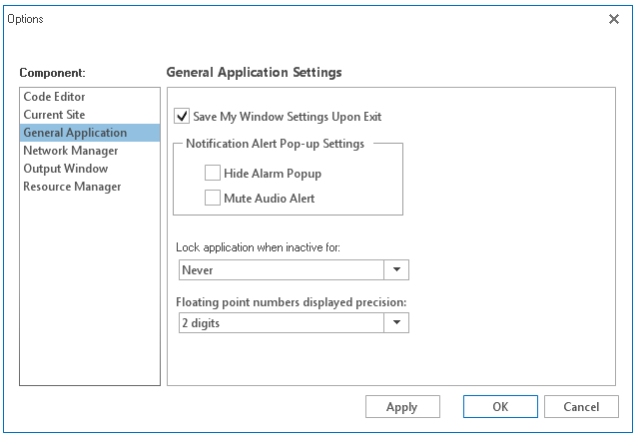
Notification Alert Popup Settings KMC Connect will open an alarm popup and play an audible alarm when it receives a notification from a device on a building automation system network. You may enable or disable the popup or audible alarm with the check boxes in the General Application options dialog.

- Select the **Hide Alarm Popup** check box to disable the alarm popup from opening when KMC Connect processes an alarm.
- Select the **Mute Audio Alert** check box to disable the audible alarm when KMC Connect processes an alarm.

Lock application when inactive Locks KMC Connect after the specified time. Enter a user name and password to unlock the application.

Floating point numbers displayed precision Change the precision of displayed floating point numbers. Default is 2 decimal places.

Illustration 3–5 General Application options dialog



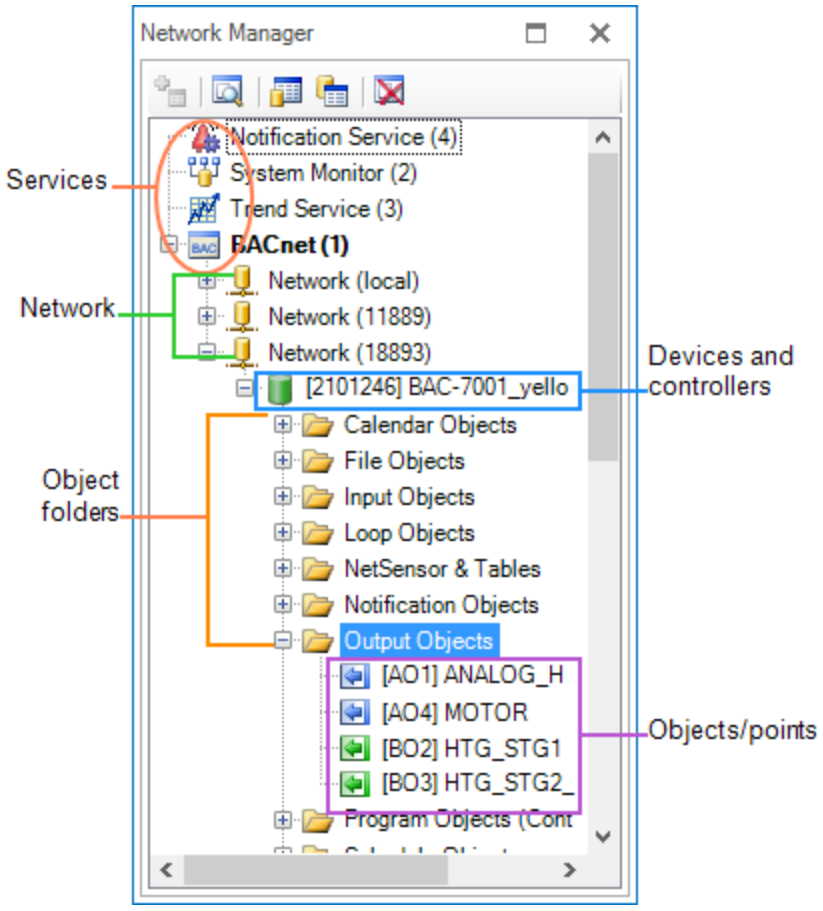
Section 4: Using Network Manager

Network Manager is the site management tool with which you can view and configure networks, devices, objects and points.

Networks, devices, objects, points and properties are displayed and configured through Network Manager. The Network Manager list is a display of the actual building automation system job site to which KMC Connect is connected. The list can be expanded to manipulate a specific point or property or collapsed to view only networks, devices or controllers. Use Network Manager for the following:

- Display the structure of KMC Connect services, networks, and devices in the building automation system job site.
- To locate and configure objects, properties, and points in the BAS controllers.

Illustration 4–1 Typical Network Manager



Configuring controllers

- *Configuring and connecting to BACnet devices and networks* on page 297

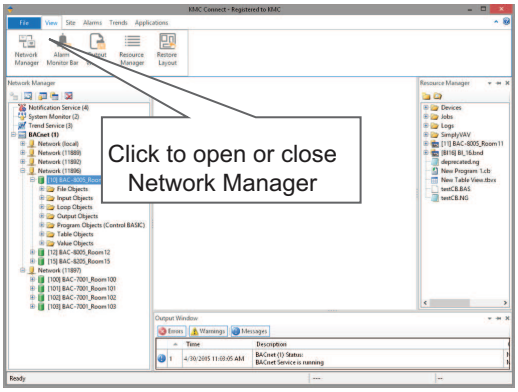
Related topics

- *Using Network Manager* on page 37
- *Network Manager toolbar commands* on page 38
- *Opening and closing the Network Manager pane* on page 38
- *Changing the Network Manager options* on page 41

Opening and closing the Network Manager pane

To open or close the Network Manager pane, on the **View** group click **Network Manager**.

Illustration 4–2 Open and close the Network Manager

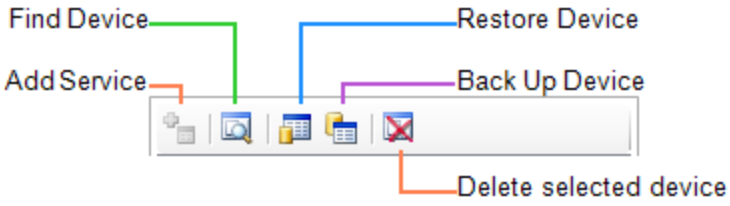


Network Manager toolbar commands

Use one of the following methods to choose a Network Manager command.

- Click the icons on the Network Manager toolbar.
- Right-click an icon in the Network Manager list and then choose a command from the shortcut menu.

Illustration 4–3 Network manager toolbar



Find Device Locates the devices connected to the building automation system. Once located, the devices are displayed in the Network Manager list. See *Changing the Network Manager options* on page 41.

Add Service Connect a KMC Connect Building Service to the site database.

Restore Device Configure a device with the contents of a backup file.

Backup Device Save the configuration of a device in a backup file.

Delete selected item Remove the selected item from the Network Manager list.

Discovering devices

Discovery is the process of populating the Network Manager list with the devices on the network. KMC Connecttwo types of discovery.

Limited discovery—KMC Connect will read only the device object properties. This discovery is faster but may limit other functions. Devices are discovered under limited discovery are represented by a limited discovery icon  .

Full discovery—KMC Connect reads all objects and properties in every device. Devices discoverd under full discovery are represented by a full discovery icon  .

Initial discovery

1. Start KMC Connect.
2. In the Network Manager list, locate the service icon  .
3. Right-click the icon and choose Regenerate All Networks from the shortcut menu.

Updating a network

1. In the Network Manager list, locate the a network icon  .
2. Right-click the icon and choose one of the following from the shortcut menu.
   - Regenerate Network—Deletes all current devices in the network and repopulates the network list with only the devices currently on the network.
   - Find Devices—Leaves the existing devices unchanged and adds new devices to the network to the list.

Limited discovery

To make the discovery process faster, KMC Connect can be set for limited discovery. In limited discovery, only the device object properties are loaded into the KMC Connect Network Manager list.

1. On the ribbon click the **File** tab and then click **Options**.
2. From the **Component** list choose **Current Site**.
3. Choose either **Limited Discovery** or **Full Discovery**.
4. Click **Apply** and then **OK**.

Complete discovery

Use complete discovery to fully discover all objects in selected devices.

1. Locate the service icon, network icon, or the device icon.
2. Right-click the icon and choose Complete Discovery from the shortcut menu. When discovery is complete, device icons change to.

See the related topic *Adding BACnet devices to the Network Manager list* on page 298.

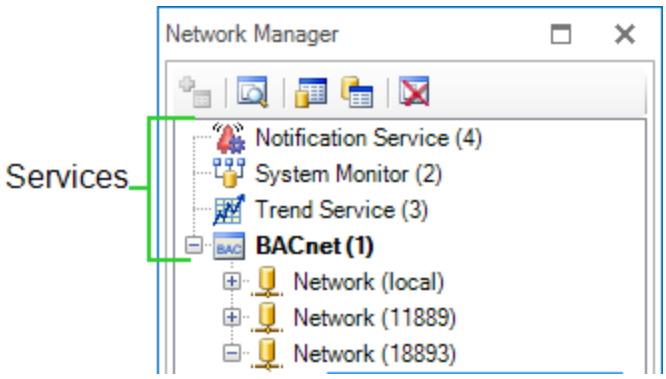
Configuring an existing service

The installed KMC Connect services are at the top of the Network Manager list. If a service is in the Network Manager list, use the following procedure to make changes. The exact changes that can be made will depend upon the service.

To configure an installed service, do the following:

1. Open Network Manager.
2. Locate and select the service.
3. Right-click the service and then choose **Configure Service** from the shortcut menu. A configuration tab opens in the workspace.
4. Make the changes to the service.
5. When changes are complete, click **Save Changes to Service** in configuration tab.

Illustration 4–4 Services in the Network Manager list



Related topics

- *Using Network Manager* on page 37
- *Network Manager toolbar commands* on page 38
- *Opening and closing the Network Manager pane* on page 38

Changing the Network Manager options

To change the way Network Manager displays objects and points in the list, change the settings in the Network Manager options.

1. On the ribbon click the **File** tab and then click **Options**.
2. From the **Component** list choose **Network Manager** and then change any of the following properties.

Display Mnemonics first Select or clear this check box to change the listing of a point or object in the Network Manager list.

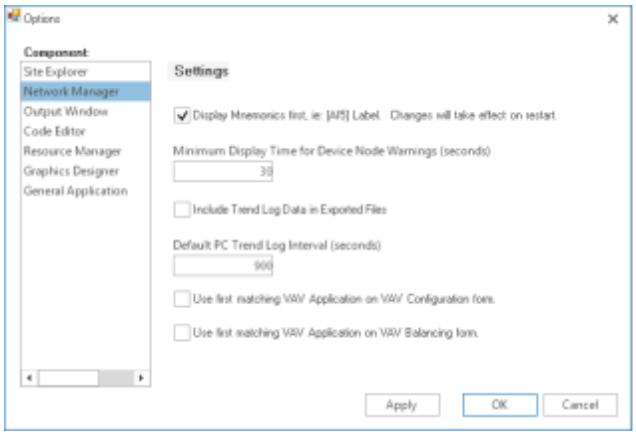
- When selected, displays the point or object's mnemonic identifier first in the Network Manager list.
- When cleared, displays the point or object's name or label first in the Network Manager list.

Minimum Display Time for Device Node Warnings (seconds) Sets the minimum period—in seconds—that warning icons and tool tips remain in place over icons in the Network Manager list.

Include Trend Log Data in Exported Files When this check box is selected the trend log data is included with a trend log object or point when the point or object is moved to the Resource Manager list.

Default PC Trend Log Interval (seconds) Sets the default logging interval value in the Trend Manager when setting up PC trends. See *Configuring a BACnet PC trend log* on page 96.

Illustration 4–5 Network Manager options dialog



Related topics

- *Using Network Manager* on page 37
- *Network Manager toolbar commands* on page 38
- *Opening and closing the Network Manager pane* on page 38

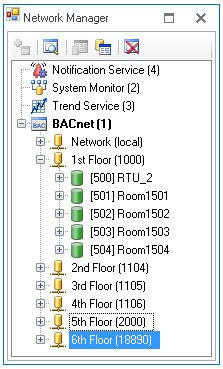
Naming BACnet networks

BACnet networks are identified by a network number assigned by a BACnet router. In KMC Connect, networks can be assigned a name for easy identification.

To rename a network, do the following:

1. Right-click the network icon and choose **Rename** from the shortcut menu.
2. Enter a descriptive name for the network.
3. Click **Okay** when finished.

Illustration 4–6 Network Manager with network names



Related topics

- *Using Network Manager* on page 37
- *Network Manager toolbar commands* on page 38
- *Opening and closing the Network Manager pane* on page 38

Restarting BACstac from the Network Manager

For various reason, the BACstac BACnet driver may need to be restarted after computer maintenance or upgrade. The driver can be restarted from the BACnet service in the Network Manager.

To restart the BACstac driver from the Network Manager, do the following.

1. In the Network manager, locate the BACnet service icon.
2. Right-click the icon and choose **Restart BACstac** from the shortcut menu.

Related topics

- *Using Network Manager* on page 37
- *Network Manager toolbar commands* on page 38
- *Opening and closing the Network Manager pane* on page 38

Section 5: Output Window

The Output Window is a continuously updated list of significant actions that have taken place in a KMC Connect managed site.

As components of Design Studio perform tasks, the success or failure of the function is indicated by a message in the Output Window. The information listed is tagged as either an error, warning or message.

Information listed in the Output Window is also saved to `.log` file in the Log folder of the Resource Manager.

Table 5-1 Output window buttons

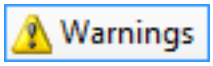
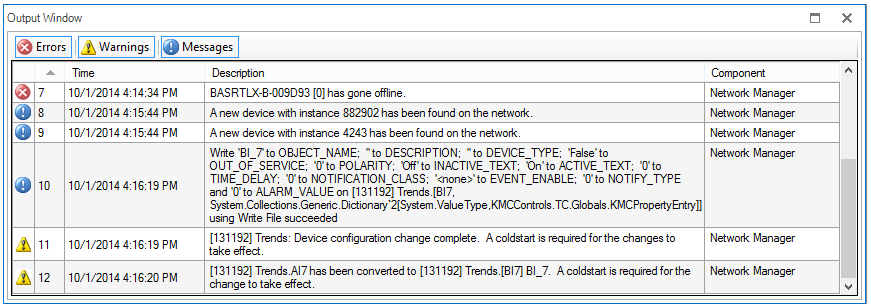
Button	Condition
	A severe condition that could be either system or user related.
	Less severe than an error; operation may proceed.
	For information purposes.

Illustration 5–1 Output window



At the top of the Output Window are three buttons. Clicking a button hides or reveals all of the messages associated with that button.

Related topics

- *Sorting the list in Output Window* on page 44
- *Opening and closing the Output window* on page 43

Opening and closing the Output window

To open or close the Output window, on the ribbon click the **View** tab and then click **Output Window**.

Sorting the list in Output Window

To change the order or content of the Output Window list, right-click on the list and select one of the following commands from the shortcut menu.

Clear Removes all messages from the list.

Sort By Sorts the list and displays the messages based on either message number, time of the message or the KMC Connect component from which the message originated.

Show By Hides or reveals the message by the component from which the message originated.

Output window options

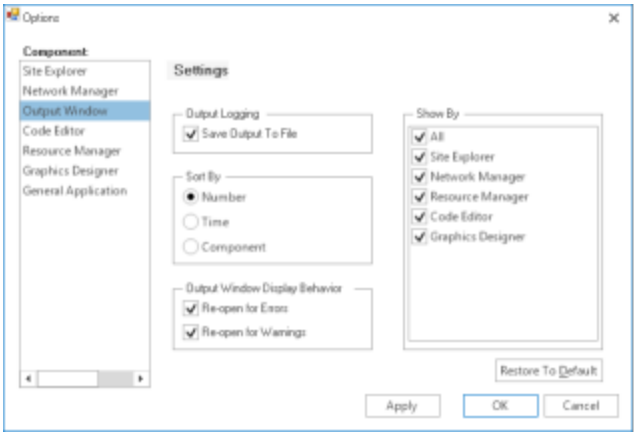
Permanently change the sorting or contents of the Output Window list with settings in the Output Window Options dialog.

To change the Output Window sorting, do the following:

1. On the ribbon click the **File** tab and click **Options**.
2. Click **Output Window**.
3. Change any of the following settings.
   - Select or clear the **Save Output To File** check box.
   - Change the path to the output log file.
   - Select the **Sort By** method.
   - Select or clear the **Show By** check boxes.

Click **Apply** or **Ok** when finished.

Illustration 5–2 Output Window Options dialog



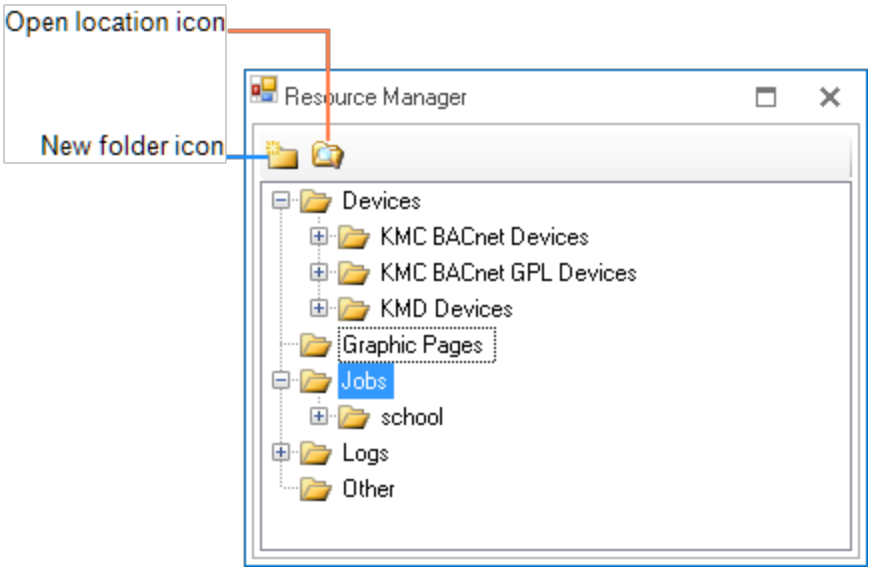
Section 6: About Resource Manager

Resource Manager is a tool with which operators can manage local files for backing up the building automation system and editing files offline to prepare templates. The following topics include procedures to use Resource Manager to create backup files, restore the configuration of controllers with backup files, and edit files offline.

Resource Manager displays and manages a list of files stored on the computer that is running KMC Connect. Through Resource Manager, these local files can be edited without connecting to a site. By opening an item in the Resource Manager list, you can:

- Create backup files of properties, points, devices, or the entire building automation system
- Restore device configurations with backup files
- Write Control Basic programs
- Open other files with their associated Windows application
- Import and export files from other programs.

Illustration 6–1 Resource Manager



Related topics

- *Opening and closing the Resource Manager pane* on page 47
- *Editing items in the Resource Manager list* on page 47
- *Restoring a configuration* on page 48
- *Creating backup files* on page 51

- *Managing the Resource Manager list* on page 51
- *Adding files to the Resource Manager folder* on page 52
- *Importing and exporting BACstage .bac files* on page 310

Opening and closing the Resource Manager pane

To open or close the Resource Manager, on the ribbon click the **View** tab and then click **Output Window**.

Related topics

- *About Resource Manager* on page 45
- *Editing items in the Resource Manager list* on page 47
- *Restoring a configuration* on page 48
- *Creating backup files* on page 51
- *Managing the Resource Manager list* on page 51
- *Adding files to the Resource Manager folder* on page 52
- *Importing and exporting BACstage .bac files* on page 310

Editing items in the Resource Manager list

You may edit offline any item in the Resource Manager list. To edit an item in the list:

1. Click ⊞ or ⊟ to expand or collapse the Resource Manager list to locate a device, object, point, or diagram file.
2. Use one of the following methods to open the item:
   - Right-click and choose **Configure**
   - Double-click the item
3. The configuration tool associated with the item opens in the workspace.
   - For Control Basic programs, the Code Editor opens.
   - For BACnet objects and devices, a BACnet configuration tab opens.
   - For items that are not part of KMC Connect, the Windows program associated with the file type opens.
4. Make changes and then save the configuration page.

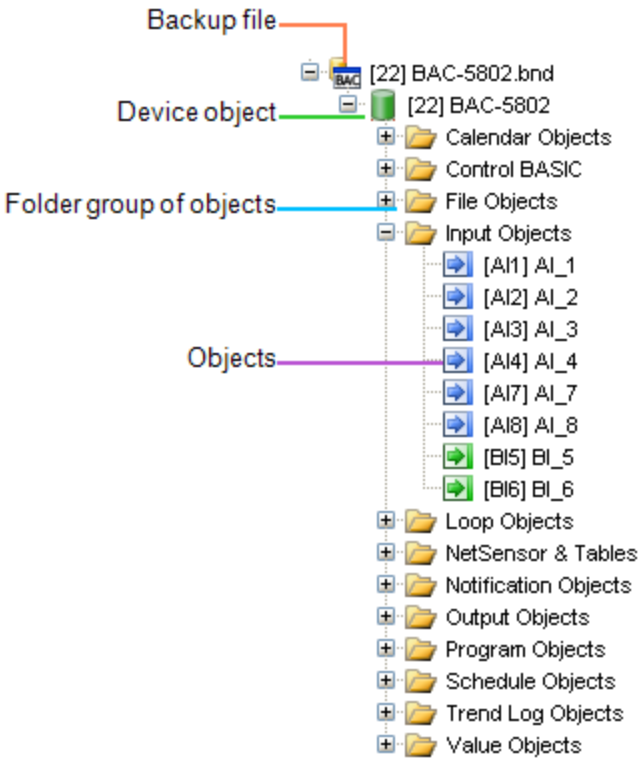
Related topics

- *About Resource Manager* on page 45
- *Opening and closing the Resource Manager pane* on page 47
- *Restoring a configuration* on page 48
- *Creating backup files* on page 51
- *Managing the Resource Manager list* on page 51
- *Adding files to the Resource Manager folder* on page 52
- *Importing and exporting BACstage .bac files* on page 310

Restoring a configuration

Dragging an object from a backup file in the Resource Manager list and to a compatible item in the Network Manager list is a ***restore*** process.

Illustration 6–2 Devices and objects in a .bnd file



When dragging a file to an item in Network Manager, the following actions are permitted:

Dragging objects to objects, devices or networks

- Drag an object from a backup file onto an object of similar type in the Network Manager list. For example, drag only input objects onto input objects.
- Drag an object from a backup file onto a device that contains objects of similar type. A dialog opens with which you can select the objects to restore.
- Drag an object from a backup file onto a network that includes devices with similar objects.
- Drag a group of objects, such as the input objects folder, onto a similar group. The objects in the backup file are matched by object instance number to the objects in the Network Manager list.

Dragging devices to devices or networks

- Drag a device from a backup file onto a device of the same model and manufacturer. For example, only BAC–5801 files may be dropped onto BAC-5801 devices in the Network Manager list. A dialog opens with which you can choose the objects within

the device to restore.

- Drag a device onto a network that includes similar devices. A dialog opens with which you can select the devices to restore.

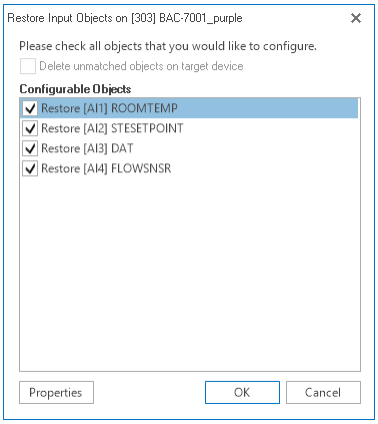
Dragging a folder to another folder or device

When dragging a folder to another folder or device, Resource Manager restores objects on an instance matching basis. For example:

- If the source folder contains input objects numbers 1-4, then objects 1-4 in the target folder or devices are restored.
- If the source item has 16 objects and the target item has eight objects, only eight target objects are restored with objects 1-8.

When the folder is dropped, the **Select Objects** dialog opens. Select or clear the check boxes next to the object names to designate the objects to restore.

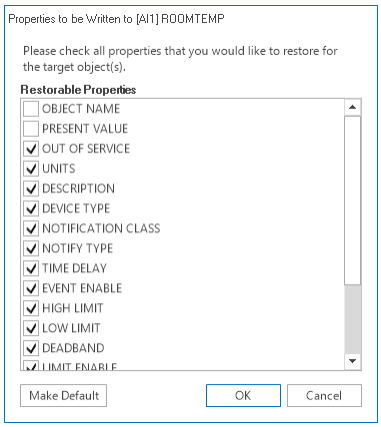
Illustration 6–3 Select Objects dialog



Tip: You may also right-click anywhere in the dialog and then select **Check All** or **Clear All** from the shortcut menu.

From the Select Objects dialog, click Properties to open the Properties dialog box. Select the properties to restore. Clicking **Make Default** saves the settings for the next time a device is restored.

Illustration 6–4 Select properties dialog

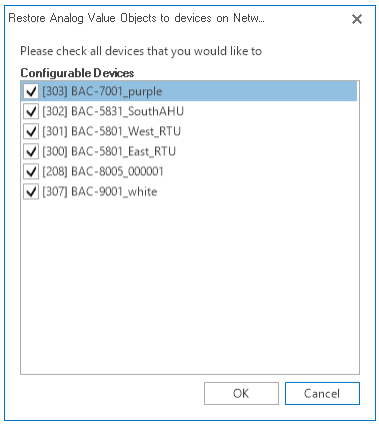


Tip: You may also right-click anywhere in the dialog and then select **Check All** or **Clear All** from the shortcut menu.

Dragging folders to networks

When dragging a folder group to a network, a dialog opens listing the devices on the network that contains compatible objects. Select the check boxes of the devices to restore.

Illustration 6–5 Select Devices dialog



Tip: You may also right-click anywhere in the dialog and then select **Check All** or **Clear All** from the shortcut menu.

Related topics

- *About Resource Manager* on page 45
- *Opening and closing the Resource Manager pane* on page 47
- *Editing items in the Resource Manager list* on page 47
- *Restoring a configuration* on page 48
- *Managing the Resource Manager list* on page 51

- *Adding files to the Resource Manager folder* on page 52
- *Importing and exporting BACstage .bac files* on page 310

Creating backup files

Dragging an item from the Network Manager list to the Resource Manager creates a backup file.

- For BACnet devices and objects, a `.bnd` file is placed in the Resource Manager list.

Related topics

- *About Resource Manager* on page 45
- *Opening and closing the Resource Manager pane* on page 47
- *Editing items in the Resource Manager list* on page 47
- *Restoring a configuration* on page 48
- *Managing the Resource Manager list* on page 51
- *Adding files to the Resource Manager folder* on page 52
- *Importing and exporting BACstage .bac files* on page 310
- *Backing up and restoring BACnet devices* on page 309

Managing the Resource Manager list

The Resource Manager list displays the hierarchical structure of folders, files, and devices in the Resource folder. You can delete, copy, move, and rename any item in the list. For example, you can open a folder that contains a file you want to copy or move, and then drag the file to another folder.

Creating a new folder

Creating a new folder in the Resource Manager list creates also a new folder in the Resource folder on the computer hard drive. For details about folders, see *Adding files to the Resource Manager folder* on page 52.

1. Click the new folder icon in the Resource Manager toolbar.
2. Enter a name for the new folder.
3. Once the folder is in the list, you may drag it to new location.

Removing an item from the Resource Manager list

1. Right-click the item.
2. Choose **Delete** from the shortcut menu.

Changing the name of an item in the Resource Manager list

1. Right-click the item.

2. Choose **Rename** from the shortcut menu.
3. Enter the new name.

Duplicating an item in the Resource Manager list

1. Right-click on the item.
2. Choose **Clone** from the shortcut menu.
3. Enter a new name for the duplicated item.

If the cloned item is a BACnet device, you may choose also to select the **Auto Increment Device Instance** check box. This increases the device instance number to prevent duplicate instance numbers.

Related topics

- *About Resource Manager* on page 45
- *Opening and closing the Resource Manager pane* on page 47
- *Editing items in the Resource Manager list* on page 47
- *Restoring a configuration* on page 48
- *Creating backup files* on page 51
- *Adding files to the Resource Manager folder* on page 52
- *Importing and exporting BACstage .bac files* on page 310

Adding files to the Resource Manager folder

All files and folders in the Resource Manager list are stored in the Resource folder. The folder is located on the same computer on which KMC Connect is running. Use Windows Explorer to move files to the Resource folder.

To open the Resource folder do one of the following:

- Click the Open Location icon  to open the Resource folder in Windows Explorer.
- Use Windows Explorer to locate the file at `C:\ProgramData\KMC Controls\KMC Connect\Resource`.

Once the Resource folder is open in Windows Explorer, you may move files into or out of it as you would any other folder.

Related topics

- *About Resource Manager* on page 45
- *Opening and closing the Resource Manager pane* on page 47
- *Editing items in the Resource Manager list* on page 47
- *Restoring a configuration* on page 48
- *Creating backup files* on page 51

- *Managing the Resource Manager list* on page 51
- *Importing and exporting BACstage .bac files* on page 310

Section 7: Viewing objects with Table View

Table Views are a method to view all of the properties of a group of objects. This section describes the methods and procedures to use Table Views.

To view all or some of the properties of a group of objects use the Table View feature of Design Studio. Table View can be used two different ways.

- To view all of the objects in a group folder. See the topic *Viewing groups of similar objects* on page 56.
- A custom view with a mix of objects from one or more controllers, see the topic *Custom Table Views* on page 59.

Illustration 7–1 Table View of input points

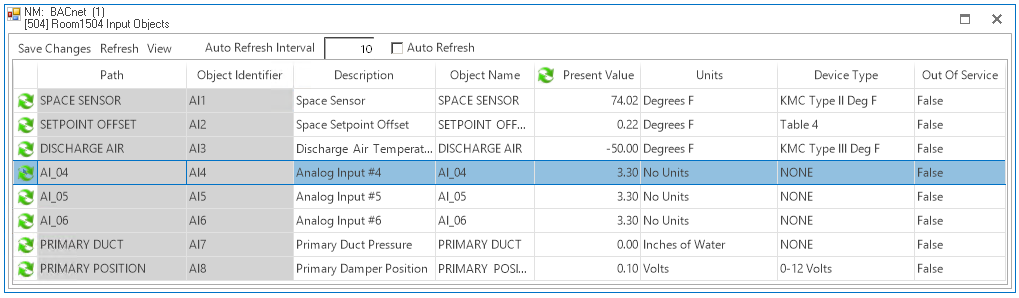
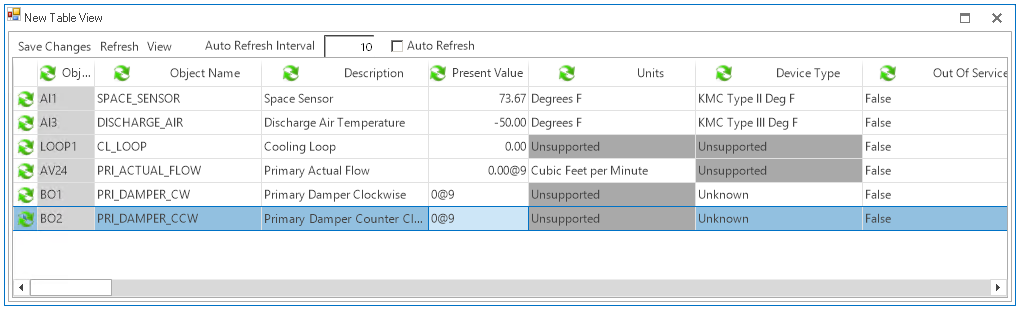


Illustration 7–2 Custom Table View



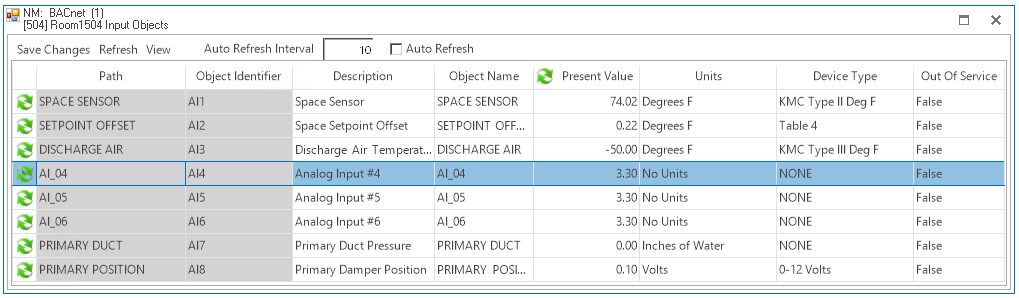
Viewing groups of similar objects

Opening a group of objects in the Table View is an alternative to using a device configuration tab. The Table View is opened from the Network Manager list and displays all of the properties for a group of objects in one Table. The view can be customized by rearranging properties, hiding or revealing properties, and adding other objects to the table.

To open the Table View, do the following:

1. In the Network Manager list, click ⊞ or ⊟ to expand or collapse the list of devices and objects to locate a folder of objects such as input objects.
2. To open the Table Viewer, do one of the following:
   - Double-click on the folder of objects. The Table Viewer opens.
   - Right-click the folder of points of objects and choose **Configure Objects** from the shortcut menu.

Illustration 7–3 Table View of input objects



Path	Object Identifier	Description	Object Name	Present Value	Units	Device Type	Out Of Service
SPACE SENSOR	AI1	Space Sensor	SPACE SENSOR	74.02	Degrees F	KMC Type II Deg F	False
SETPOINT OFFSET	AI2	Space Setpoint Offset	SETPOINT OFF...	0.22	Degrees F	Table 4	False
DISCHARGE AIR	AI3	Discharge Air Temperat...	DISCHARGE AIR	-50.00	Degrees F	KMC Type III Deg F	False
AI_04	AI4	Analog Input #4	AI_04	3.30	No Units	NONE	False
AI_05	AI5	Analog Input #5	AI_05	3.30	No Units	NONE	False
AI_06	AI6	Analog Input #6	AI_06	3.30	No Units	NONE	False
PRIMARY DUCT	AI7	Primary Duct Pressure	PRIMARY DUCT	0.00	Inches of Water	NONE	False
PRIMARY POSITION	AI8	Primary Damper Position	PRIMARY POSI...	0.10	Volts	0-12 Volts	False

Objects, located in the rows of the tables, and properties, located in the columns of the table, can moved or hidden to customize the view.

- To close the Table View, click the close button ✕ in the upper right corner of the page.
- To change the order of the columns, drag a column heading across the top of the table.
- To hide or reveal rows and columns, see the following procedures.
- Objects and properties with the refresh icon are selected for automatic update if Auto Refresh is selected.

Copying table data

The data in a Table View can be copied to the Clipboard and then pasted into other programs such as a spreadsheet, text editor, or word processor. The columns are separated with tabs.

1. Drag across the rows and columns.
2. Right-click the selected area and choose **Copy Selection to Clipboard**.
3. Open a document in an application such as spreadsheet, text editor, or word processor.
4. Paste the data into the document.

Hiding and revealing the objects and properties in the table

Use one of the following methods to hide or reveal rows or columns.

- To hide a row, right-click the first column in the row and select **Hide Object**from the shortcut menu.
- To hide a column, right-click the top of the column and select **Hide** from the shortcut menu.
- Choose one of the Select Visible commands from the View menu.

Hiding and revealing rows To hide or reveal hidden rows with the Select Visible Rows command, do the following.

1. From the **View** menu, choose **Select Visible Rows**.
2. When the Select Visible Object Rows dialog opens, do either of the following.
   - To hide a visible object, clear the check box next to the name of the object.
   - To reveal a hidden object, select the check box next to the name of the object.
3. Click **OK** when finished.
4. To make the change permanent, choose **Save** from the **View** menu.

Hiding and revealing columns To hide visible or reveal hidden columns with the Select Visible Columns command, do the following.

1. From the **View** menu, choose **Select Visible Columns**.
2. When the Select Visible Columns dialog opens, do either of the following.
   - To hide a visible property, clear the check box next to the name of the object.
   - To reveal a hidden property, select the check box next to the name of the object.
3. Click **OK** when finished.
4. To make the change permanent, from the **View** menu choose **Save**.

Refreshing the table data

Data in the table can be refreshed by clicking **Refresh** at the top of the tab. Objects and properties selected for automatically refresh are marked with the refresh icon  .

To automatically refresh the data, do the following:

1. Select the **Auto Refresh** check box.
2. Enter a time value in the **Auto Refresh Interval** text box. Time is entered in seconds; the minimum refresh interval is 1 second.

3 To choose specific objects (rows) and properties (columns) to refresh, do one or both of the following procedures.

- From the **View** menu, choose **Select Auto Refresh Rows**. When the auto refresh dialog opens, select or clear the check boxes next to the name of the row.
- From the **View** menu, choose **Select Auto Refresh Columns**. When the auto refresh dialog opens, select or clear the check boxes next to the name of the properties. After the dialog is closed, the column header turns to blue.

4 Click **Save Changes** when finished.

To add additional object types or properties from other devices

In addition to the objects from a single group of objects, objects from either the same device or other devices can be added to the viewer.

To add additional objects, do the following.

1 Open a Table View for a group of objects.

2 Locate the additional object types in the same device or the other device and objects in the Network Manager list.

3 Drag the additional objects or properties to the table.

Note: Other object types or objects from other devices are not saved in the Table View for a group of objects. To save a Table View with a mix of objects, see *Custom Table Views* on page 59.

Editing property configurations in the table viewer

Any properties that can be changed by KMC Connect can be changed in the table viewer.

1 In an open Table View, locate the property in the viewer.

2 Change the value of the property. For complex properties such as for editing Control Basic programs, a dialog will open. The value background changes to green to indicate unsaved changes.

3 When all changes have been made, click **Save Changes** at the top of the tab.

Click **Refresh** at the top of the table to undo changes that have not been saved.

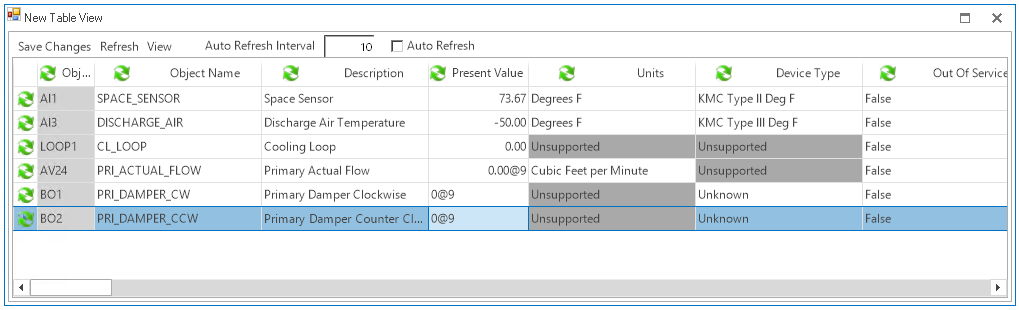
Related topics

- *Viewing objects with Table View* on page 55
- *Custom Table Views* on page 59
- *Using Network Manager* on page 37

Custom Table Views

Use a custom Table View to place a mix of object types from one or more controllers into a single Table View. The custom view can then be saved as a `.tbvx` file in the Resource Manager for future use.

Illustration 7–4 Custom Table View



Objects, located in the rows of the tables, and properties, located in the columns of the table, can moved or hidden to customize the view.

- To close the Table View, click the close button ✕ in the upper right corner of the page.
- To change the order of the columns, drag a column heading across the top of the table.
- To hide or reveal rows and columns, see the following procedures.
- Objects and properties with the refresh icon are selected for automatic update if Auto Refresh is selected.

Starting a new custom view

1. Do on of the following to open a custom view.
   - On the **View** tab, click **New Table View**.
   - On the ribbon click the **File** tab and then click **New** and then **New Table View**.
2. Locate objects or folders of objects in the Network Manager list.
3. Drag the objects or folders to the Table View workspace.
4. When all objects are in the new table, click **Save** or **Save As** to save the view as a `.tbvx` file in the Resource Manager.

Starting custom views from a group of objects

A custom Table View can also be started from a group of objects in the Network Manager list.

To start a custom Table View from a group of objects, do the following:

1. In the Network Manager list, click ⊞ or ⊟ to expand or collapse the list of devices and objects to locate a folder of objects such as input objects.

2 To open the Table Viewer, do one of the following:

- Double-click on the folder of objects. The Table Viewer opens.
- Right-click a folder of objects and choose **Configure Objects** from the shortcut menu.

3 As needed, drag other objects to the viewing area.

4 Click **Save As** to save the view as a `.tbvx` file in the Resource Manager.

Adding objects by path name

Objects can be added to a Table View by entering the path of objects that are located in either the Resource Manager or the Network Manager list. By using this method, multiple similar objects can be added at one time without dragging individual objects to the Table View.

1 From an open Table View, choose **View** and then **Edit Object Paths**. The paths dialog opens.

2 Click **Add** or **Edit** from the Object Paths dialog.

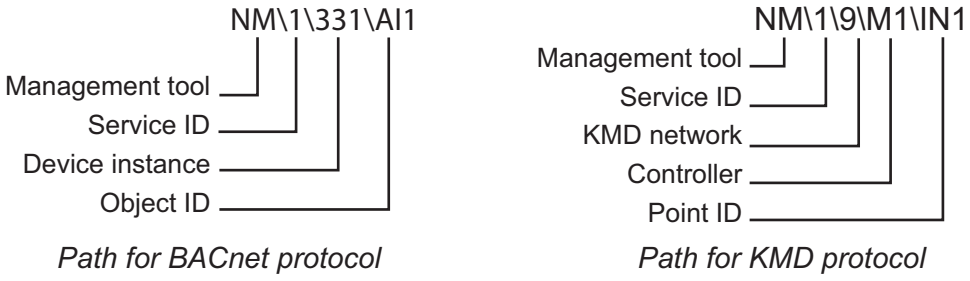
3 In the New Objects Path dialog, enter the path to the objects to place in the table.

- The parts of the path are separated by a backslash (\).
- Use ***NM*** if the objects are in the Network Manager list and ***RM*** for objects in the Resource Manager.
- A wildcard character (*) may be substituted for some of the characters in some parts of the path.
- BACnet and KMD paths are slightly different. See the illustration *Path names* for the parts of each path.

4 When the path is complete, click **OK** to close and save the New Object Paths dialog.

5 When all paths are added or edited, click **OK** to close the Object Paths dialog.

Illustration 7–5 Path names



Copying table data

The data in a Table View can be copied to the Clipboard and then pasted into other programs such as a spreadsheet, text editor, or word processor. The columns are separated with tabs.

1 Drag across the rows and columns.

2 Right-click the selected area and choose **Copy Selection to Clipboard**.

3 Open a document in an application such as spreadsheet, text editor, or word processor.

4 Paste the data into the document.

Hiding and revealing the objects and properties in the table

Use one of the following methods to hide or reveal rows or columns.

- To hide a row, right-click the first column in the row and select **Hide Object**from the shortcut menu.
- To hide a column, right-click the top of the column and select **Hide** from the shortcut menu.
- Choose one of the Select Visible commands from the View menu.

Hiding and revealing rows To hide or reveal hidden rows with the Select Visible Rows command, do the following.

1 From the **View** menu, choose **Select Visible Rows**.

2 When the Select Visible Object Rows dialog opens, do either of the following.

- To hide a visible object, clear the check box next to the name of the object.
- To reveal a hidden object, select the check box next to the name of the object.

3 Click **OK** when finished.

4 To make the change permanent, choose **Save** from the **View** menu.

Hiding and revealing columns To hide visible or reveal hidden columns with the Select Visible Columns command, do the following.

1 From the **View** menu, choose **Select Visible Columns**.

2 When the Select Visible Columns dialog opens, do either of the following.

- To hide a visible property, clear the check box next to the name of the object.
- To reveal a hidden property, select the check box next to the name of the object.

3 Click **OK** when finished.

4 To make the change permanent, from the **View** menu choose **Save**.

Refreshing the table data

Data in the table can be refreshed by clicking **Refresh** at the top of the tab. Objects and properties selected for automatically refresh are marked with the refresh icon .

To automatically refresh the data, do the following:

1 Select the **Auto Refresh** check box.

2 Enter a time value in the **Auto Refresh Interval** text box. Time is entered in seconds; the minimum refresh interval is 1 second.

3 To choose specific objects (rows) and properties (columns) to refresh, do one or both of the following procedures.

- From the **View** menu, choose **Select Auto Refresh Rows**. When the auto refresh dialog opens, select or clear the check boxes next to the name of the row.
- From the **View** menu, choose **Select Auto Refresh Columns**. When the auto refresh dialog opens, select or clear the check boxes next to the name of the properties. After the dialog is closed, the column header turns to blue.

4 Click **Save Changes** when finished.

Related topics

- *Viewing objects with Table View* on page 55
- *Viewing groups of similar objects* on page 56
- *Using Network Manager* on page 37

Section 8: Backing up and restoring a site

Use the Backup Site and Restore Site tools to save a site configuration and data stored on a computer running KMC Connect. This section describes procedures to back up the entire site and to restore the site with the backup files.

With KMC Connect you can back up site configuration and data files into one compressed file. A site backup file may then be taken to another computer. This type of site backup is often used to initially set up a site on one computer and then deploy the site onto a second computer.

Related topics

- *Backing up a site* on page 63
- *Restoring site data* on page 64

Backing up a site

A site backup copies all of the configuration and data files and stores them in a single compressed `.zip` file. Any of the following items may be saved during a site backup.

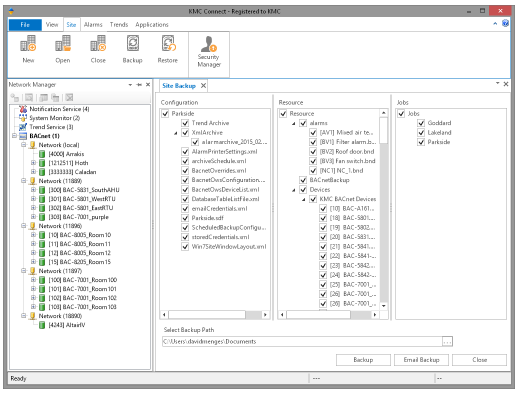
- Data from the site database
- Files stored in the Resource Manager
- Configuration job files

Even though `.zip` files are compressed, the backup file can take up several gigabytes of disk space. The file name is a combination of the site name and the date and time of the backup. Backing up a site may take several minutes to complete.

To perform a site backup, do the following:

1. On the **Site** tab click **Backup**.
2. Choose the items to be save in the backup file. To save time and make a smaller file, consider the following:
   - Alarm and trend data may not be relevant if the backup file will be used to move the site to a new computer.
   - The files under Resource Manager Files are not part of actual site operation. They can be copied anytime with Windows Explorer.
   - KMC Connect lists items in the Job folder separately but the folder is located inside the Resource folder.
3. Click **Backup** to start the backup process.

Illustration 8–1 Site backup settings



Select Backup Path The location in which KMC Connect will save the backup file. The default location is the My Documents folder for the current Windows user.

Configuration KMC Connect saves the items selected in the Configuration list in the backup file.

Resource The items in the Resource list are the folders and files in the Resource Manager. These files are not part of the site but may have been used to create the site.

Jobs Select the configuration jobs to save with the backup file.

Email Backup Send a previously saved backup file attached to an email message. The message is sent with the default email program running on the same computer as KMC Connect. The **From:** part of the message will have to be added before sending.

See the related topic *Restoring site data* on page 64.

Restoring site data

Restore data from either KMC Connect backup file with the KMC Connect restore site tool.

Note: When restoring a site, KMC Connect will not overwrite existing files.

To restore the data from a backup file, do the following:

1. On the **Site** tab click **Restore**.
2. Choose the backup file from the **Select Restore File** list. If the file is not in list, click the browse button … and locate the backup file.
3. Choose the items to restore from the backup file.
4. Click **Restore** when ready.

See the related topic *Backing up a site* on page 63.

Section 9: Security and passwords

This section covers procedures to manage user access, create security groups, and assign permissions from KMC Connect.

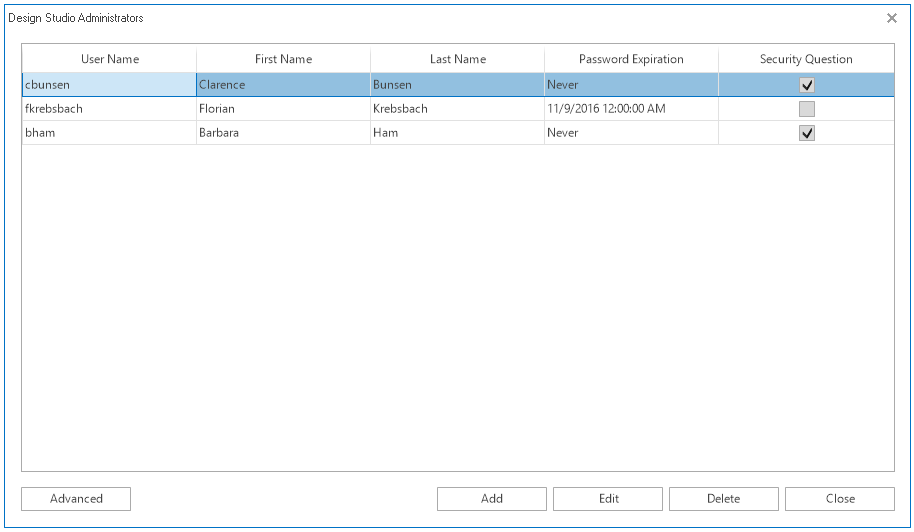
Master operators use the KMC Connect Security Manager to set up user access, passwords and optional security groups.

- With KMC Connect basic security, users are added as administrators. Basic security is what most users will use with KMC Connect.
- With the Security Manager, users are added to TotalControl security groups. This is useful if the KMC Connect site will be used to build a TotalControl site.

Topics covered in this section

- *KMC Connect basic security* on page 66
- *About the Security Manager* on page 68
- *Adding a new user to a group* on page 71
- *Changing a password* on page 72
- *Deleting a user* on page 72
- *Adding and modifying groups* on page 73

Illustration 9–1 KMC Connect basic security dialog



KMC Connect basic security

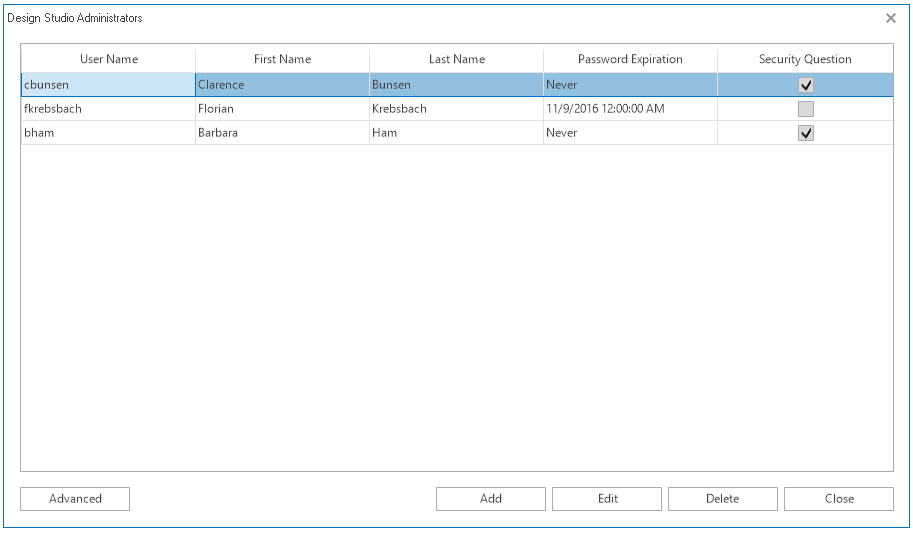
With KMC Connect basic security, users are added as administrators. Basic security is what most users will use with KMC Connect. This is the same level of security needed for TotalControl Design Studio.

With Advanced security, users are added to TotalControl security groups. This is useful if the KMC Connect site will be used to build a TotalControl site. See the topic *Using Security Manager in Design Studio*.

Add a user with basic security

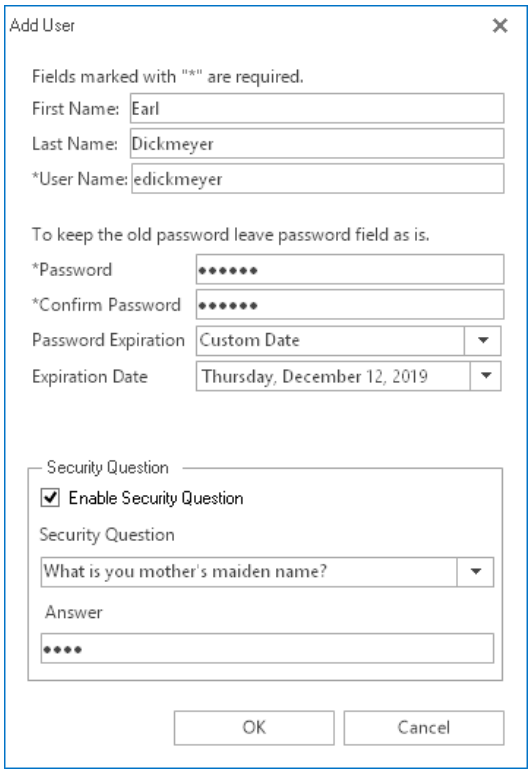
To add a new user, do the following:

1. On the **Site** tab click **Security Manager**.
2. Click **Add**.



3 In the Add User dialog, enter the following information:

- First and last name
- User name
- Password and confirmation password
- Password expiration date
- A security question and answer



4 Click **OK** when finished.

Deleting a user

To delete a user, do the following:

1 On the **Site** tab click **Security Manager**.

2 Select a user from the users list.

3 Click **Remove**.

4 Click **OK** when finished.

KMC Connect security features

First and Last Name Not required but helps to identify specific individuals with similar user names or to associate individuals with obscure user names.

Username The name by which a user will be identified for security purposes in KMC Connect.

Password A password is a secret word or string of characters that is used to authenticate the user.

Confirm Password The confirming password must match the password.

Password Expiration Passwords can be set to expire after a set period. When a password expires the user must enter the old password and establish a new password. The new password must be different than the old password. The default value is to never expire.

Expiration Date Sets the date the password will expire.

Security Question A security question helps to verify a user's identity when a password is lost. If the security question is answered correctly, the user can then establish a new password.

To use a security question, select the **Enable Security Question** check box.

Question Select from one of the several security questions in the question list.

Answer Enter the answer a user must enter correctly to the security question.

Advanced Click to open the TotalControl security dialog. See the topic *Using Security Manager in Design Studio*.

About the Security Manager

The TotalControl Security Manager is the tool with which users and security groups are managed with KMC Connect. Security groups are not used in KMC connect but may be used to set up a KMC Connect site that will be used to build a TotalControl site. The manager is composed of two parts, the User tab and Groups tab.

- The *User tab* manages all individual properties such as user names and passwords.
- The *Groups tab* establishes user groups that control access to various parts of the TotalControl web portal.

You can do the following with the Security Manager.

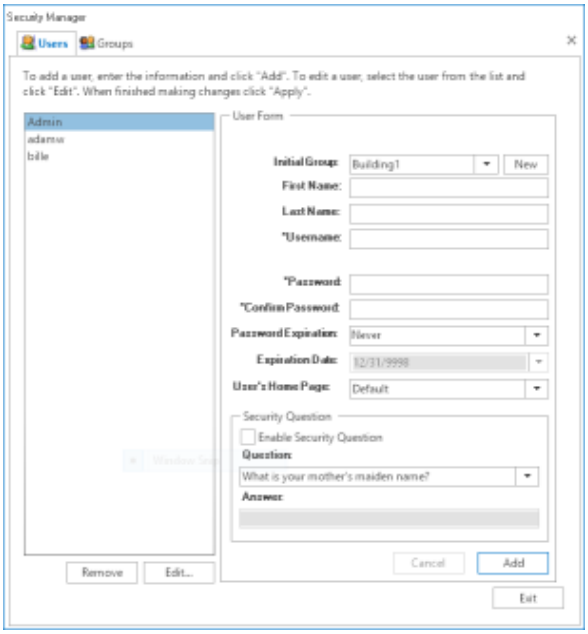
- Add a new operator name—*Adding a new user to a group* on page 71.
- Change an operator's password or security question— *Changing a password* on page 72.
- Remove an operator— *Deleting a user* on page 72.
- Add a security group or change group permissions— *Adding and modifying groups* on page 73.

User tab

Individual operator information is managed with the **User** tab.

- Names of new users are added.
- User passwords are established and changed.
- If required, a security question is added or changed.

Illustration 9–2 Security Manager Users tab



Initial Group All new users must be assigned to at least one group. Users can be added to the other groups after they are added to the users list.

First and Last Name Not required but helps to identify specific individuals with similar user names or to associate individuals with obscure user names.

Username The name by which a user will be identified for security purposes in KMC Connect.

Password A password is a secret word or string of characters that is used to authenticate the user.

Confirm Password The confirming password must match the password.

Password Expiration Passwords can be set to expire after a set period. When a password expires the user must enter the old password and establish a new password. The new password must be different than the old password. The default value is to never expire.

Expiration Date Sets the date the password will expire.

User's Home Page Sets the page in the TotalControl web portal to which the user will be directed after log in.Not applicable for KMC Connect.

Security Question A security question helps to verify a user's identity when a password is lost. If the security question is answered correctly, the user can then establish a new password.

To use a security question, select the **Enable Security Question** check box.

Question Select from one of the several security questions in the question list.

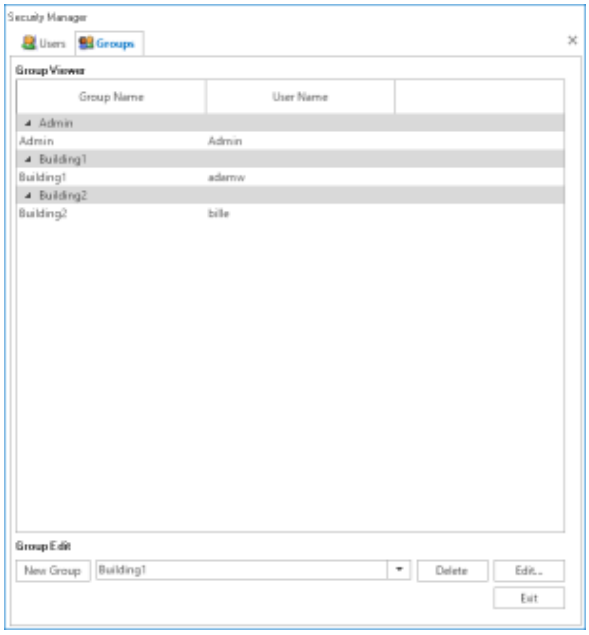
Answer Enter the answer a user must enter correctly to the security question.

Groups tab

The permissions and users in the group are managed with the **Groups** tab. Use the Groups tab to do the following:

- Add a user to or delete a user from a security groups.
- Grant or change permissions to an existing security group.
- Add new security groups.

Illustration 9–3 Security Manager Groups tab



Only operators assigned to a group with Admin or Administrator can connect to a site with KMC Connect.

Adding a new user to a group

Users can only be added to existing security groups. See the topic *Adding and modifying groups* on page 73 to add groups to a site.

Note: Users cannot be added to the user group Admin.

To add a new user to an existing group, do the following:

1. On the **Site** tab click **Security Manager**.
2. Choose the group for the user from the **Initial Group** list.
3. Under the Users tab, enter the following information:
   - First and last name
   - User name
   - Password and confirmation password
   - Password expiration date
   - A security question and answer
4. Click **Add** or **Exit** when finished.

Related topics

- To add or modify a group, see *Adding and modifying groups* on page 73.
- To modify a user's password or security question, see *Changing a password* on page 72.
- To entirely remove a user, see *Deleting a user* on page 72.
- For an overview of using Security Manager, see *About the Security Manager* on page 68.

Changing a password

To change a user password, do the following:

1. On the **Site** tab click **Security Manager**.
2. Select a user from the users list under the **Users** tab.
3. Click **Edit**.
4. Enter a new password in **Password** and **Confirm Password**.
5. Click **Ok** or **Apply** when finished.

Related topics

- To add or modify a group, see *Adding and modifying groups* on page 73.
- To add a new operator name, see *Adding a new user to a group* on page 71.
- To entirely remove a user, see *Deleting a user* on page 72.
- For an overview of using Security Manager, see *About the Security Manager* on page 68.

Deleting a user

Deleting a user removes the user name from all groups in KMC Connect.

Note: You cannot delete the user **Admin.**

To delete a user, do the following:

1. On the **Site** tab click **Security Manager**.
2. Select a user from the users list under the **Users** tab.
3. Click **Remove**.
4. Click **Yes** to delete the user and **No** to cancel.

Related topics

- To add or modify a group, see *Adding and modifying groups* on page 73.
- To add a new operator name, see *Adding a new user to a group* on page 71.
- To modify a user's password or security question, see *Changing a password* on page 72.
- For an overview of using Security Manager, see *About the Security Manager* on page 68.

Adding and modifying groups

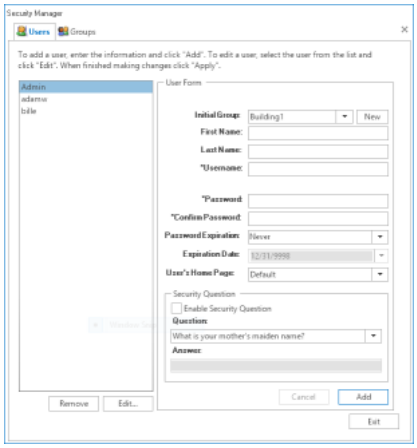
Create new security user groups and configure existing groups from the Create/Edit dialog.

- A security user group consists of one or more users.
- Permissions and roles are assigned to groups.
- Users are assigned to one or more groups.
- Users can perform tasks based upon the groups to which they belong.
- A user's permissions is the sum of the permissions granted to all of the groups to which the user is a member.

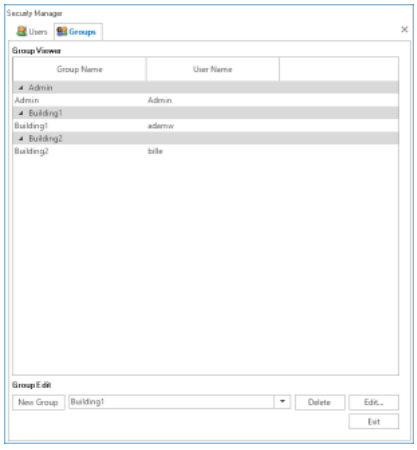
Note: All KMC Connect user groups must be assigned as an Administrator.

To add a new group, do the following.

1 On the **Site** tab click **Security Manager**.



2 Click the **Groups** tab.



3 Click **New Group**.

- Enter a name for the group in **Group Name**.
- In Group Roll, choose **Administrator**.

4 In the **Create\Edit Groups** dialog enter the following information.

a. Select a user name from the **Users** list.

b. Click >> to transfer the name to the **Belong to Group** list.

5 Add at least one user name to the group.

6 When finished, do one of the following.

- Click **Add** and then continue to add or modify groups.
- Click **Done** to save changes and close the dialog.

To modify an existing group, do the following.

1 On the **Site** tab click **Security Manager**.

2 Click the **Groups** tab.

3 Choose a group name from the **Group Edit** list.

4 Click **Edit**. The **Create\Edit Groups** opens.

5 Add or delete a user or modify **Group Role** or **Module Permissions**.

6 When finished, do one of the following.

- Click **Add** and then continue to add or modify groups.
- Click **Done** to save changes and close the dialog.

Related topics

- To add a new operator name, see *Adding a new user to a group* on page 71.
- To modify a user's password or security question, see *Changing a password* on page 72.
- To entirely remove a user, see *Deleting a user* on page 72.
- For an overview of using Security Manager, see *About the Security Manager* on page 68.

Section 10: The KMC Connect Audit Log

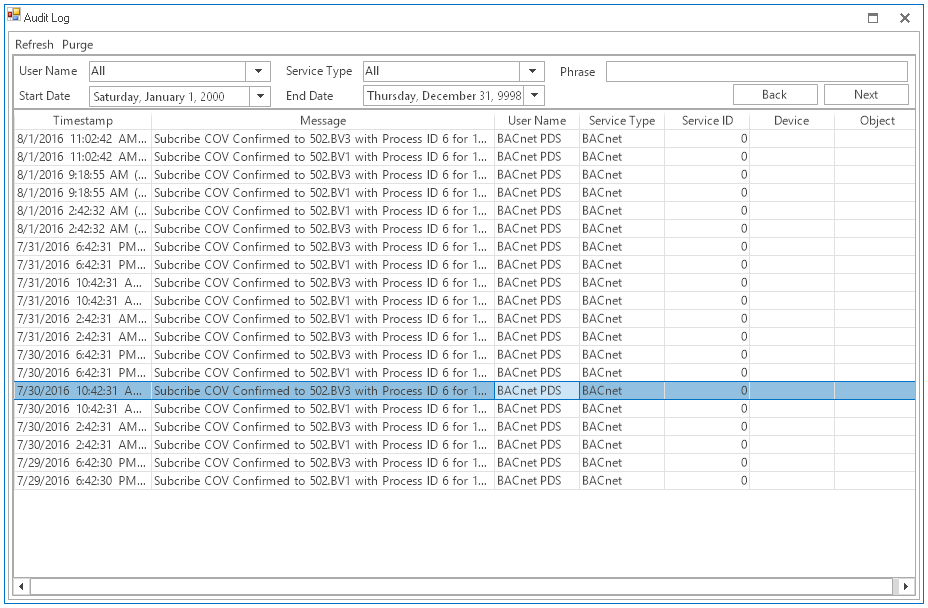
KMC Connect includes an Audit Log that tracks changes to the site.

The Audit Log viewer lists operational changes to the system in a list that can be filtered by any of the following criteria.

- User Name—Select from the list of authorized users
- Service Type—Changes made to BACnet or KMD controllers, or any of the building services
- Start and End Dates—The time and date when the change was made
- Phrase—A simple text search.

Note: Changes made to controllers by operator workstations other than the current instance of KMC Connect are not recorded.

Illustration 10–1 KMC Connect audit log



Viewing the Audit log

To view the Audit log, do the following:

1. On the ribbon choose **Audit Log** from the **Site** group. The Audit Log viewer opens.
2. From the **Search Category** list, choose the criteria for the search.
3. Click **Refresh**.

Purging old records

To remove records that are no longer useful, purge the Audit Log of old records.

To purge old records, do the following:

1. Open the Audit log for viewing.
2. At the top of the log click **Purge**.
3. In the Purge Audit Log dialog, set the time and date of the oldest record to be retained.
4. Click **OK** and then **YES** to confirm the purge. Clicking **NO** leaves the log untouched.

Section 11: Updating firmware

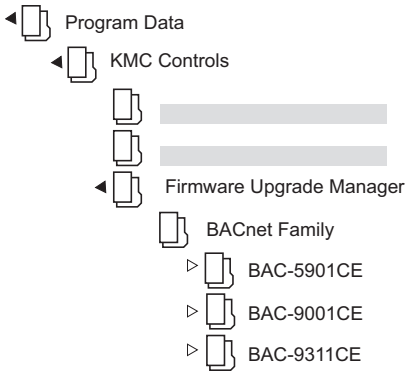
This section covers updating controller firmware over the network.

Later model Conquest series controllers can have their firmware updated—or FLASHed—over the network. Refer to the release notes for each controller to verify compatibility.

Before the firmware can updated, you will need a firmware file. Firmware files are available from the download section of our partner website at partners.kmccontrols.com. You will need a user name and password to log in. Each model for update will require a firmware file.

Downloading firmware

1. Download the firmware from the KMC Controls partners site. The firmware is packaged in a `.zip` file.
2. In the .zip file are two folders. Extract the file from the "For over the network" folder. The extracted file is self-running file that will install the firmware.
3. Double-click the extracted file. It will install the firmware at `C:\ProgramData\KMC Controls\Firmware Upgrade Manager\BACnet Family`. The file is placed in a folder with the model name for the firmware.



Updating firmware

1. Locate the device, network, or service icon in the Network Manager list.
2. Right-click the device, network , or service icon and choose **Update Firmware** from the shortcut menu.
3. Once the Firmware Upgrade Manager opens, drag any additional devices, networks, or services to the device list of the Firmware Upgrade Manager.
4. In the Firmware Upgrade Manager, click **Browse** and locate the new firmware file. The default location for the firmware is `C:\ProgramData\KMC Controls\Firmware Upgrade Manager\BACnet Family`. The firmware will be in a folder with the model name for the controller.

5 Click Update. KMC Connect will begin loading the new firmware into the selected controllers.

6 Once the new firmware is loaded, the Firmware Update Confirmation dialog opens.

- To finish updating, select the devices and click **Commit**.
- Click **Abort** to cancel the update and leave the devices with the original firmware intact.

Reducing errors

To reduce problems with upgrading device firmware over a network, reduce the volume of network traffic that is competing for bandwidth with the upgrade traffic by doing the following:

- Temporarily disable or remove devices such as J.A.C.E. controllers that continuously poll the network.
- Reduce the number of controllers that are selected for update at the same time.
- Temporarily isolate MS/TP networks and connect directly to the network with a BAC-5051E router.
- Verify that Control Basic is correctly managing the interval for reading from and writing to off-panel objects with WAIT, ALIAS, or NETPOINT functions.

The values at the bottom of the Firmware Upgrade Manager can be adjusted to increase the efficiency of the software upgrade.

Broadcast Interval The interval at which KMC Connect sends data. This can be set as low as 50 ms but at the expense of other network traffic. The default is 250 ms. Increasing this value will slow down the firmware upgrade but, it will keep the firmware upgrade from interfering with traffic on busy networks.

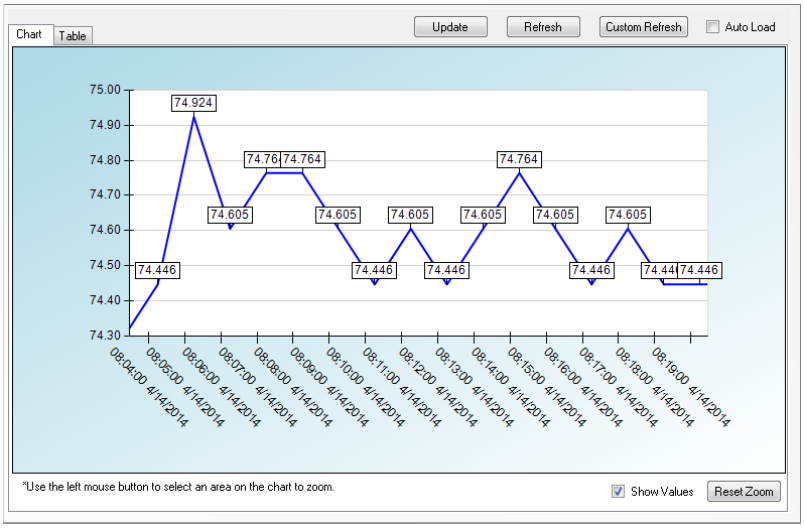
Response Timeout Typically changing this value is not required. The default value is 30 seconds. If there are numerous timeout messages in the output window, increase this value to give busy devices additional time to handle and respond to firmware upgrade requests.

Payload Size This property approximately corresponds to the BACnet network's maximum APDU size. For MS/TP networks, set this property to 450 or less depending on equipment constraints. For Ethernet and BACnet IP without MS/TP networks this can be set as high as 1400 but may still result in rejected packets.

Retries The number of times KMC Connect will attempt to send any one packet before the target device is dropped from the upgrade session. Retries slow down the upgrade session for the entire network.

KMC Connect

Part III: Trend logs



Section 12: About trend logs

Collecting data from a building automation system and saving it as a historical trend log is part of any building automation system. Topics in this section are an overview of trend logs in KMC Connect.

Trend logs are a record of the value of a point or property as measured by a building automation device or controller. KMC Connect collects data with one of three types of trends.

Controller trends

Controller trend logs are sets of historical data collected and stored only within the memory of an individual controller. Each controller has only a finite amount of allocated memory — also referred to as a buffer—in which it can store trend data. As the buffer nears capacity, one of three actions can take place:

- The controller continues to collect data until the buffer is full. It then stops collecting data.
- The controller continues to collect data by replacing the oldest data in the buffer with the newest collected data. This is often referred to as a circular buffer.
- The controller continues to collect data and—at a predetermined point—notifies a trend handling service to store the existing data. This makes room in the controller for newer data. In KMC Connect this is referred to as a device based trend log.

Device trends

Device trend logs are controller trend logs that have been retrieved from a device—either BACnet or KMD—and then stored by KMC Connect. As a controller trend log reaches a preset threshold, KMC Connect retrieves the data from the controller and then appends it to previously stored data. Threshold criteria depends upon the type of controller but, it is based upon one of the following:

- A preset time of day such as 1:00 AM.
- A preset interval such as every 15 minutes or every 3 hours.
- The number of samples in the log buffer.

PC trends

A PC trend log is a polling process that is performed by KMC Connect. On regular intervals, the value of a property or point is sampled by KMC Connect and then stored on the local computer.

Setting up trend logs

To set up trend logs, see the following topics.

- *Trend configuration wizard* on page 90
- *Configuring BACnet controller trend logs* on page 92

Viewing trend logs

To view trend logs, see the following topics.

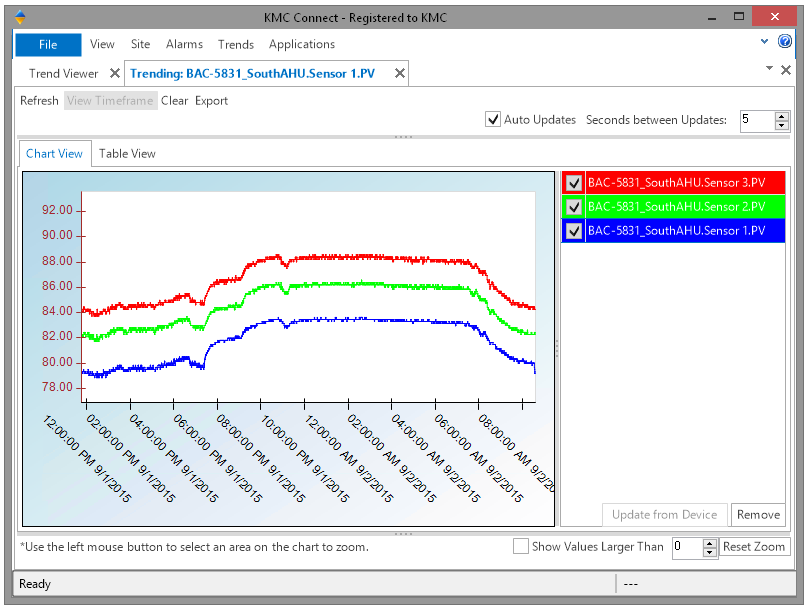
- *Viewing controller trend logs* on page 87
- *Viewing with the Trend Log viewer* on page 84

Section 13: Viewing trend logs in KMC Connect

Topics in this section cover various methods by which you can view trend logs. This section also covers exporting trend log data.

The data collected by KMC Connect as trend logs can be viewed by one of three methods.

- To view data in the BACnet trend log object, see *Viewing controller trend logs* on page 87.
- To view temporary data from one or more points or objects, see the topic *Viewing with the Trend Log viewer* on page 84 and set up a live trend log.
- To view device or PC trend logs, see the ***Trend Log viewer*** and then follow the procedure for *Viewing data from the Trend Manager*.

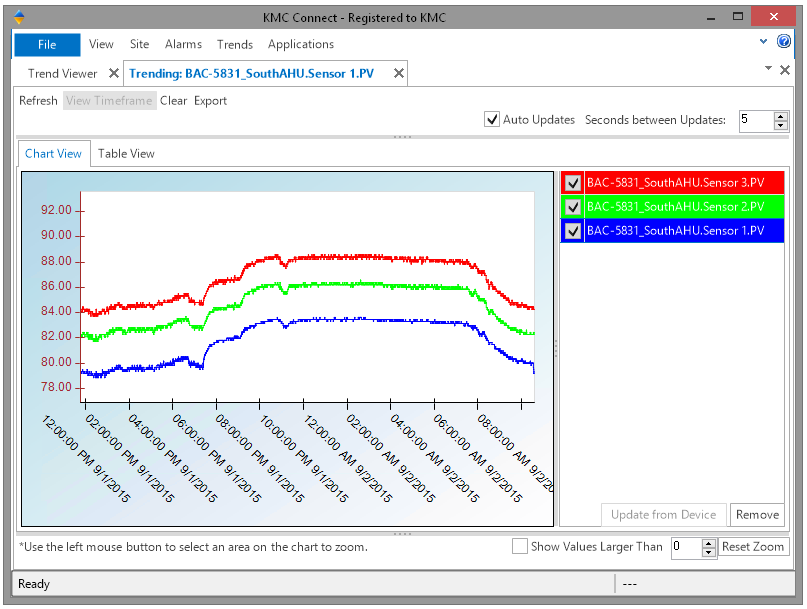


Viewing with the Trend Log viewer

The Trend Log viewer is a tool to view historical trend data from within KMC Connect. The viewer can be used for two types of data.

- To view temporary data from one or more points or objects, see the procedure *Starting a live trend log* on page 84.
- To view device trend logs, PC trend logs, or trend groups, see *Viewing data from the Trend Manager*.

Illustration 13–1 Trend Log viewer



Starting a live trend log

A live trend log view will start collecting trend data for viewing data as soon as an object is added to the Trend Viewer. The data in the live trend log chart is temporary and is not stored by KMC Connect.

1. On the ribbon choose **Trend Viewer** from the **Trends** group.
2. Locate an object to collect data from in the Network Manager list.
3. Drag the object from Network Manager list to the Trend Viewer.
   - The number of analog objects is limited only by viewing clarity.
   - If a mix of analog, binary or multistate objects are added to the same window, only 20 objects can be present.

Starting from the Network Manager

To start a live log from the Network Manager, do the following.

1. Locate an object to collect data from in the Network Manager list.

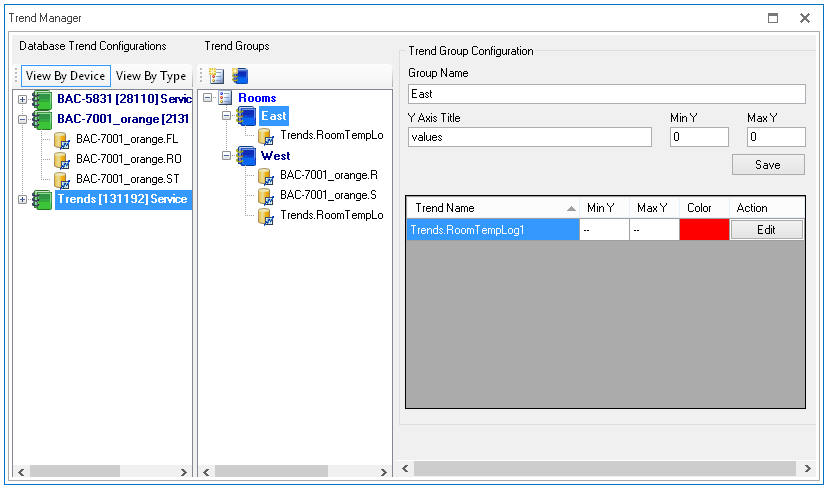
2 Right-click the object and select **New Live Trend** from the shortcut menu.

3 As needed drag additional objects from the Network Manager list to the Trend Viewer.

Viewing data from the Trend Manager

You can use the Trend Viewer to view either PC trends, device trends or trend groups from the Trend Manager.

Illustration 13–2 KMC Connect Trend Manager



1 On the ribbon choose **Manager** from the **Trends** group.

2 Do one of the following:

- Right-click on an PC trend object icon or Device Log icon in the Database Trend Configurations list and choose **Open Trend Viewer** from the shortcut menu.
- Right-click on a trend group icon in the Trend Groups list and choose **Open Trend Viewer** from the shortcut menu.

Parts of the Trend Viewer

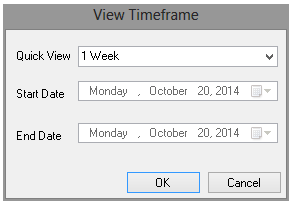
Values in the viewer can be displayed as either a chart or table.

- To change the view, click the **Chart View** or **Table View** tab.
- To hide an individual graph, clear the check box next to the name of the graph.

Refresh Click to manually update the chart or table with the newest data in the objects.

View Timeframe Choose a time frame for the display from the Quick View list or set a Start Date and End Date for a custom time frame.

Illustration 13–3 The Trend Viewer Timeframe dialog



Show Values Larger Than Select this check box to display the actual recorded values in text of the trend log. Enter a threshold for the minimum value to show as text.

Clear Click to remove all data from the window.

Zoom in/Zoom out In the Chart view, only a limited amount of trend data is displayed on the chart.

- To zoom in, drag from left to right across an area of the chart.
- To zoom out, drag from right to left across an area of the chart.

Reset Zoom Restores the display to the original view.

Export Click to export the data in the display to a comma separated value (`.csv`) file.

Related topics

- *Configuring BACnet trend logs and groups* on page 89
- *Viewing controller trend logs* on page 87

Viewing controller trend logs

KMC Connect includes a controller trend log object viewer for examining data collected by BACnet controller trend objects or KMD trend logs.

To view controller trend data from a point or object, do the following.

1. Open Network Manager.
2. In the Network Manager list locate the BACnet or KMD controller.
3. Within the controller locate and click on the trend log folder to reveal the list of trend object or point icons .
4. Double click a trend icon and then choose **View Trend** to open the trend log.

Device and PC trend logs are viewed with the Trend Viewer. See the topic *Viewing with the Trend Log viewer* on page 84.

BACnet controller trend logs can be displayed as either a chart or table. To change the view, click the **Chart** or **Table** tab.

Update Click to retrieve new data not already on the chart and add it to the chart data.

Refresh Clears data from the chart, retrieves the controller trend and plots it on the chart.

Custom Refresh Enter a specific number of records beginning at a designated time.

Autoload When selected, new trend data in the controller is added to the existing data in the trend viewer.

Show text Select this check box to display the actual recorded values in text of the trend log.

In the Chart view, only a limited amount of trend data is displayed on the chart.

- To zoom in, drag from left to right across an area of the chart.
- To zoom out, drag from right to left across an area of the chart.
- Click **Reset Zoom** to restore to the original view.

Illustration 13–4 BACnet controller trend object chart with text displayed

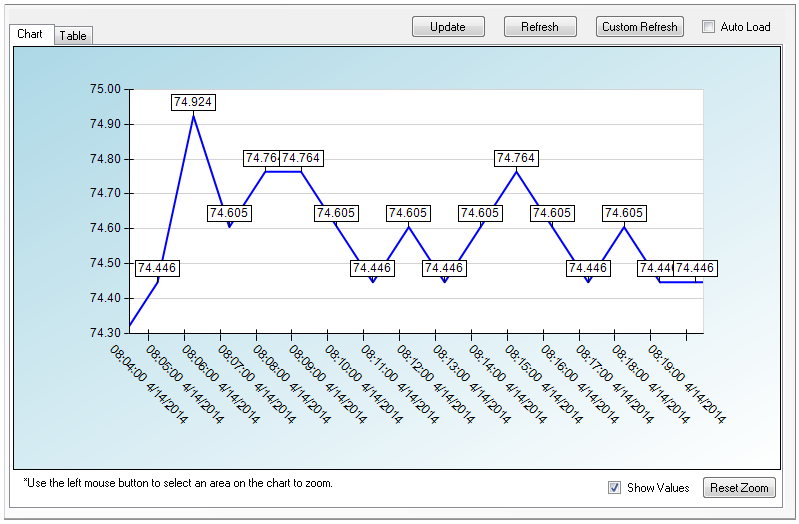
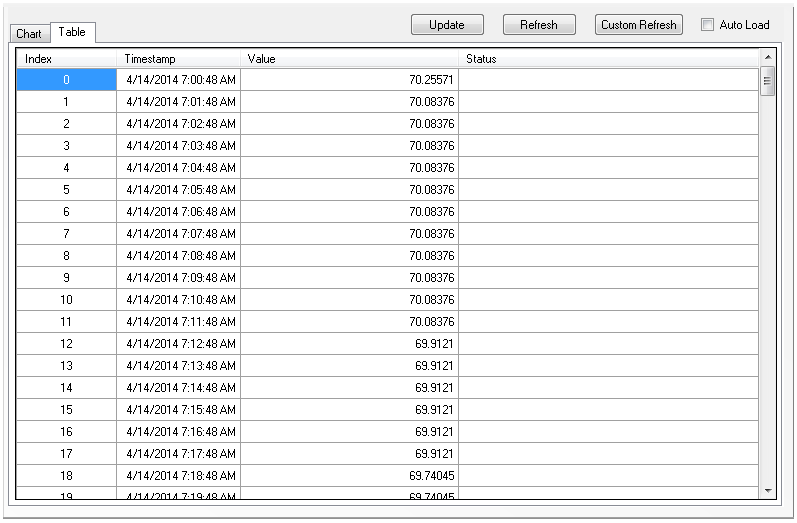


Illustration 13–5 BACnet controller trend object—table view



Index	Timestamp	Value	Status
0	4/14/2014 7:00:48 AM	70.25571	
1	4/14/2014 7:01:48 AM	70.08376	
2	4/14/2014 7:02:48 AM	70.08376	
3	4/14/2014 7:03:48 AM	70.08376	
4	4/14/2014 7:04:48 AM	70.08376	
5	4/14/2014 7:05:48 AM	70.08376	
6	4/14/2014 7:06:48 AM	70.08376	
7	4/14/2014 7:07:48 AM	70.08376	
8	4/14/2014 7:08:48 AM	70.08376	
9	4/14/2014 7:09:48 AM	70.08376	
10	4/14/2014 7:10:48 AM	70.08376	
11	4/14/2014 7:11:48 AM	70.08376	
12	4/14/2014 7:12:48 AM	69.9121	
13	4/14/2014 7:13:48 AM	69.9121	
14	4/14/2014 7:14:48 AM	69.9121	
15	4/14/2014 7:15:48 AM	69.9121	
16	4/14/2014 7:16:48 AM	69.9121	
17	4/14/2014 7:17:48 AM	69.9121	
18	4/14/2014 7:18:48 AM	69.74045	
19	4/14/2014 7:19:48 AM	69.74045	

Section 14: Configuring BACnet trend logs and groups

Each of the supported BACnet trend logs in KMC Connect have specific configuration requirements for proper operation. Topics in this section cover detailed procedures for configuring KMC Connect trend logs in BACnet controllers.

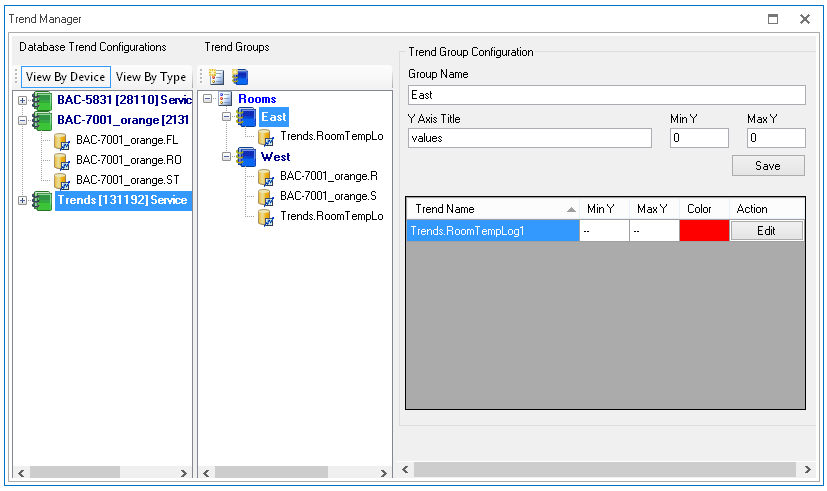
The easiest method for setting up device and controller trend logs is to use the Trend wizard. See *Trend configuration wizard* on page 90.

Trend logs can also be set up manually with any of the following procedures.

- *Configuring BACnet controller trend logs* on page 92
- *Configuring BACnet device trend logs* on page 94
- *Configuring a BACnet PC trend log* on page 96

To view trend data see, *Viewing trend logs in KMC Connect* on page 83.

Illustration 14–1 Trend Manager



Trend configuration wizard

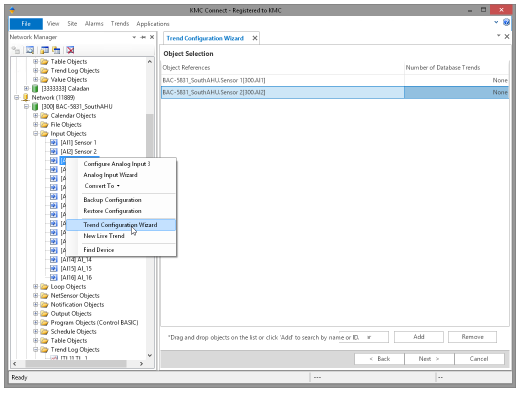
The Trend Wizard is an easy way to configure trend logs for BACnet controllers. Using the wizard is an alternative to manually setting up the properties for trend log, notification class, and event enrollment objects.

- Use the wizard to configure trends for input, accumulator, output, or value objects.
- For controllers that support Change Of Value subscriptions, the wizard can set up COV trends.
- If the object is in a controller without a Trend Log object, the wizard will set up a PC trend.

Start the wizard

1 Start the Trend Configuration wizard by doing one of the following:

- On the ribbon choose **Trend Configuration Wizard** from the **Trends** group.
- Right-click an object in the Network Manager list and choose **Trend Configuration Wizard** from the shortcut menu. This starts the wizard and adds the object to the Object Selection list.



2 Add objects to the Object Selection list by doing the following:

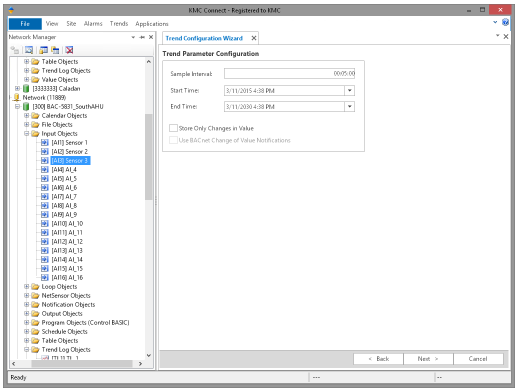
- Drag input, output, or value objects from Network Manager list to the Object Selection list.
- Right-click an object in the Network Manager list and choose **Trend Configuration Wizard**.
- Click **Add** and then search for objects. See the procedure *Add objects by searching* on page 91.

Note: If an object has been previously assigned, the number of previous trends is shown in the ***Number of Database Trends*** column.

3 When ready, click **Next**.

4 Change or accept the Trend Parameter Configuration properties.

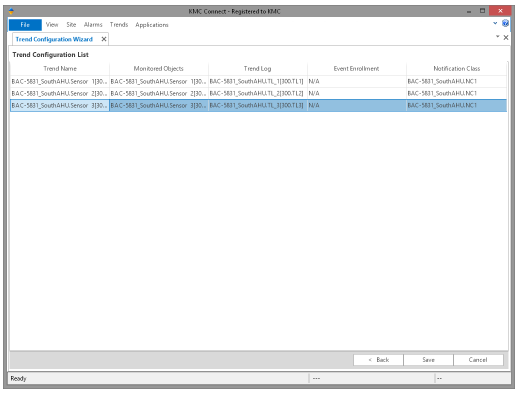
- Sample Interval sets the time between samples.
- **Start** and **End Time** sets the time span for logging data.
- Select the **BACnet Change of Value Notifications** check box to set up a COV subscription. COV subscriptions are not supported in all devices. For additional information, see the topic *Change Of Value (COV) notifications* on page 119.



5 Click **Next** when ready.

6 The Trend Configuration list shows the assignments of the Trend Log, Notification Class, and where applicable, Event Enrollment objects.

- Right-click on a cell to change assignments.
- The name of the trend in the first column can be changed.
- If a Trend Log object is not available, a PC trend is set up.



7 Click **Save** when finished.

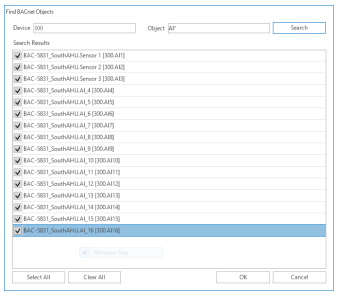
- All information in the list is written to the objects
- For viewing, the trend logs are also added to a the Trend Configuration Manager.

Add objects by searching

Objects in the Network Manager list can be added to the Trend Wizard Object Selection list by using the Search and Add function.

To add objects by searching, do the following:

1. Start the wizard.
2. In the Object Select list, click Add.
3. Enter the instance number of a device in the Network Manager list.
4. Enter the mnemonics for the object followed by the object number.
   - Use AI for Analog Input, BO for Binary Output objects, MSV for Multistate Value objects, etc.
   - Use an asterisk (*) for all objects of the same type. For example AI* for all analog input objects.
   - To search for all devices, use an asterisk (*) in the Device box.
   - Use a question mark (?) to substitute for characters when searching by name. For example ***SENSOR?*** will find both ***SENSOR1*** and ***SENSOR2*** but not ***SENSOR10***.
5. Click **Search**.
6. Select or clear the check boxes next to each object found by the search.
7. Click **OK** to add the selected objects to the list.



For details on the BACnet objects related to trends, see the following topics.

- *Trend Log object* on page 384
- *Notification object* on page 365
- *Event enrollment object* on page 329

To manually configure BACnet trend logs, see the following topics.

- *Configuring BACnet controller trend logs* on page 92
- *Configuring BACnet device trend logs* on page 94
- *Configuring a BACnet PC trend log* on page 96

Configuring BACnet controller trend logs

A controller trend log is data that is stored in a BACnet trend log object.

- To view the data in the trend log object, see *Viewing controller trend logs* on page 87.
- To view the data with the Trend Log viewer, add the controller trend to a trend group. See *Adding BACnet trend logs to trend groups* on page 98

- To set up trend logs with a wizards, see *Trend configuration wizard* on page 90.

To set up a BACnet controller trend log, do the following:

1. In the Network Manager list, locate the device and then select the trend object that will collect the trend data.
2. Expand **General Properties**.
3. Enter a name for the trend log in **Object Name**.
4. Enter additional information in **Description**.
5. Enter the start and stop dates and time in **Start Time** and **Stop Time**. The start and stop times are based on the time and date maintained within the controller.
6. Enter the frequency to sample the data in **Log Interval**. This is entered in hundredths of a second.
7. For **Buffer Size** enter 256 for KMC BACnet controllers. Entering zero (0) clears the buffer.
8. Enter the number of samples that will trigger the notification recipient that the buffer has available data. For device trend logs, a value of 10 provides good trade-off between responsiveness and excessive network traffic.
9. Under **Device** in **Log Object Property Reference**, do one of the following:
   - In the **Device** list, choose an object to log.
   - Drag an object from the Network Manager list to **Device**.
10. Under **Property** in **Log Object Property Reference**, if the property to log is not **Present Value**, choose the correct property.
11. Select the **Log Enable** check box.
    - Clear the **Stop When Full** check box to continuously collect data for device trend logs.
    - Select the **Stop When Full** check box to halt data collection when the buffer is full.
12. Select or clear the **Stop When Full** check box.
13. If a notification class object will be used with this trend log, expand **Event/Alarm Properties** and do the following:
14. In **Notification Class**, enter the object number of the notification class object.
15. Under **Event Enable** do the following.
    - Select the **To Normal** and **To Off Normal** check boxes
    - Clear the **To Fault** check box.
16. In **Notification Type**, select **Event.**
17. When all selections and entries are correct, click **Save Changes** at the top of the tab.

Related topics

- *About trend logs* on page 81

- *Viewing trend logs in KMC Connect* on page 83
- *Configuring BACnet device trend logs* on page 94
- *Configuring a BACnet PC trend log* on page 96
- *Adding BACnet trend logs to trend groups* on page 98

Configuring BACnet device trend logs

A device trend log is a controller trend log that is periodically retrieved from a controller and then stored by KMC Connect.

- To view the data with the Trend Log viewer, see *Adding BACnet trend logs to trend groups* on page 98
- To set up trend logs with a wizards, see *Trend configuration wizard* on page 90.

To configure a device trend log for a BACnet controller, do the following:

Configuring BACnet trend objects for a device trend log

1. Follow the procedure in the topic *Configuring BACnet controller trend logs* on page 92 to set up a controller trend log.
2. In the configuration tab for the trend log object that was configured for the controller trend, do the following:
   a. Expand **Event/Alarm Properties**.
   b. In **Notification Class**, enter the number of the notification class object.
   c. Under **Event Enable** do the following.
      - Select the **To Normal** and **To Off Normal** check boxes.
      - Clear the **To Fault** check box.
   d. In **Notification Type**, select **Event**.
   e. Save the changes.
3. In the same device in which you are configured the trend log object, open a notification class object.
4. Add to **Recipient List** the device instance number of the notification handling device. For KMC Connect, enter the device instance of the BACnet service. The KMC Connect device instance is found in the BACnet node the Network Manager list. Do one of the following to add a notification handling device.
   - Drag the icon of the notification handling device from the Network Manager list to the Recipient List text box.
   - Click **Add** and then enter the device instance in the Recipient Identifier dialog.
   - Holding CTRL and then dragging a device icon will also open the Recipient Identifier dialog.
5. Under **Ack. Required** clear all check boxes.

6 Select or clear the check boxes for the applicable days of the week and then set the time of day.

- The days of the week and time of notification are based upon the clock in the controller.
- For 24-hour notification, set the time from 12:00:00 AM to 11:59:00 PM.

7 In **Transitions** enter the following settings:

- Select the **To Normal** and **To Off Normal** check boxes.
- Clear the **To Fault** check box.

8 Enter any non-zero number for **Process ID**.

9 Select the **Send Confirmed Notifications** check box.

10 When selection and entries are correct, click **Save Changes** at the top of the tab.

11 Continue the set up by adding the trend log to the trend group.

Add the trend log to a trend group

1 Set up a BACnet object, trend object and notification class object as described by the previous steps.

2 On the ribbon choose **Trend Manager** from the **Trends** group.

3 Locate the trend object  in the Network Manager list.

4 Drag the trend log object from the **Network Manager** list to the **Database Trend Configuration** list in the Trend Manager. The Device Based Trends dialog opens. See *Using the Device Based Trend dialog* on page 95.

5 Enter the trend configuration settings in the dialog. Click **Submit** when finished.

6 Drag the trend object from the **Database Trend Configurations** list to a trend group in the **Trend Groups** list.

Using the Device Based Trend dialog

Use the Device Based Trend dialog to configure a BACnet device based trend. The dialog opens automatically when a BACnet object is dropped into the Database Trends Configuration list.

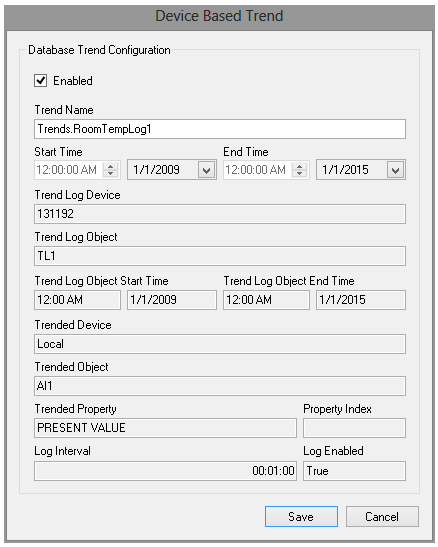
To open the dialog after the trend object is placed in the list, do the following:

1 Right-click an object in either the **Database Trend Configurations** or **Trend Groups** list.

2 Choose **Edit** from the shortcut menu.

3 Make changes in the open dialog and then click **Save** when done.

Illustration 14–2 BACnet device trend dialog



Enable Clear the check box to stop adding recorded data from this trend object to KMC Connect. Select the check box to add data from the trend object by KMC Connect.

Trend Name Enter the name of the trend as you want it to appear in the trend viewer. The default value for **Trend Name** is the object name. Changing Trend Name does not change the object name.

Configuring a BACnet PC trend log

A PC trend log is a polling process performed by KMC Connect. For a detailed overview on PC trends, see *About trend logs* on page 81.

- Any object except trend objects may be selected for a PC trend.
- An object may be placed in more than one trend group.
- To view the data with the Trend Log viewer, see *Adding BACnet trend logs to trend groups* on page 98
- To set up trend logs with a wizards, see *Trend configuration wizard* on page 90.

To set up a PC trend log for a BACnet object, do the following:

1. On the ribbon choose **Trend Manager** from the **Trends** group.
2. In the Network Manager list locate the object to trend.
3. Drag the object from the Network Manager list to the **Device Trend Configurations** list in the Trend Manager. The **PC Based Trends** dialog opens. See *Using the PC Based Trends dialog* on page 97.
4. Enter the trend configuration to the dialog. Click **Submit** when finished.
5. Drag the object from the **Device Trend Configurations** list to a trend group in the **Trend Groups** list.

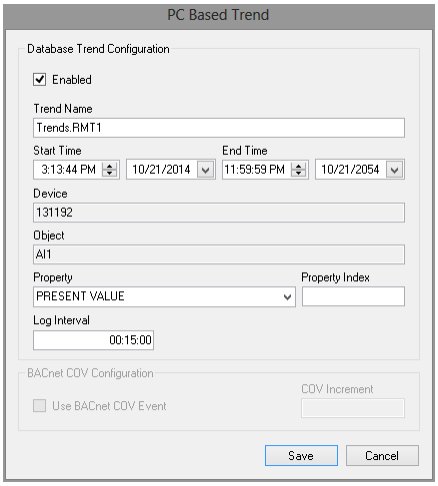
Using the PC Based Trends dialog

Use the PC Based Trend dialog to set up a PC based trend for a BACnet object. The dialog opens automatically when a BACnet object is dropped into the Database Trends Configuration list.

To open the dialog:

1. Right-click a PC trend icon in either the **Device Trend Configurations** or **Trend Groups** list.
2. Choose **Edit** from the shortcut menu.
3. Make changes in the open dialog and then click **Save** when done.

Illustration 14–3 PC Based Trends dialog



Enable Select the check box to add the records for an individual point from a KMD trend log to the Device Trend Configurations list; clear the check box to disable the individual point.

Trend Name Enter the name of the trend as you want it to appear in the trend viewer. The default value for **Trend Name** is the object name.

Start and Stop Date Enter the date and times to start and end data collection. The date and times are based on the clock in the computer hosting KMC Connect.

Property (Available only when dragging the object to the Device Trend Configurations list.) Select from the properties in the object. Typically the property is Present Value.

Property Array (Available only when dragging the object to the Device Trend Configurations list.) Usually left blank.

Log Interval Enter the frequency to sample data in **Log Interval**. See *Changing the Network Manager options* on page 41 to change the default value.

Use BACnet COV Event (Available only when dragging objects that support change of value to the Device Trend Configuration list.) Select the check box to record the value of the property only when an COV notification takes place. See *Change Of Value (COV) notifications* on page 119.

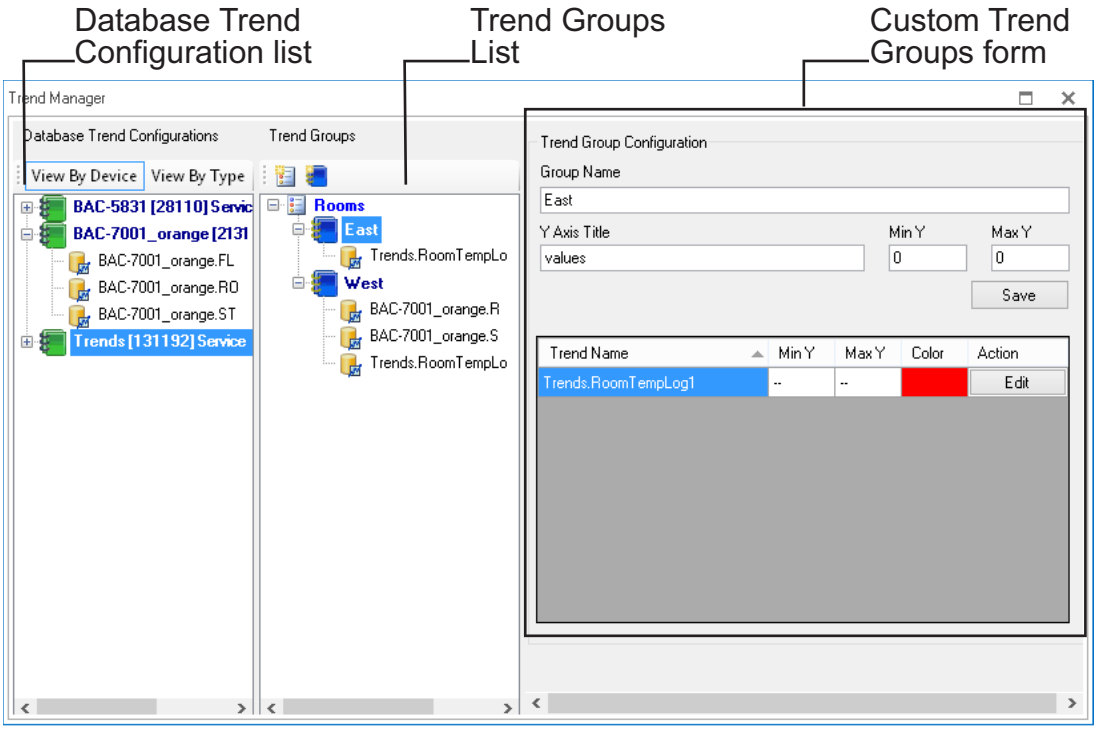
COV Increment (Available only when dragging objects that support change of value to the Device Trend Configuration list.) Enter the value by which **Property** must change to initiate a COV notification.

Adding BACnet trend logs to trend groups

Trend logs are organized in the KMC Connect trend viewer with trend categories and groups.

- Trend groups can be placed only in trend categories.
- To set up trend logs with a wizard, see *Trend configuration wizard* on page 90.
- To view the data with the Trend Log viewer, see *Adding BACnet trend logs to trend groups* on page 98

Illustration 14–4 Trend Manager



Add a trend category

To add a trend category do either of the following:

- Click the new category icon above the **Trend Groups** list.
- Right-click in the **Trend Groups** list and choose **New Category** from the shortcut menu.

Add a trend group

To be visible in the KMC Connect Trend Viewer, a trend log—either a PC or device trend log—must be in a trend group. Trend groups can be placed only in trend categories.

1 If there is no trend category in the Trend Groups lsit, add a new category.

2 Select a trend category in the Trend Groups list and then add a new group by doing one of the following:

- Click the new group icon  above the **Trend Groups** list.
- Right-click in the **Trend Groups** list and choose **New Group** from the shortcut menu.

3 In the **Custom Trend Group** dialog enter the following data:

- Under **Group Name** enter the name to appear in the **Trend Selector** list in the Trend Viewer.
- In the **Y Title** text box, enter a label for the value (y) axis chart that will display in the Trend Viewer.
- Set the top and bottom values of the value axis chart with **Max Y** and **Min Y**.

4 Click **Submit** when finished.

Delete a trend, group, or category

1 In the **Trend Groups** list, select the trend, group, or category to delete.

2 Right-click the item and then choose **Remove** from the shortcut list.

Rename a trend group

1 In the **Trend Groups** list, select the trend group to rename.

2 In the **Trend Group Configuration** area, enter a new name in **Group Name**.

3 Click **Save** when finished.

Rename a trend category

1 In the **Trend Groups** list, select the category to rename.

2 Right-click the category and choose **Rename** from the shortcut list.

The parts of the Trend Manager

Use the Trend Manger to set up trends and trend groups to display in the Trend Viewer.

Group Name The entry in **Group Name** is the name under which the group will appear in the Trend Selector list in the Trend Viewer.

Y Title Enter a label for the value (y) axis of the chart that will display in the Trend Viewer.

Y Min and Y Max Set the top and bottom values of the value axis chart with Max Y and Min Y.

Database Trend Configuration list A list of all the devices with objects that are the source for a trend log.

Trend Groups A list of the individual trends, trend groups, and trend categories that are available for viewing with the Trend Viewer.

Trend Name list A list of all of the individual trends within the trend group.

Archiving trend logs

Use the Schedule Archiving and Deleting Trend dialog to export trend data to an `.xml` file. The file is saved to the Trend Archive folder located at the following location.

`C:\ProgramData\KMC Controls\KMC Connect\Trend Archive`

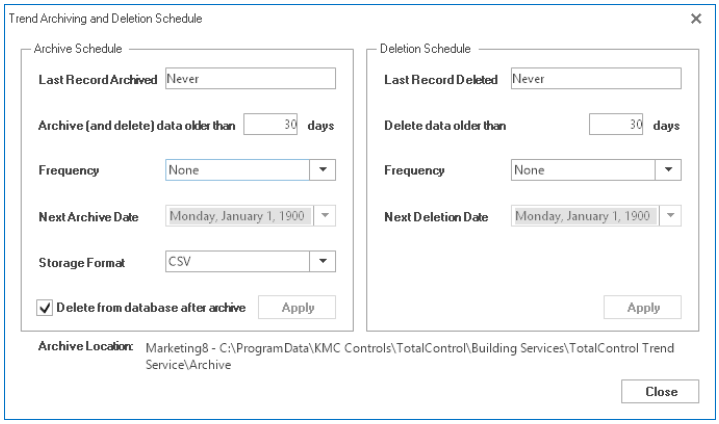
When archiving data you can choose to do the following with data in the site database.

- **Archive** copies the data collected within the date range specified by Start Date and End Date to an `.xml` file.
- **Delete** removes the data from the site database.
- **Archive and Delete** first copies the data to the designated `.xml` file and then removes the data from the database.

To archive trend data, do the following:

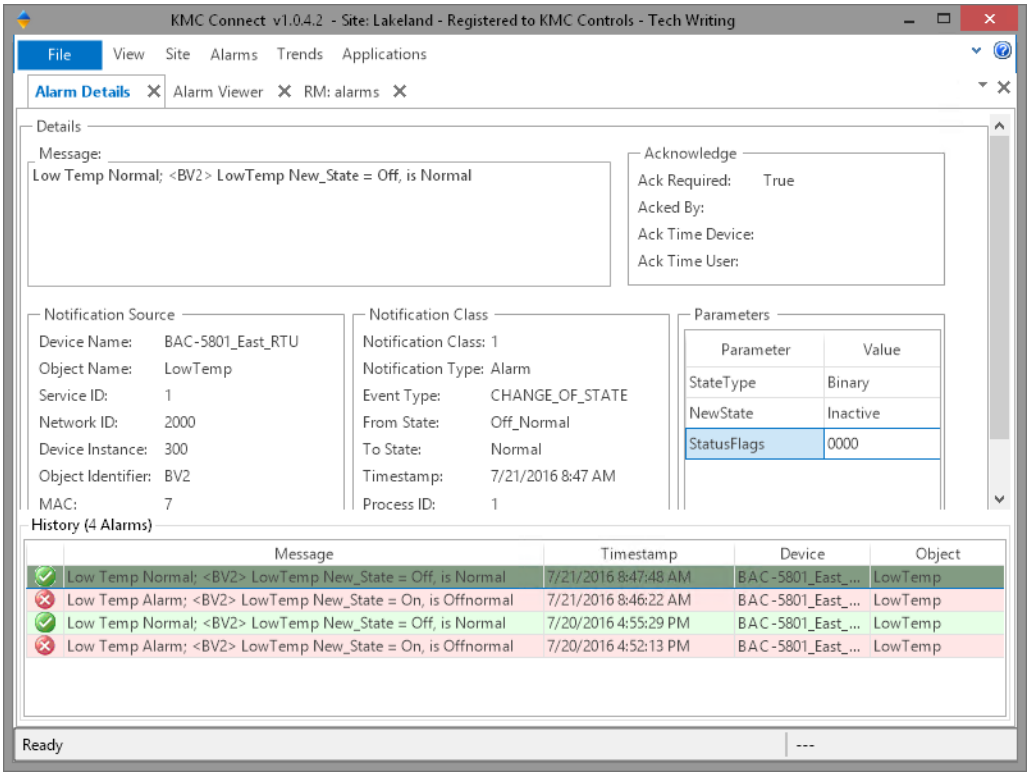
1. On the ribbon choose the **Archive** from the Trends group. The Schedule Archiving and Deleting Trend dialog opens.
2. Select **Archive**, **Delete,** or **Archive and Delete**.
3. Select a period to archive or delete with **Start Date** and **End Date**. Only data within these dates will be archived or deleted.
4. Select an option from **Frequency**.
   - Choose **IMMEDIATE** to archive or delete data as soon as **Send** is clicked.
   - Choose from the other options to repeat the action on a regular schedule. Frequency selection ranges from daily to yearly.
5. Click **Send** when done.

Illustration 14–5 Schedule Archiving and Deleting Trend dialog



KMC Connect

Part IV: Alarms, notifications, and events



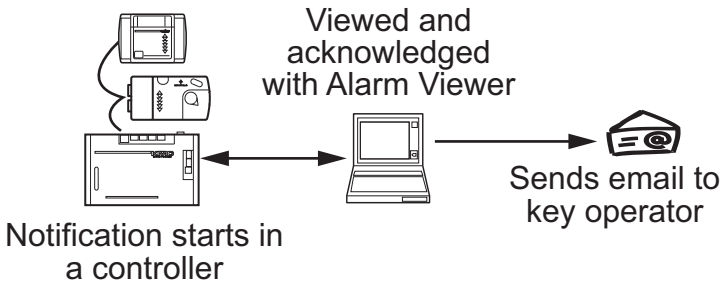
Section 15: An overview of notifications, alarms, and events

KMC Connect includes alarm functions to automatically notify operators that critical system values have changed. This section describes how operator notifications are initiated and how KMC Connect receives, processes, and displays events and alarms.

The alarm process centers around the KMC Connect notification service but includes also the controllers.

- A notification is initiated when the value of a point or property within a controller moves outside of predetermined limits.
- The notification is received by KMC Connect for display in the Alarm Viewer.
- KMC Connect takes action based upon the configuration parameters that created the notification. Actions include alerting key operators, sounding audible alarms and storing the event.
- An operator uses the Alarm Viewer in KMC Connect to view and acknowledge alarms.

Illustration 15–1 KMC Connect alarm process



Origination of notification events Programs within each device monitors as condition and then—when a value moves outside of normal parameters—a notification event is created that KMC Connect can then process.

Programming within the controller determines if the notification should be processed as an alarm or event by KMC Connect. The exact method by which an event is created depends upon the type of controller.

- For BACnet controllers, see *Configuring devices for BACnet alarms* on page 105.

Notification processing KMC Connect processes events for display and operator notification.

- Processes the notification based upon the conditions from which it was generated.
- Stores the event on the computer running KMC Connect.
- Places the event in the alarm viewing list.
- If the event is associated with an alarm class, notifies an operator.

Operator notification KMC Connect alarm classes are the method that designate alarm and event routing to one or more key recipient operators. The recipient operators receive the notifications by e-mail.

Event and alarm viewing Operators view and acknowledge alarms and events with the Alarm Viewer.

Related topics

- *Viewing, acknowledging, and clearing alarms and events* on page 127
- *Emailing operator alarms* on page 135

Section 16: Configuring devices for BACnet alarms

Topics in this section are an overview on intrinsic and algorithmic BACnet notifications. It covers also methods to configure BACnet devices to generate alarms and events from algorithmic and intrinsic notifications.

Alarm and event notifications are set up within the objects in a BACnet controller. Typically, the notifications are initiated with either of the following methods:

- **Intrinsic** This method of generating alarm and event notification is configured using only standard BACnet properties within the objects that support events and alarms.
- **Algorithmic** Algorithmic reporting initiates alarm and event notifications based on criteria maintained in an event enrollment object.

A third method, change of value reporting, is discussed in the topic *Change Of Value (COV) notifications* on page 119.

The following topics provide information about the intrinsic and algorithmic notification processes and the methods to set up the notifications.

- *About intrinsic BACnet alarms and events* on page 108
- *Configuring BACnet devices for intrinsic alarms* on page 111
- *About algorithmic reporting in BACnet devices* on page 113
- *Configuring event enrollment objects* on page 114
- *Configuring a notification class object* on page 116
- *Change Of Value (COV) notifications* on page 119

Tip: The Alarm wizard is quick method to set up intrinsic or algorithmic alarms. See the topic *Using the Alarm wizard for BACnet alarms* on page 106 for the procedure.

Using the Alarm wizard for BACnet alarms

The Alarm wizard is an easy way to configure alarms and notifications in BACnet controllers. Using the wizard is an alternative to setting up properties in monitored objects and the associated notification class and event enrollment objects.

- The wizard can configure input, accumulator, output, or value objects.
- Only one type of object can be configured at one time.
- The objects can be from a mix of controllers that support either algorithmic or intrinsic alarms.

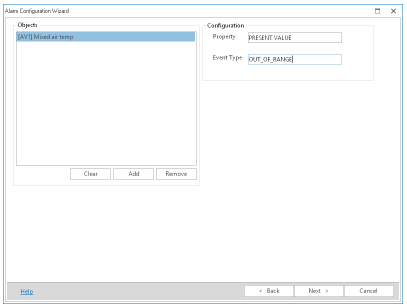
Start the wizard

1 On the ribbon choose **Alarm Configuration Wizard** from the **Alarm** group.

2 Add objects to the Object Selection list.

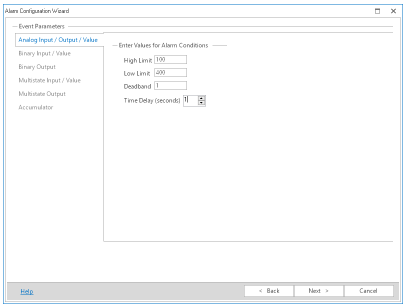
- Drag the objects from Network Manager list to the Objects list.
- Click **Add** and then search for objects. See the procedure *Add by searching* on page 107.
- All of the objects must be of the same type. For example all analog input objects or all binary value objects.

Add additional similar objects and then click **Next** to advance.



3 Enter the alarm parameters. The exact properties will be depend upon the type of object.

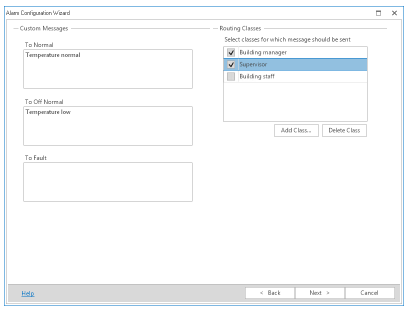
Click **Next** to advance.



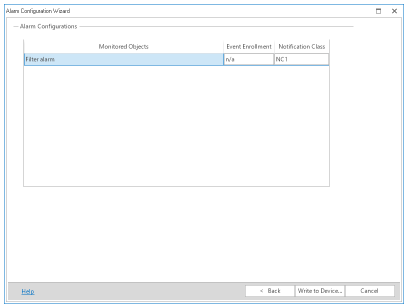
4 Enter custom messages and select the operators that will receive the alarm routing.

- Custom messages display in the Alarm viewer.
- Routing classes are groups of recipient operators that will receive the same alarm by email. See *Emailing operator alarms* on page 135.

Click **Next** to advance.



5 The Alarm Configurations list shows the Notification Class assignments , and where applicable, Event Enrollment objects. Right-click on a cell to change assignments.



6 Click **Write to Devices** to finish. KMC Connect updates the alarm properties in the selected objects.

Add by searching

Objects in the Network Manager list can be added to the Alarm wizard Object Selection list by using the Search and Add function.

To add objects by searching, do the following:

1 Start the wizard.

2 In the Object Select list, click Add.

3 Enter the instance number of a device in the Network Manager list.

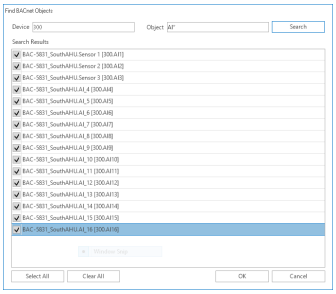
4 Enter the mnemonics for the object followed by the object number.

- Use AI for Analog Input, BO for Binary Output objects, MSV for Multistate Value objects, etc.
- Use an asterisk (*) for all objects of the same type. For example AI* for all analog input objects.
- To search for all devices, use an asterisk (*) in the Device box.
- Use a question mark (?) to substitute for characters when searching by name. For example ***SENSOR?*** will find both ***SENSOR1*** and ***SENSOR2*** but not ***SENSOR10***.

5 Click **Search**.

6 Select or clear the check boxes next to each object found by the search.

7 Click **OK** to add the selected objects to the list.



Additional information on BACnet alarms

- *Emailing operator alarms* on page 135
- *About intrinsic BACnet alarms and events* on page 108
- *About algorithmic reporting in BACnet devices* on page 113
- *Reference to BACnet objects* on page 315

About intrinsic BACnet alarms and events

Intrinsic notification reporting is a method of monitoring the value of a property and issuing notifications from within a BACnet object.

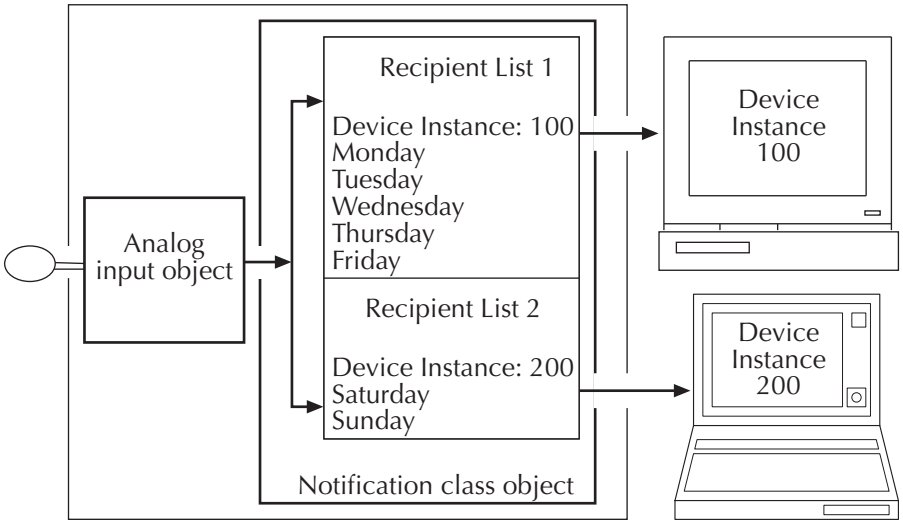
Notification takes place when a monitored value moves outside of predefined limits. The object then sends a notification to a notification class object. Within notification class objects are one or more recipient lists that include the designated notification handling device such as KMC Connect.

An example of intrinsic notification

In the example shown in the diagram *The intrinsic alarm and event notification flow* on page 109, a thermistor is connected to an analog input object. The object is configured to generate events when the temperature input moves outside of a predefined temperature range. When the temperature makes the transition outside of the normal range, the following actions take place:

- The input object sends the event to the designated notification class object for further handling.
- On weekdays, Recipient List 1 directs the event to the computer with workstation configured as device instance #100.
- On weekends, Recipient List 2 sends the event to the computer with workstation configured as device instance #200.
- The operator workstation processes the event. The exact actions that take place at the operator workstation depend upon the capabilities of the workstation and the type and priority of the notification.

Illustration 16–1 The intrinsic alarm and event notification flow



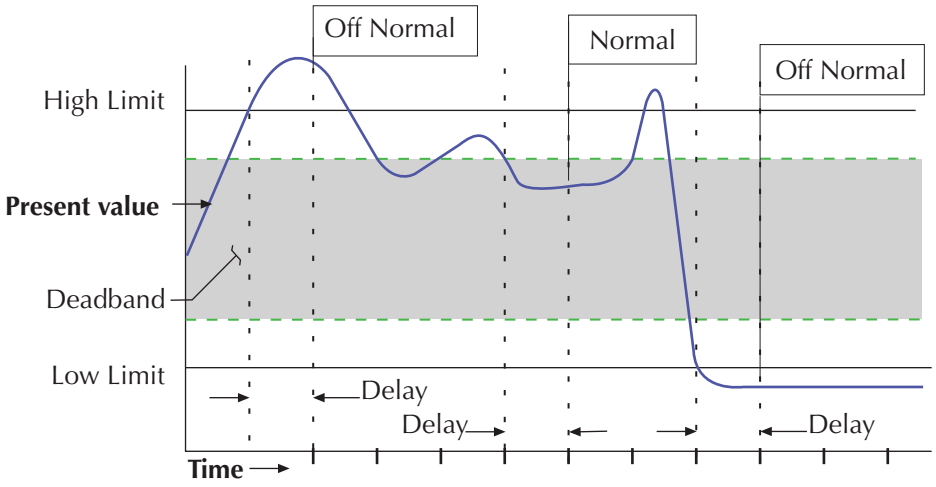
Notifications in analog objects

Each BACnet object that supports notifications has within it properties that define conditions under which an event is initiated.

- When the monitored value makes a transition from normal conditions to a value outside of normal conditions, the object sends a ***To Off Normal*** event to a notification class object.
- ***To Normal*** events are generated when present value makes the transition from outside the defined range to normal conditions.

The following illustration is an example of an analog object configured to generate events when present value falls outside of predefined limits.

Illustration 16–2 Example of Off Normal and Normal events in analog objects



Conditions for generating a ***To Off Normal*** event when ***Present Value*** exceeds normal limits:

- ***Present Value*** must be either greater than ***High Limit*** or less than ***Low Limit*** for the period specified by ***Time Delay,*** and
- ***Low/High*** must be selected in ***Limit Enable***, and
- The selection in ***Event Enable*** must include ***Off Normal.***

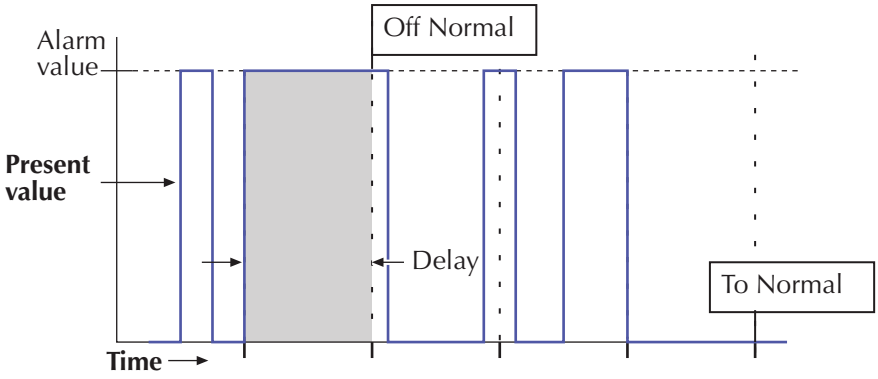
Conditions for generating a ***To Normal*** event when ***Present Value*** returns to a normal value:

- ***Present Value*** must return to a level that is less than ***High Limit*** minus ***Deadband*** and greater than ***Low Limit*** plus ***Deadband*** for the period specified by ***Delay***, and
- ***Low/High*** must be selected in ***Limit Enable***, and
- The selection in ***Event Enable*** must include ***NORMAL***.
- ***Delay*** and ***Deadband*** provide a buffer that ignores minor fluctuations in ***Present Value***.

Notifications in binary objects

The following illustration is an example of an binary object configured to generate events when present value falls outside of predefined limits.

Illustration 16–3 Example of Off Normal and Normal events



Conditions for generating a ***To Off Normal*** event in a binary object.

- ***Present Value*** must maintain the value specified by ***Alarm Value*** for the period specified by ***Time Delay,*** and
- The selection in ***Event Enable*** must include ***To Off Normal.***

Conditions for generating a ***To Normal*** event in binary objects:

- ***Present Value*** must remain unequal to the value specified by ***Alarm Value*** for the period specified by ***Time Delay,*** and
- The selection in ***Event Enable*** must include ***To Normal***.

Configuring BACnet devices for intrinsic alarms

Configuring a BACnet device to send a notification to KMC Connect is a two part process.

1. First, configure the alarm limits within an object. Use one of the following procedures to configure the limits.
   - For analog inputs, output or value objects, see *Notifications in analog objects* on page 111.
   - For loop objects or binary input, output or value objects, see *Notifications in binary objects* on page 112.
2. Next, configure a notification class object to send the notification to a handler. See *Configuring a notification class object* on page 116.

Tip: An alternative to using the procedures in this topic is to use the Alarm wizard. See the topic *Using the Alarm wizard for BACnet alarms* on page 106

Notifications in analog objects

To configure an object to detect an out-of-range value in an analog property maintained within a BACnet object, do the following:

1. Locate and open the device and object in the Network Manager list.
2. Expand the section **Event/Alarm Properties**.

3 Choose the state that will initiate the notification by selecting the **Event Enable** check box. Selections are **To Normal**, **To Fault**, or **To Off Normal**. This selection must match the selection of the **Transitions** in the recipient list of the notification class object that routes the notification.

4 Select the notification class object that will receive the event.

5 Under Notification Type select either **Alarm** or **Event.** The manner in which KMC Connect displays events and alarms is described in the topic *Viewing, acknowledging, and clearing alarms and events* on page 127.

6 Enter values for the following alarm limits:

- Low Limit
- High Limit
- Deadband
- Time Delay

7 Expand the section **Alarm Routing Properties** and then select the **Class Routing** from the list.

8 When done, click **Save Changes** at the top of the object tab.

9 Configure the notification class object referenced in step 4. See the topic *Configuring a notification class object* on page 116.

Notifications in binary objects

To configure an object to detect a binary value that changes state in a property maintained within a BACnet object, do the following:

1 Locate and open the device and object in the Network Manager list.

2 Expand the section **Event/Alarm Properties**.

3 Choose the state that will initiate the notification by selecting the **Event Enable** check box. Selections are **To Normal**, **To Fault**, or **To Off Normal**. This selection must match the selection of the **Transitions** in the recipient list of the notification class object that routes the notification.

4 Select the notification class object that will receive the event.

5 Under Notification Type select either **Alarm** or **Event.** The manner in which KMC Connect displays events and alarms is described in the topic *Viewing, acknowledging, and clearing alarms and events* on page 127.

6 Under Alarm Value select either **Active** or **Inactive**.

7 Expand the section **Alarm Routing Properties** and then select the **Class Routing** from the list.

8 When done, click **Save Changes** at the top of the object tab.

9 Configure the notification class object referenced in step 4. See the topic *Configuring a notification class object* on page 116.

Related topics

- *About intrinsic BACnet alarms and events* on page 108
- *Notification object* on page 365

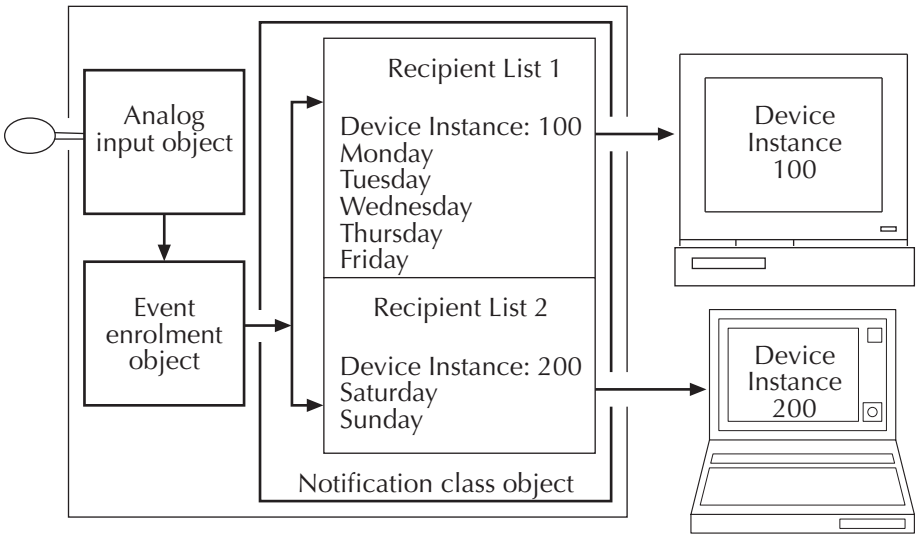
About algorithmic reporting in BACnet devices

Algorithmic reporting initiates alarm and event notifications based on criteria maintained in an event enrollment object. Event enrollment objects are standard BACnet objects that monitor the value of properties in other BACnet objects for alarm or event conditions. When the condition is detected, the event enrollment object sends the notification to a notification class object for routing to the notification handling device. Within notification class objects are one or more recipient lists that include the designated BACnet notification handling device.

In the example shown in the diagram *Algorithmic notification flow* on page 113, a thermistor is connected to an analog input object. The input object is monitored by an event enrollment object that will generate events when the temperature input moves outside of a predefined temperature range. When the temperature makes the transition outside of the normal range, the following actions take place:

- The event enrollment object sends the event to the designated notification class object for further handling.
- On weekdays, Recipient List 1 directs the event to the computer with workstation configured as device instance #100.
- On weekends, Recipient List 2 sends the event to the computer with workstation configured as device instance #200.
- The operator workstation processes the event. The exact actions that take place at the operator workstation depend upon the capabilities of the workstation and the type and priority of the notification.

Illustration 16–4 Algorithmic notification flow



Related topics

- *Configuring event enrollment objects* on page 114
- *Event enrollment object* on page 329
- *Notification object* on page 365
- *About intrinsic BACnet alarms and events* on page 108

Configuring event enrollment objects

Configuring algorithmic notification reporting in KMC Connect is a two part process.

1. First, configure the alarm limits within an event enrollment object. Use one of the following procedures to configure the limits.
   - For analog inputs, output or value objects, see *Configuring an event enrollment object for analog values* on page 114.
   - For loop objects or binary input, output or value objects, see *Configuring an event enrollment object for binary values* on page 115.
2. Next, configure the notification class object to send the notification to a handler. see *Configuring a notification class object* on page 116.

Tip: An alternative to using the procedures in this topic is to use the Alarm wizard. See the topic *Using the Alarm wizard for BACnet alarms* on page 106

Tip: The following procedures describe methods similar to configuring intrinsic notification for analog and binary objects. There are, however, methods for configuring an event enrollment object other than the ones described here.

Configuring an event enrollment object for analog values

The following procedure describes configuring an event enrollment object to detect an out-of-range analog value. This procedure is similar to notifications initiated by intrinsic notifications as described in the topic *Notifications in analog objects* on page 109.

1. Locate and open the event enrollment object in the Network Manager list.
2. Expand the section **General Properties**.
3. Under Event Type select **Out of Range**.
4. Enter values for the following alarm limits:
   - Low Limit
   - High Limit
   - Deadband
   - Time Delay

5 Under **Device, Object and Property**, enter the following:

- Choose **Local** or enter the device instance of a remote controller
- The object to monitor
- The analog property to monitor

6 Expand the section **Event/Alarm Properties**.

7 Choose the state that will initiate the notification by selecting the **Event Enable** check box. Selections are **To Normal**, **To Fault**, or **To Off Normal**. This selection must match the selection of the **Transitions** in the recipient list of the notification class object that routes the notification.

8 Select the notification class object that will receive the event.

9 Under Notification Type select either **Alarm** or **Event.** The manner in which KMC Connect displays events and alarms is described in the topic *Viewing, acknowledging, and clearing alarms and events* on page 127.

10 Expand the section **Alarm Routing Properties** and then select the **Class Routing** from the list.

11 When done, click **Save Changes** at the top of the object tab.

12 Configure the notification class object referenced in step 8. See *Configuring a notification class object* on page 116.

Configuring an event enrollment object for binary values

The following procedure describes configuring an event enrolment object to detect when a binary value changes state. This procedure is similar to notifications initiated by intrinsic notifications as described in the topic *Notifications in binary objects* on page 110.

1 Locate and open the event enrollment object in the Network Manager list.

2 Expand the section **General Properties**.

3 Under Event Type select **Change Of State**.

4 Under Property Type, choose Binary.

5 Under List of Values do the following:

a. Click **Add**.

b. Select either **Active** or **Inactive**. This is the state that initiates a To Off Normal notification.

6 Under **Device, Object and Property**, enter the following:

- Choose **Local** or enter the device instance of a remote controller
- Enter the object to monitor
- The binary property to monitor

7 Expand the section **Event/Alarm Properties**.

8 Choose the state that will initiate the notification by selecting the **Event Enable** check box. Selections are **To Normal**, **To Fault**, or **To Off Normal**. This selection must match

the selection of the **Transitions** in the recipient list of the notification class object that routes the notification.

9 Select the notification class object that will receive the event.

10 Under Notification Type select either **Alarm** or **Event.** The manner in which KMC Connect displays events and alarms is described in the topic *Viewing, acknowledging, and clearing alarms and events* on page 127.

11 Expand the section **Alarm Routing Properties** and then select the **Class Routing** from the list.

12 When done, click **Save Changes** at the top of the object tab.

13 Configure the notification class object referenced in step 9. See *Configuring a notification class object* on page 116.

Related topics

- *About algorithmic reporting in BACnet devices* on page 113
- *Event enrollment object* on page 329
- *Notification object* on page 365
- *Viewing all event enrollment subscriptions* on page 124

Configuring a notification class object

Notification class objects receive event notification from other objects and then route them to an alarm handling device such as KMC Connect. A notification class object does the following:

- Designates the priority for the handling device to process the notification as either an alarm or event.
- Maintains one or more recipient lists that designate—by device instance—the devices that will handle the event. Each recipient list includes also a day of the week and a time span. By configuring multiple recipient lists for different days of the week or periods of time, notifications can be sent to the workstations or devices where operators can take appropriate action.
- Designates that an acknowledgment is required from an operator at a BACnet Operator Workstation such as KMC Connect.

To configure a notification class object, do the following:

1 Locate and open the notification class object in the Network Manager list.

2. Add or edit the following in the recipient list:
   - Enter the device instance number of the BACnet operator workstation that will receive the notification. For KMC Connect the device instance is displayed in Network Manager under Network 0 for the BACnet service.
   - Under ***Days***, select the day of the week and time for notification.
   - Under ***Transitions***, select the state that triggers the event. This must match the selection under ***Event Enable*** in the properties of the object-in-alarm.
   - Select ***Send Confirmed Notifications***. This is a required selection for KMC Connect.
3. For transitions that require an acknowledgment by an operator, select the conditions in **ACK Required**.
4. Set the process identifier to any non-zero value.
5. When done, click **Save Changes** at the top of the object tab.

Related topics

- *Notification object* on page 365
- *Configuring BACnet devices for intrinsic alarms* on page 111
- *Configuring event enrollment objects* on page 114
- *Configuring email notifications* on page 136

Section 17: Change Of Value (COV) notifications

The BACnet Change Of Value notification services are an alternative to polling methods of monitoring values in a BACnet system. Topics in the section cover information about the COV services and procedures to implement them.

BACnet Change Of Value (COV) services send notifications when a monitored value within a device changes by a specified increment. These services are a subset of the alarm and notification services.

BACnet COV notifications have two advantages over the algorithmic or intrinsic notifications polling methods.

- Bandwidth is reduced because only changes of value create notifications.
- Short duration changes that occur between polling intervals are captured.

The following topics describe the COV services and set up procedures.

- *The parts of COV* on page 119
- *Subscribing to an object on a configuration page* on page 121
- *Subscribing to a property on a configuration page* on page 122
- *Subscribing trend logs to a COV server device* on page 124
- *Unsubscribing from COV subscriptions* on page 123

The parts of COV

There are two key components of change of value notification.

- The COV server is part of a standard BACnet device that detects and reports the changes in value of a monitored property.
- The COV client is the notification handler for COV servers.

COV server

A COV server is a BACnet device that supports COV subscriptions and sends COV notification messages to a COV client. The COV server device monitors the value of the object property and compares it with criteria in a subscription list. When the criteria matches the criteria in the subscription list, the COV server device sends notifications of the new value to the COV clients in its subscription lists.

A COV server device supports two types of COV subscriptions.

Subscribe to the object When the COV subscription is to an object, only the present value and the status flags are monitored. Either of the following will trigger a COV notification.

- When the status flags change for any reason.
- When the present value in the object changes by the value specified by the value in the increment property.

Subscribe to a property When the COV subscription is to a specific property in an object, either of the following will trigger a notification.

- When the status flags change for any reason.
- When the value of the specified property changes by the value specified by the value in the subscription list.

Subscription list Each COV server device maintains a subscription list with space for at least one subscription. The subscription list describes a relationship between the COV server and the COV client. The list includes the following information:

- **Subscription type**—BACnet provides two services for subscription. One subscribes to an object and the other subscribes to a specific property of an object.
- **COV client identity**—The client processes COV notifications sent by the COV server.
- **Monitored object identity**
- **Subscription period**—A subscription can be for an indefinite period or for a specific number of seconds. For subscriptions that are not indefinite, the maximum period is 4,294,967,295 seconds.
- **Notification type**—Notifications can be either confirmed or unconfirmed. A confirmed notification is sent until the client acknowledges the notification. An unconfirmed notification is sent only once without requiring acknowledgment from the client.
- **COV increment**—This parameter is only used in subscriptions when subscribing to a specific property. If the subscription is to an object, the increment used is the increment property in the object.

COV client

A COV client, referred to also as a subscriber, is a service in a BACnet device that subscribes to COV servers and receives COV notifications. Upon receipt of the notifications the COV client processes notification messages. KMC Connect is a client that receives notifications with new values and then caches the new values for use by the following:

- Subscribed BACnet PC trends
- Open configuration pages

BACnet COV servers are not required to maintain subscription lists during a power loss. To prevent loss of data, KMC Connect maintains a list of subscriptions that have been established between KMC Connect and COV server devices. KMC Connect automatically renews unexpired subscriptions under the following conditions.

- Every eight hours
- When the PDS detects that the device has restarted

Subscribing to an object on a configuration page

Open configuration pages for BACnet objects may receive COV notifications from the object to which KMC Connect has subscribed. By subscribing to an object, the Present Value property in a configuration page automatically updates without clicking Refresh at the top of the tab. When subscribing to an object, only a change in present value or the status flags will create a notification.

Use one of the following procedures to subscribe a configuration page to an object:

In the Network Manager list

1. Locate the object in the Network Manager list.
2. Right-click the object and then choose **COV** from the shortcut menu.
3. In the COV shortcut menu choose one of the following:
   - Subscribe Confirmed
   - Subscribe COV Unconfirmed

From the object configuration tab

1. Locate the object in the Network Manager list.
2. Open the configuration tab for the object.
3. Right-click the over the Present Value text box and then choose **COV** from the shortcut menu.
4. In the COV shortcut menu choose one of the following:
   - Subscribe Confirmed
   - Subscribe COV Unconfirmed
5. Enter the increment in **COV Increment**.
6. When done, click **Save Changes** at the top of the object tab.

Related topics

- *The parts of COV* on page 119
- *Subscribing to a property on a configuration page* on page 122
- *Unsubscribing from COV subscriptions* on page 123

Subscribing to a property on a configuration page

Open configuration pages for BACnet objects may receive COV notifications from the a specific property in an object to which KMC Connect has subscribed. By subscribing to a property, the monitored property in a configuration page automatically updates without clicking Refresh at the top of the tab. Subscribing to a property offers more choices for COV notifications than subscribing to an object.

Use one for the following procedures to subscribe KMC Connect to a specific property in an object.

In the Network Manager list

1. Locate the object in the Network Manager list.
2. Right-click the object and then choose **COV** from the shortcut menu.
3. In the COV shortcut menu choose **Subscribe COV Property**.
4. Enter settings in the **Subscribe COV Property** dialog.
5. When done, click **OK**.

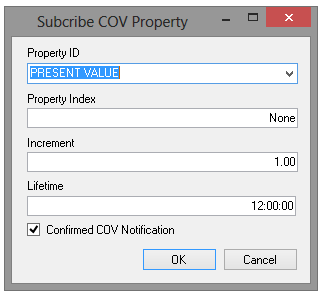
From the object configuration tab

1. Locate the object in the Network Manager list.
2. Open the configuration tab for the object.
3. Right-click over the Present Value text box and then choose **COV** from the shortcut menu.
4. In the COV shortcut menu choose **Subscribe COV Property**.
5. Enter settings in the Subscribe COV Property dialog.
6. Click **OK** to save the changes in the dialog
7. When done, click **Save Changes** at the top of the object tab.

The COV Property dialog

Use the COV Property dialog to set up subscribing to a COV property in an object.

Illustration 17–1 Subscribe COV Property dialog



Property ID The property within the object to monitor. Typically this is set to Present Value.

Property Index Used only for properties that are arrays.

Increment This entry specifies the minimum change of the value specified in Property ID that will send a COV notification to subscriber notification clients.

Lifetime Sets the duration of the subscription.

Confirmed COV Notification Sets the requirement for confirmed or unconfirmed notification.

- When this check box is clear, the COV notification is sent to the client—KMC Connect—without a confirmation.
- When the check box is selected, the notification is sent until the client—KMC Connect—returns a confirmation.

Unsubscribing from COV subscriptions

When a COV client no longer needs the subscription, it unsubscribes from the COV server device. If it needs the subscription maintained then the client should periodically re-subscribe.

- Subscriptions for a KMC Connect configuration page are automatically canceled when KMC Connect closes.
- Subscription periods for trend logs are specified when an PC trend log is created for a BACnet object. If a value is not specified, the subscription is indefinite.

To cancel a subscription, do one of the following:

From the object configuration tab

1. Locate the object in the Network Manager list.
2. Open the configuration tab for the object.
3. Right-click over the Present Value text and then choose **COV** from the shortcut menu.
4. In the COV shortcut menu choose **Unsubscribe COV**.
5. Enter settings in the **Unsubscribe COV Property** dialog.
6. Click **OK** to save the changes in the dialog
7. When done, click **Save Changes** at the top of the object tab.

In the Network Manager list

1. Locate the object in the Network Manager list.
2. Right-click the object and then choose **COV** from the shortcut menu.
3. In the COV shortcut menu choose **Unsubscribe COV**.
4. Enter settings in the **Unsubscribe COV Property** dialog.
5. Click **OK** to save the changes in the dialog.

From the device object configuration page

1. Locate the object in the Network Manager list.

2 Expand the section **Active COV Subscription Properties.**

3 Right-click the subscription in the **Active Change of Value Subscriptions** list.

- Unsubscribe COV
- Unsubscribe All COVs.

4 When done, click **Save Changes** at the top of the object tab.

From the BACnet service icon

1 Locate the BACnet service icon in the Network Manager list. Double-click the icon to open the configuration tab.

2 Expand the COV Subscriptions area.

3 Scroll through the list to locate the subscription.

4 Right-click the subscription and then choose one of the following from the shortcut menu:

- Unsubscribe COV
- Unsubscribe All COVs

5 When changes are complete, click **Save Changes to Service** at the top of the configuration page.

Subscribing trend logs to a COV server device

A PC trend log may be set to subscribe to a device that supports COV subscriptions when it is set up.

KMC Connect automatically renews the subscription for a PC trend under the following conditions.

- Every eight hours unexpired subscriptions are automatically reestablished.
- When KMC Connect detects that the device has restarted.

Trend logs with COV subscriptions can also be set up with the Trend configuration wizard. See the topic *Trend configuration wizard* on page 90.

Viewing all event enrollment subscriptions

To view all event enrollment subscriptions in a device, do the following:

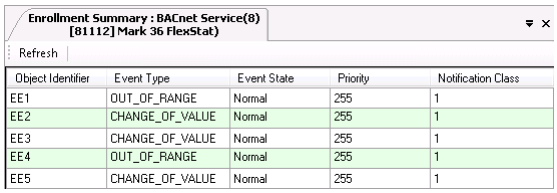
1 Locate the device in the Network Manager list.

2 Right-click the device icon and then choose **Get Enrollment Summary** from the shortcut menu.

3 A new tab opens in the KMC Connect workspace.

4 If needed, click **Refresh** to retrieve the latest enrollment information from the device.

Illustration 17–2 Summary of event enrollment subscriptions



Object Identifier	Event Type	Event State	Priority	Notification Class
EE1	OUT_OF_RANGE	Normal	255	1
EE2	CHANGE_OF_VALUE	Normal	255	1
EE3	CHANGE_OF_VALUE	Normal	255	1
EE4	OUT_OF_RANGE	Normal	255	1
EE5	CHANGE_OF_VALUE	Normal	255	1

Related topics

- *Change Of Value (COV) notifications* on page 119
- *Event enrollment object* on page 329
- *Configuring event enrollment objects* on page 114

Section 18: Viewing, acknowledging, and clearing alarms and events

Topics in this section describe how to view, acknowledge and clear alarm and event notifications within KMC Connect.

Notifications from controllers in a KMC Connect monitored site are displayed as either alarms or events. Either can be viewed, acknowledged and cleared from the Alarm Viewer.

- *An overview of notifications, alarms, and events* on page 103
- *Viewing alarms* on page 128
- *Acknowledging alarms* on page 130
- *Clearing alarms* on page 130
- *Filtering alarms* on page 133
- *Disabling alarms by network* on page 134
- *Viewing a summary of BACnet notifications* on page 130
- *Using the Alarm Monitor bar* on page 131

Viewing alarms

You can view and acknowledge alarm and event notifications from the Alarm Viewer.

To open Alarm Viewer, on the ribbon choose **Alarm Viewer** from the **Alarms** group.

The Alarm Viewer tab opens with a list of all points and objects that have alarms or events that have not been archived.

- The icons in the first column indicate the type of notification. See the table *Notification type icons* on page 128.
- The icons in the second column indicate the status of the notification. See the table *Acknowledgment status icons* on page 128.
- The background color of each row indicates the type of notification.
  - Red—Notification of an alarm condition in a BACnet or KMD controller.
  - Green—Notification that an alarm condition in a BACnet or KMD controller has returned to normal.
  - Yellow—Alarms that do not require acknowledgment.
  - White—Notification of a BACnet event.

Illustration 18–1 Alarm Viewer tab in the Alarm Viewer

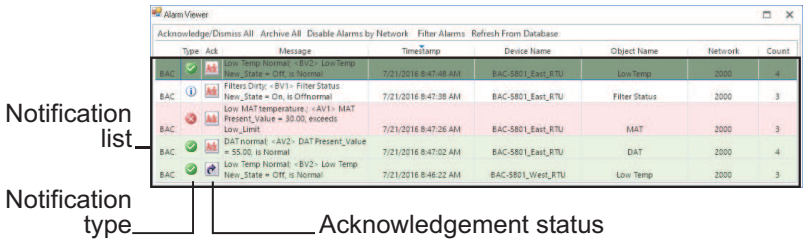


Table 18-1 Notification type icons

Icon	Description
	Notification of a BACnet event
	Notification of a BACnet or KMD *To Off Normal* alarm
	Notification of a BACnet or KMD *To Normal* alarm

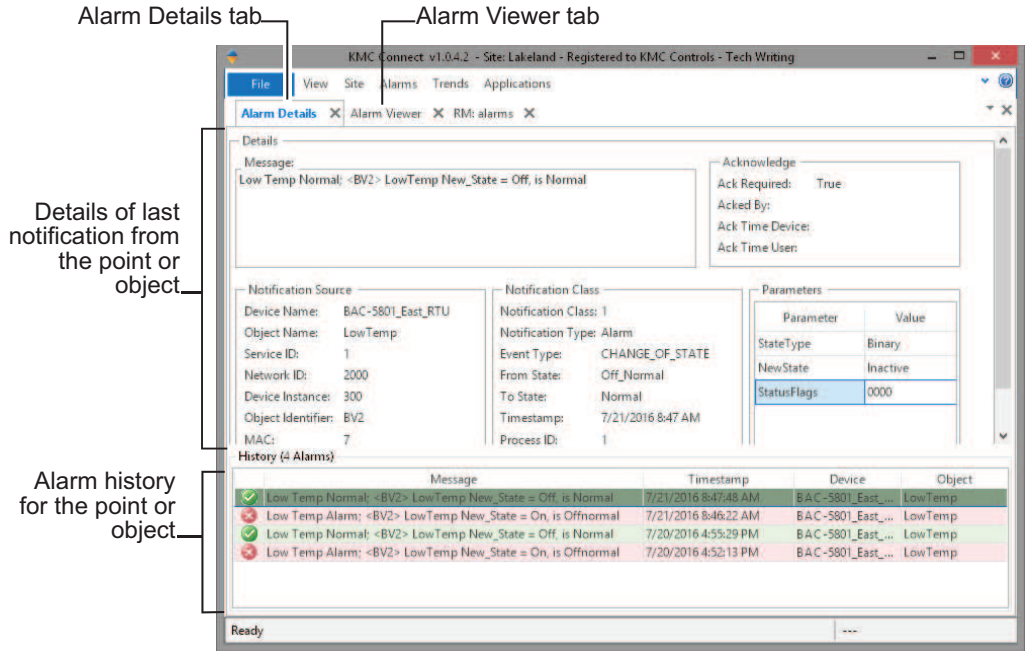
Table 18-2 Acknowledgment status icons

Icon	Description
	A notification in the database is waiting for an acknowledgment from an operator.
	A notification that does not require an operator acknowledgment.
	KMC Connect is updating the status.
	An operator has acknowledged the notification.

To view notification details

1. Right-click an alarm in the **Alarm Viewer** list.
2. Choose **Show Details** from the shortcut menu. The **Alarm Details** tab opens.

Illustration 18–2 Alarm Details tab



Related topics

- *An overview of notifications, alarms, and events* on page 103
- *Acknowledging alarms* on page 130
- *Clearing alarms* on page 130
- *Using the Alarm Monitor bar* on page 131
- *Filtering alarms* on page 133
- *Disabling alarms by network* on page 134

Acknowledging alarms

Notifications that require an operator acknowledgment are identified with the  icon in the **Ack** column of the notification list.

Do the following to acknowledge an alarm in KMC Connect:

1. On the ribbon choose **Alarm Viewer** from the **Alarms** group.
2. Click the **Ack** icon in the alarm status column of the notification list.

Related topics

- *An overview of notifications, alarms, and events* on page 103
- *Viewing alarms* on page 128
- *Clearing alarms* on page 130
- *Using the Alarm Monitor bar* on page 131
- *Filtering alarms* on page 133
- *Disabling alarms by network* on page 134

Clearing alarms

Archive alarms to remove them from the Alarm Viewer list. This removes the alarm or event from the alarm list but leaves it in the database.

1. On the ribbon choose **Alarm Viewer** from the **Alarms** group.
2. Right-click the alarm in the **Alarm Viewer** tab and choose **Archive**.

Related topics

- *An overview of notifications, alarms, and events* on page 103
- *Viewing alarms* on page 128
- *Acknowledging alarms* on page 130
- *Using the Alarm Monitor bar* on page 131
- *Filtering alarms* on page 133
- *Disabling alarms by network* on page 134

Viewing a summary of BACnet notifications

To view a summary of current alarms or events in a BACnet device, do the following:

1. Locate the device in the Network Manager list.
2. Right-click the device icon and then choose one of the following from the shortcut menu.
   - Get Event Information
   - Get Alarm Summary

3 A new tab opens in the KMC Connect workspace with a list of current alarms or events.

4 If needed, click **Refresh** to retrieve the latest information from the device.

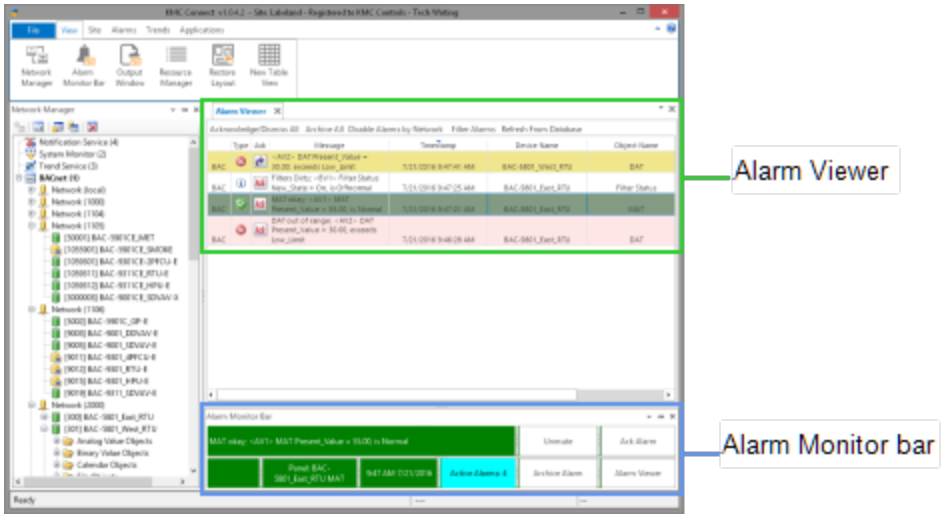
Related topics

- *An overview of notifications, alarms, and events* on page 103
- *Viewing alarms* on page 128
- *Acknowledging alarms* on page 130
- *Clearing alarms* on page 130
- *Using the Alarm Monitor bar* on page 131
- *Filtering alarms* on page 133
- *Disabling alarms by network* on page 134

Using the Alarm Monitor bar

The Alarm Monitor bar opens when a new notification, either an alarm or event, is received by KMC Connect. It remains in the workspace until it is closed by an operator.

Illustration 18–3 Alarm Viewer and Alarm Monitor bar



The background of the Alarm Monitor bar changes color depending on the type and state of the notification. See the following table for background colors.

Table 18-3 Alarm Monitor colors

	Ack required	Acked	Ack Error	Ack in progress
To Off Normal Ack Required	Red	Green	Red	Green
To Normal Ack Required	Green	Green	Green	Green
To Off Normal Ack Not Required	Yellow	Green	Yellow	Green
To Normal Ack Not Required	Green	Green	Green	Green
Event	White	White	White	White

The operator can perform the following actions from the Alarm Monitor bar.

Mute Enables and disables the audible alarm.

Ack or Dismiss Alarm Click to acknowledge an alarm or to dismiss an alarm that does not require acknowledgment.

Archive Alarm Click to remove the alarm from the list. The alarm remains in the database.

Alarm Viewer Opens the KMC Connect alarm viewer. See *Viewing alarms* on page 128.

Filtering alarms

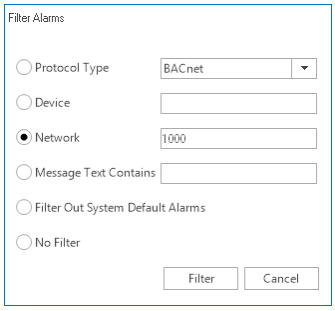
In the Alarm Viewer, alarms and events can be filtered by the following criteria.

- Protocol type
- Device instance number
- Network number
- Contents of the message text
- System default alarms.

To filter alarms, do the following:

1. At the top of the Alarm Viewer, click **Filter Alarms**.
2. Choose the criteria for filtering.
3. Click **Filter** when finished.

Illustration 18–4 Filter Alarms dialog



Related topics

- *An overview of notifications, alarms, and events* on page 103
- *Viewing alarms* on page 128
- *Acknowledging alarms* on page 130
- *Clearing alarms* on page 130
- *Disabling alarms by network* on page 134
- *Viewing a summary of BACnet notifications* on page 130
- *Using the Alarm Monitor bar* on page 131

Disabling alarms by network

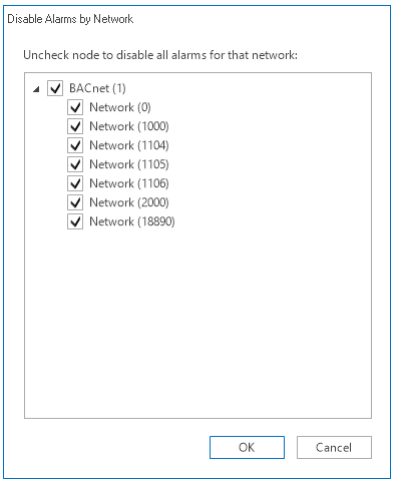
Alarms can be disabled by the network from where they originate. This is useful during initial installation and start up to avoid nuisance or invalid alarms.

Alarms from disabled networks are not handled or stored by KMC Connect. They cannot be retrieved a latter date.

To disable alarms by network, do the following:

1. At the top of the Alarm Viewer, click **Disable Alarms by Network**.
2. Choose the networks to excluded. Network 0 is the network to which KMC Connect is connected.
3. Click **OK** when finished.

Illustration 18–5 Disable Alarms by Network dialog



Related topics

- *An overview of notifications, alarms, and events* on page 103
- *Viewing alarms* on page 128
- *Acknowledging alarms* on page 130
- *Clearing alarms* on page 130
- *Filtering alarms* on page 133
- *Viewing a summary of BACnet notifications* on page 130
- *Using the Alarm Monitor bar* on page 131

Section 19: Emailing operator alarms

Topics in this section cover how to configure KMC Connect to process notifications and alert key operators to alarm conditions.

In addition to alarms and event being placed in alarm viewer lists, alarms and events can be routed to specific users by e-mail.

- *Setting up alarm emails* on page 135
- *Configuring email notifications* on page 136
- *Filtering alarms by security groups* on page 139
- *Configuring the Notification Service* on page 141

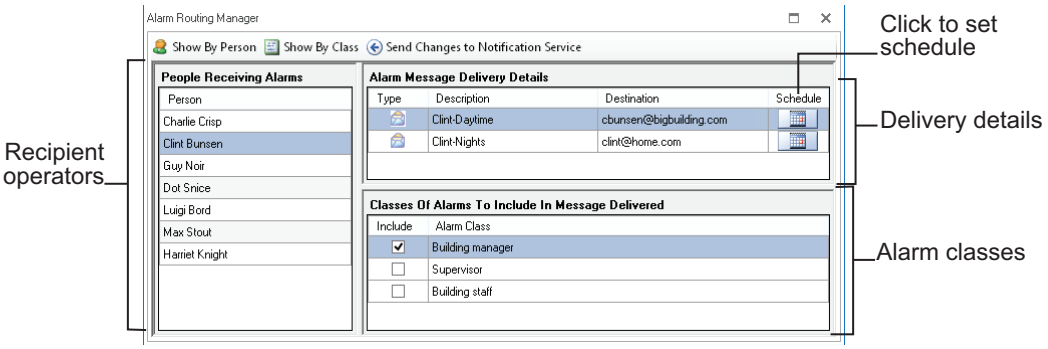
Setting up alarm emails

Use the Alarm Routing Manager to configure KMC Connect to alert key operators when it processes an event or alarm.

- The notification process is explained in the topic *An overview of notifications, alarms, and events* on page 103.
- To set up operator notifications see *Configuring email notifications* on page 136.

To open the routing manager, on the ribbon choose **Routing Manager** from the **Alarms** group.

Illustration 19–1 Alarm Routing Manager



The components of the alarm routing manager

Recipient operators This is a list of operators that will receive alarms and event notifications by email.

Delivery details Each recipient operator can receive notifications by email. Delivery methods must be associated with one or more alarm classes before a notification is sent to an operator.

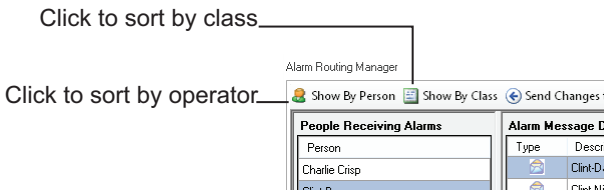
Classes of alarms Alarm classes are a method by which similar event notifications are grouped for operator notifications. The event is designated in the device at the point of origin as belonging to one or more alarm classes. When an event is received by the KMC Connect

notification service ***and*** it is associated with an alarm class, email notifications are sent to key operators based on the configuration in Alarm Routing Manager.

Changing the listing view

Click **Show By Person or Show By Class** to change the display listing. By displaying by alarm class and then choosing a class, you can quickly choose the operators to add to or delete from a class.

Illustration 19–2 Changing the display listing



Related topics

- *Configuring devices for BACnet alarms* on page 105
- *Configuring email notifications* on page 136
- *Viewing, acknowledging, and clearing alarms and events* on page 127

Configuring email notifications

In addition to displaying a list of alarms and events in the KMC Connect Alarm Viewer, KMC Connect will send the same notification by email to individual operators. Use the Alarm Routing Manager to configure the operator notifications.

To open the routing manager, on the ribbon choose **Routing Manager** from the **Alarms** group.

Recipient operators

Recipient operators are key operators that must receive notifications by email. The names in this list are managed separately from the user names in Security Manager.

To add a name to the recipient operators list, do the following:

1. Right-click the operator list and choose **Add Operator** from the shortcut menu.
2. Enter the first and last name of the operator.
3. Choose **Close** when finished.

Alarm Message Delivery Details

Delivery Details designate the key operators that are notified by email of an alarm. Operators may have more than one entry in the delivery details list. For example, an operator may choose to have email messages sent to an office address during working hours and to a home address on nights and weekends.

To set the delivery details, do the following:

1. Choose an operator's name from the operator list.

2 Right-click and then select **Add e-mail** from the shortcut menu.

   a. Enter the email address.

   b. Enter a description for the delivery detail.

   c. Click **OK** when email entry is complete.

3 Click the schedule button and enter the times of day for the entry.

4 Click **Close** when finished.

Alarm Class

An Alarm Class a similar group of operators that need to receive alarm notification in email messages. Choose the alarm class for which operator notification is required.

1 Choose an operator's name from the operator list.

2 Choose the delivery method from the **Delivery Details** list.

3 Select one or more alarm classes in the **Alarm Class** list.

4 Click **Close** when finished.

Tip: This alarm class must also be selected in every object or point that originates an event for which this operator must be notified. See *Configuring devices for BACnet alarms* on page 105 for details about configuring devices and objects for event notification.

Section 20: Filtering alarms by security groups

Alarms in KMC Connect can be excluded from viewing by designated security user groups. This topic covers setting up alarm groups with the Alarm Group Manager.

Users in security groups can be included or excluded from seeing, archiving and acknowledging alarms.

- Use the Alarm Group Manager tool to set up filtering of alarms.
- When filtering is applied, specific security user groups are excluded from seeing alarms from designated devices, networks, or services.
- The procedure for setting up security user groups is described in the topic *Security and passwords* on page 65.

Before beginning the setup procedure, determine which devices, networks or services will require filtering.

- In BACnet internetworks the source of notification are devices that contain the notification class object that designates KMC Connect as the alarm handling device.
- In KMD networks the source of alarms are controllers that initiate alarms from Control Basic programs.

To set up alarm filtering do the following:

1. On the **Alarms** tab click **Group Manager**.
2. Use existing alarm exclusion category or add a new category by doing one of the following:
   - Click the category icon at the top of the page and enter a unique name for the category.
   - Right-click anywhere in the list and choose **New Category** from the shortcut menu. Enter a unique name for the group.
3. Select the alarm exclusion category and add a new exclusion group by doing one of the following:
   - Click the group icon at the top of the list and enter a unique name for the group. Enter a new name for the group. A group icon is added to the list.
   - Right-click anywhere in the list and choose **New Group** from the shortcut menu. Enter a new name for the group. A group icon is added to the list.

4 Drag a service, network or device from the Network Manager list to the exclusion group icon .

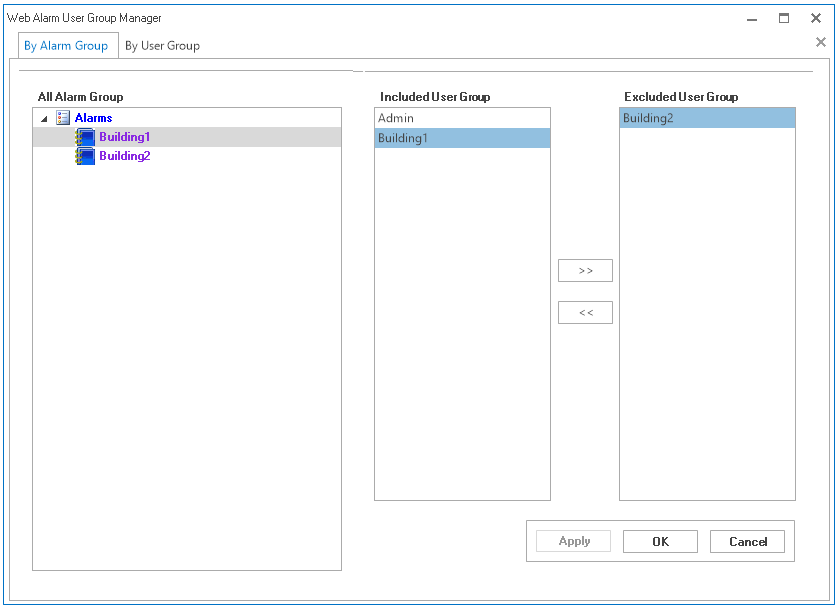
5 In the Alarm Group Manager list, right-click a group icon and choose **Exclude User Group** from the shortcut menu. The Alarm Group Manager dialog opens.

6 Select an exclusion group icon from the **All Alarm Group** list.

7 Select a user group from the **Include User Group** list.

8 Click » to move the group to the **Exclude User Group** list.

Illustration 20–1 Alarm Group Manager dialog



Section 21: Configuring the Notification Service

The KMC Connect Notification Service is the link between a BACnet building automation system and KMC Connect.

To process alarms, events and notifications originating in a KMC Connectmanaged building automation system, a Notification Service is installed as part of the KMC Connect installation. To change a property in the Notification Service, do the following:

1. Open the Network Manager.
2. Locate the Notification Service icon  in the Network Manager list.
3. Right-click the service and then choose **Configure Service** from the shortcut menu. A configuration tab opens in the workspace.
4. Make the changes to the service.
5. When changes are complete, click **Save Changes to Service** in configuration tab.

General Properties

Service Name A descriptive label of the service. Service Name is the name that identifies the service in the Network Manager list.

Service ID Service ID is assigned by KMC Connect and cannot be changed.

Service Host Address Not applicable to KMC Connect.

Service Host TCP Port Not applicable to KMC Connect.

Alarm Printer

As each alarm is received, it is printed on the selected printer. The printer must be installed as a Windows printer.

Email settings

To send email message to an operator, the Notification Service must be configured with the credentials for an email server. The credentials are supplied by the IT department and include the following items.

Outgoing Mail Server The address of an SMTP email server (bigbuilding.com).

Username A user account name (alarms@bigbuilding.com).

Password The password for the user account.

Domain The domain name of the server.

Port Typically this is TCP port 25. The IT department can verify this or supply the correct value.

Return Address The reply address is that is included as part of the message (maintenance@bigbuilding.com).

Note: Some virus detection programs will block KMC Connect from sending email messages because the messages are interpreted as spam. You may need to disable email protection in the virus detection program or enter KMC Connect as an exception.

KMC Connect

Part V: Schedules

Section 22: About KMC Connect managed schedules

In a KMC Connect managed building automation system, schedules are a convenient method for controlling recurring or future events without extensive programming. Topics in this section cover setting up schedules in KMC Connect.

Use schedule to change the value of one or more properties at the same time every day. Schedules can also be set up to override normal operation on holidays, plant closings, maintenance days or other special events.

- The weekly schedule in a schedule object sets the daily schedule for each day of the week.
- An exception schedule in a schedule object overrides the weekly schedule on special days.
- A calendar object can be designated by the exception schedule for special days that are common to two or more schedule objects.

Related topics

- *Configuring BACnet schedules* on page 145
- *Setting up a Calendar object* on page 148
- *Schedule object* on page 379
- *Calendar object* on page 321

Configuring BACnet schedules

BACnet schedule objects directly control the state of one or more BACnet objects—the objects in the Object Reference List—based on the times and values entered into the weekly schedule property.

For a complete description of the Schedule object properties, see *Schedule object* on page 379.

To set up a BACnet Schedule object, do the following.

1. In the Network Manager list, locate the device and then double click the Schedule object  icon.

2. Under the **General Properties**, open the schedule object tab and set the following properties:
   a. Enter a name and description for the object.
   b. Under Schedule Default, enter a default value and choose a data type.
   c. Drag one or more BACnet objects to the **Object Reference List**. The objects must be of the same type.
   d. Set the period for which the schedule is active with **Start Time** and **Stop Time**.
   e. Enter a value for **Schedule Default**.
   f. Enter a value for **Priority for Writing**.
3. Expand **Weekly Schedules** and add time-value pairs.
4. For special days such as holidays, special events or maintenance days, expand **Exceptions Schedule** and add dates and time-value pairs.
5. Click **Save Changes** at the top of the configuration tab.

Setting the Weekly schedule time value pairs

To add a time-value pair

To add a time value pair, do either of the following:

1. Click the add button  at the bottom of the list of time-value pairs. The new pair is added to the bottom of the list.
2. Drag the remove button  next to a time-value to any day in the schedule.

To duplicate a time-value pair

1. Right-click a time value pair and choose **Copy** from the shortcut menu.
2. Select a time-value pair for another day. Right-click and choose **Paste** from the shortcut menu.

To change time in a time-value pair

1. To change time, select the hours, minute or seconds value.
2. Click the up or down arrows  next to the time value.

To remove a time-value pair

Remove any time-value pair by clicking the remove button  .

Setting up the Exception schedule

On special days, an Exception schedule overrides the time-value pairs of the weekly schedule. Typically the special dates are holidays, special events, or other days that change the normal schedule in a Schedule object. A Calendar object can be designated by the Exception schedule for special days that are common to two or more schedule objects.

To add an exception schedule to a schedule object

1. In the Schedule object tab, expand **Exceptions Schedule**.
2. Click **Add** and then choose **Date**, **Date Range** or **Week and Day** from the shortcut menu.
3. Choose an Event priority for the Exception schedule.
4. Add time-value pairs. This is similar to adding time-value pairs in the Weekly schedule.

To add a Calendar object to the Exception schedule

For details on the Calendar object see the topic *Calendar object* on page 321.

1. In the Schedule object tab, expand Exceptions Schedule.
2. In Period Selection choose **Calendar Object**.
3. Choose and Event priority for the Exception schedule.
4. Add time-value pairs. This is similar to adding time-value pairs to the Weekly schedule.

Add a day to the Exception schedule

Use either of the following methods to add a day to the Exception schedule list.

- Click **Add** and then choose **Date**, **Date Range** or **Week and Day** from the shortcut menu.
- Right-click the list and then choose **Date**, **Date Range** or **Week and Day** from the shortcut menu.

Change an entry in the list

Use any of the following methods to change an entry in the list of days.

- Right-click an entry and then make a selection from the shortcut menu.
- Select the entry and then click **Edit**.
- Select the entry and then click **Remove**.

Setting up a Calendar object

A Calendar object is a standard BACnet object that contains a list of calendar dates. Typically the special dates are holidays, special events, or other days that need to be added to the exception schedule in a Schedule object. Add a Calendar object to the exception schedule in Schedule objects that share common exception dates.

To set up a Calendar object, do the following:

1. In the Network Manager list, locate the device and then select the Calendar object  icon.
2. Double-click the object icon to open it.
3. Add or delete dates as needed.
4. When finished, click **Save**.

To add a calendar entry to the list.

Use either of the following methods to add a calendar to the calendar list.

- Click **Add** and then choose **Date**, **Date Range** or **Week and Day** from the shortcut menu.
- Right-click the list and then choose **Date**, **Date Range** or **Week and Day** from the shortcut menu.

To change an entry in the list.

Use any of the following methods to change a calendar entry in the list.

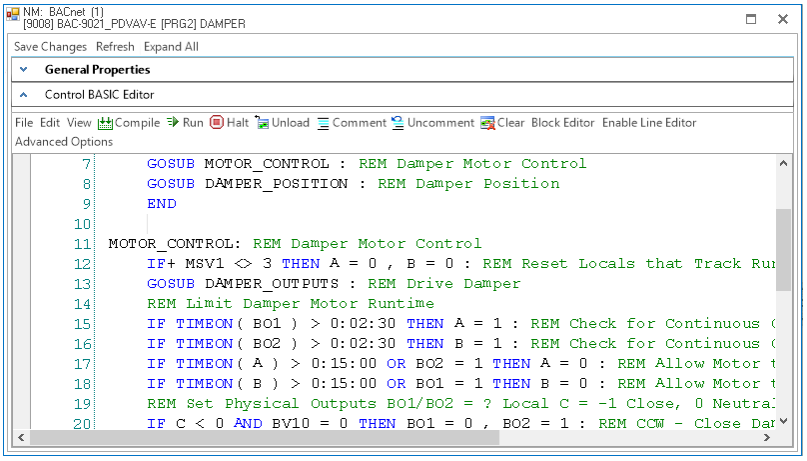
- Right-click an entry and then make a selection from the shortcut menu.
- Select the entry and then click **Edit**.
- Select the entry and then click **Remove**.

Related topics

- *Schedule object* on page 379
- *Calendar object* on page 321
- *About KMC Connect managed schedules* on page 145

KMC Connect

Part VI: Control Basic and the Code Editor



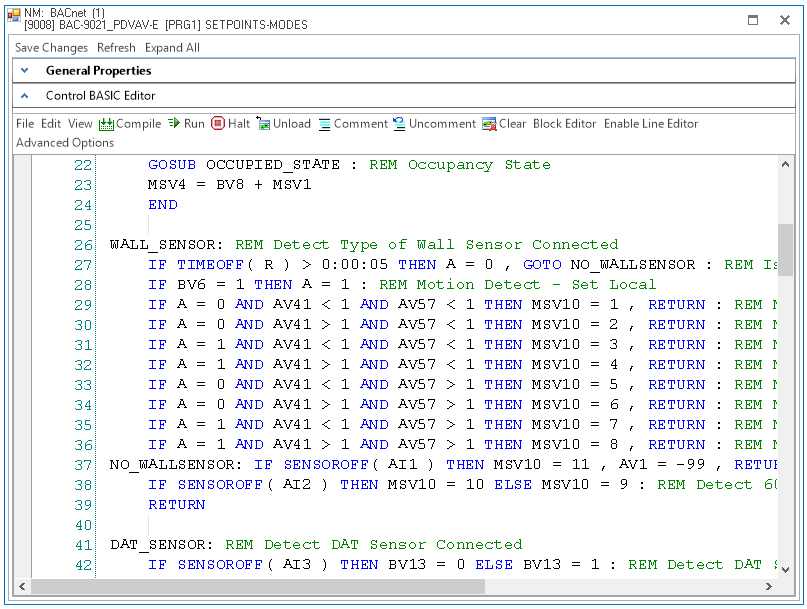
Section 23: Programming with the Code Editor

Topics in this section describe the KMC Connect Code Editor and using it to write Control Basic programs for KMC controllers.

Code Editor is the tool with which Control Basic programs are entered and edited in KMC controllers. Key features of the Code Editor include:

- Editing programs within KMD or BACnet controller from KMC Controls.
- Writing and editing programs while offline
- Automatic Control Basic line numbering
- Color-coded text
- Automatic keyword display.

Illustration 23–1 Code Editor tab



Related topics

- *About Control Basic programs* on page 165
- *Writing Control Basic programs in controllers* on page 152
- *Writing programs offline* on page 154
- *Automatically numbering program lines* on page 156
- *Keyword prompting* on page 156
- *Using keyboard shortcuts* on page 157
- *Finding and replacing text* on page 158
- *Changing Code Editor options* on page 162

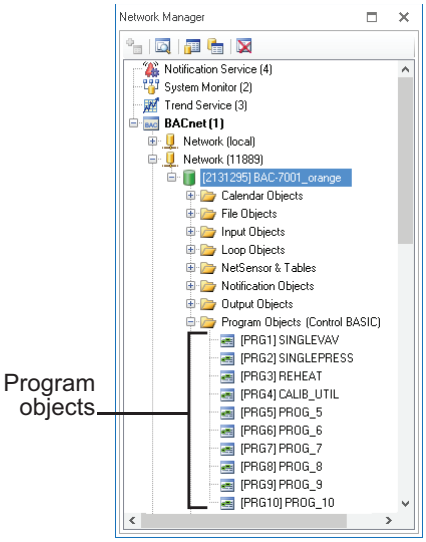
Writing Control Basic programs in controllers

Every KMC BACnet and KMD controller includes Control Basic programs. The programs are written and edited with the KMC Connect Code Editor.

Open an existing program for editing with the Code Editor by doing the following:

1. Locate a program icon in the Network Manager list.
2. Click the icon to open the program.
3. When editing is complete, save the work in the controller by doing one of the following:
   - Press CTRL+s.
   - Right-click in the editor window and choose **Save** from the shortcut menu.
   - From the Code Editor toolbar choose **File** menu choose **Save**.

Illustration 23–2 Control Basic objects within controllers



To save a save a Control Basic program as a local file

Local files have the extension of `.cb`. Save the program as local program file by doing one of the following:

- From the Code Editor toolbar choose **File** and then **Save As**.
- If you choose the **Save As** command, or if the file has never been saved, specify the new file name and location.

Code Editor toolbar items

Use the menus and buttons on the Code Editor toolbar to compile, save, start and halt Control Basic programs. Button functions are listed in the table *Code Editor toolbar items* on page 153.

Note: The Advanced Options are for special circumstances. Do not change unless directed by KMC Controls Technical Support department.

Illustration 23–3 Code Editor toolbar

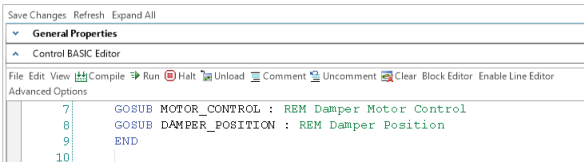


Table 23-1 Code Editor toolbar items

Action	Icon	Description
Compile		Tests the program for proper syntax but does not send it to the controller.
Run (BACnet only)		Compiles the program, saves it to the designated controller and runs the program in the controller.
Halt (BACnet only)		Stops the controller from running the program.
Unload (BACnet only)		Removes a Control Basic program from the queue in the controller.
Comment		Adds the REM keyword to the beginning of selected lines.
Uncomment		Removes the REM keyword from the beginning of selected lines.
Clear (BACnet only)		Erases the Control Basic program from the controller.
Block Editor		Opens the Block Programming Editor
Enable Line Editor		Unlocks the Block Editor generated Control Basic for copying or editing with the line editor.

Related topics

- *About Control Basic programs* on page 165
- *Writing programs offline* on page 154
- *Automatically numbering program lines* on page 156
- *Keyword prompting* on page 156
- *Using keyboard shortcuts* on page 157
- *Finding and replacing text* on page 158
- *Changing Code Editor options* on page 162
- *Writing block programs* on page 233

Writing programs offline

With the Code Editor you can write Control Basic programs without connecting to a site and controller. The Code Editor can open two types of offline Control Basic programs.

- Program objects that are part of `.bnd` and `.kmd` files that are located in the Resource Manager list. The program object or point is indicated with the program icon  .
- Control Basic `.cb` files that are typically stored in the Resource Manager list but may be located on any disk of the computer running Design Studio.

Start a new program

Start a new Control Basic program by doing the following:

1. Do either of the following to open the Code Editor workspace:
   - On the **File** menu, click **New** and then **Program File**.
   - Click  on the toolbar and then choose **New Program File**.

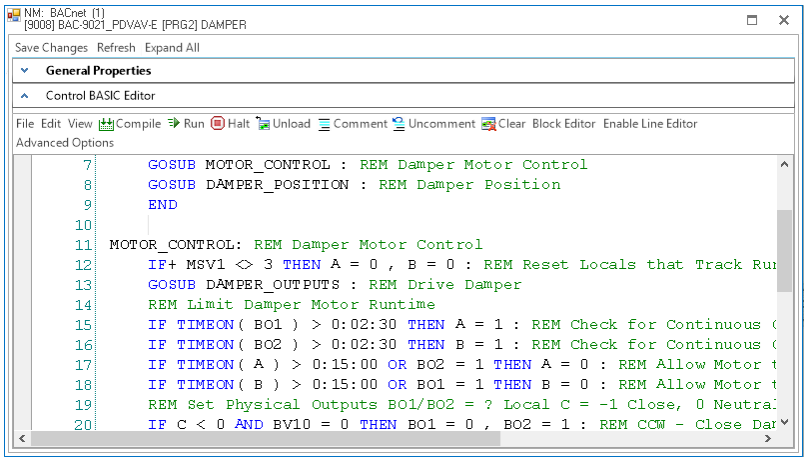
   An empty Code Editor window opens in the workspace.
2. Choose the following from the text boxes at the top of the Code Editor tab:
   - Model
   - Firmware version
   - Device ID
3. Add program statements and lines as required to operate the equipment connected to the controller.

4 Test the program for proper syntax by compiling the program. Make corrections as necessary.

5 Save the program by doing one of the following:
- Choose **Save** from either **File** menu.
- To save the file in a different folder or with a different name, choose **File** and then **Save As**. Using Save As always saves the file as a `.cb` file.

Note: Saving the file overwrites an existing file with the same name at the same location.

Illustration 23–4 A local file in the Code Editor



Edit an existing program

To use Code Editor to modify local `.CB`, `.BND` and `.KMD` files. do the following:

1 Open an existing Control Basic program by doing either of the following:
- On the **File** menu, click **Open** and then **Program File**. Locate a `.CB` file.
- In a Resource Manager locate a `.BND` or `.KMD` file and then double-click a program object or point icon .

2 When the program is open, add program statements and lines as required to operate the equipment connected to the controller.

3 Test the program for proper syntax by compiling the program. Make corrections as necessary.

4 Save the program by doing one of the following:
- Choose **Save** from the **File** menu.
- To save the file in a different folder or with a different name, choose **File** and then **Save As**. Using Save As saves the file as a `.CB` file.

Note: Saving the file overwrites an existing file with the same name at the same location.

Related topics

- *About Control Basic programs*
- *Writing Control Basic programs in controllers* on page 152
- *Automatically numbering program lines* on page 156
- *Keyword prompting* on page 156
- *Using keyboard shortcuts* on page 157
- *Finding and replacing text* on page 158
- *Changing Code Editor options* on page 162

Automatically numbering program lines

The Code Editor will automatically add numbers to program lines.

To turn automatic line numbering on or off do the following:

- From the **View** menu and then select or clear **Auto Line Number**.
- Right-click in the editor work space and then select or clear **Auto Line Number**.

Related topics

- *About Control Basic programs*
- *Writing Control Basic programs in controllers* on page 152
- *Writing programs offline* on page 154
- *Keyword prompting* on page 156
- *Using keyboard shortcuts* on page 157
- *Finding and replacing text* on page 158
- *Changing Code Editor options* on page 162

Keyword prompting

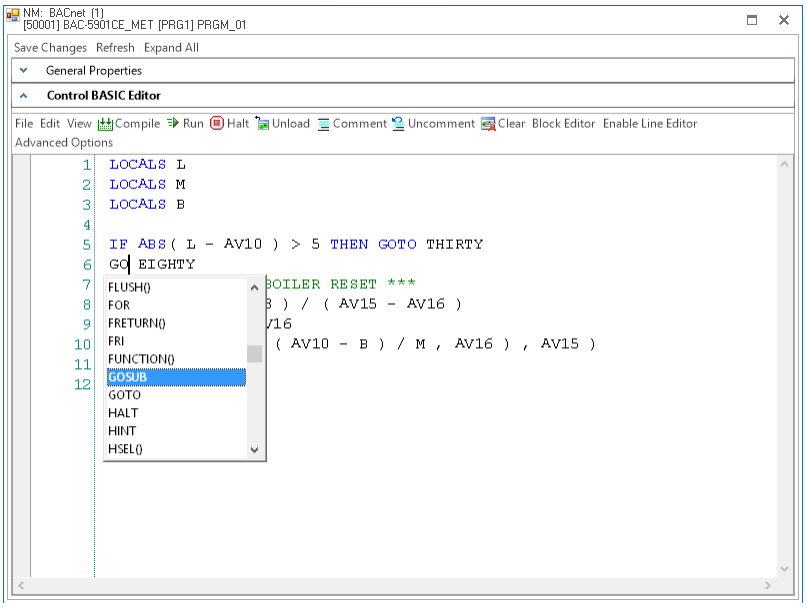
The Intellisense keyword filter speeds the writing of Control Basic programs. As you type program lines, a shortcut menu opens a list of keywords. The list starts at a keyword that matches the letters that you have already typed. For example, if you have typed *GO*, then the list opens at the keyword *GOSUB*. You can then choose either *GOSUB* or *GOTO* from the list.

To turn keyword prompting on or off, do either of the following:

- From the **View** menu select or clear **Keyword Prompt**.
- Right-click in the editor work space and then select or clear **Keyword Prompt** from the shortcut menu.

Tip: To change settings for keyword prompting, see *Changing Code Editor options* on page 162.

Illustration 23–5 Keyword prompting list



Related topics

- *About Control Basic programs* on page 165
- *Writing Control Basic programs in controllers* on page 152
- *Writing programs offline* on page 154
- *Automatically numbering program lines* on page 156
- *Using keyboard shortcuts* on page 157
- *Finding and replacing text* on page 158
- *Changing Code Editor options* on page 162

Using keyboard shortcuts

Use the shortcuts listed in the table *File menu shortcuts in Code Editor* when entering programs with the Code Editor.

- All of the shortcuts in the table are available by right-clicking in the Code Editor program listing.
- Some are available also through keyboard shortcuts.

Table 23-2 File menu shortcuts in Code Editor

Shortcut	Keyboard	Action
Renumber	CTRL+r	Renumbers all lines in the program.
Undo	CTRL+z	Reverses the last action.
Redo	CTRL+y	Repeats the last action.

File menu shortcuts in Code Editor (continued)

Shortcut	Keyboard	Action
Cut	CTRL+x	Removes the selected text and places it on the clipboard for pasting into another location.
Copy	CTRL+c	Copies the selected text to the clipboard for pasting into another location.
Paste	CTRL+v	Moves text from the clipboard to the selected location.
Select All	CTRL+a	Selects all text in the program.
Find/Replace	CTRL+f	Opens the Find and Replace dialog.
Go To	CTRL+g	Moves the insertion point to the specified line number.

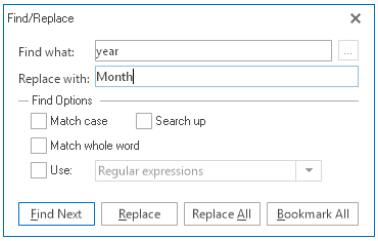
Related topics

- *About Control Basic programs* on page 165
- *Writing Control Basic programs in controllers* on page 152
- *Writing programs offline* on page 154
- *Automatically numbering program lines* on page 156
- *Keyword prompting* on page 156
- *Finding and replacing text* on page 158
- *Changing Code Editor options* on page 162

Finding and replacing text

Use Find and Replace in the Code Editor to locate and replace text and words in a Control Basic program. The search can be extended by using wildcards and regular expressions.

Illustration 23–6 Find/Replace dialog



Finding text and words

To quickly locate every occurrence of a specific word or phrase do the following:

1 From the **Edit** menu, choose **Find And Replace**.

2 In the **Find what** text box, enter the text to search for.

3 Select the check boxes for any other options.

4 Click **Find Next**.

Replacing text

You can automatically replace text— for example, you can replace *GOTO* with *GOSUB*.

1 From the **Edit** menu, choose **Find And Replace**.

2 In the **Find what** box, enter the text to search for.

3 In the **Replace with** box, enter the replacement text.

4 Select the check boxes for any other options.

5 Click **Find Next**, **Replace**, or **Replace All**.

Placing bookmarks

Bookmarks are temporary visual markers placed in Control Basic listings to help locate specific blocks of text. For example, when modifying a long program it may be helpful to bookmark a program lines that contain a reference to a specific object.

1 From the **Edit** menu, choose **Find And Replace**.

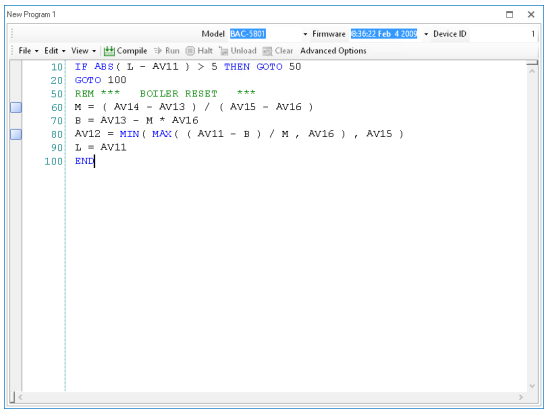
2 In the **Find what** box, enter the text to search for.

3 Click **Bookmark All**.

4 At the end of the search, each program line with the found text will have a bookmark icon in the left margin.

5 Once bookmarks are placed, use the Bookmark commands in the Edit menu to navigate to lines with bookmarks or manage bookmarks placed in program listings.

Illustration 23–7 Control Basic with bookmarks



Searching with wildcards

Use wildcards to find character patterns.

1 On the **Edit** menu, click **Find And Replace**.

2 Select the **Use** check box and then select **Wildcards**.

3 Enter a wildcard character in the **Find What** box by doing one of the following:
- Click the wildcard menu icon, click a wildcard character, and then type any additional text in the **Find what** box.
- Type a wildcard character directly in the **Find what** box. Wildcard character are described in the table *Wildcard characters* on page 160.

4 If you want to replace the item, enter the replacement text in the **Replace with** box.

5 Click **Find Next**, **Replace**, or **Replace All**.

When the **Use Wildcards** check box is selected, use the following wildcards to find character patterns.

Table 23-3 Wildcard characters

To find	Do this	Example
Zero or more characters, excluding spaces and punctuation	Enter an asterisk (*).	*f*t* finds words such as *fit* and *feet*.
Any single character, excluding spaces and punctuation	Enter a question mark (?).	*f??t* finds four-letter words that begin with f and end with t, for example, foot and feet.
Any string of numbers	Enter a (#)	*AV2#* finds *AV21* and AV23.

Searching with regular expressions

Use regular expressions to create sophisticated search patterns not possible with wildcards.

1 From the **Edit** menu, choose **Find And Replace**.

2 Select the **Use** check box and then select **Regular Expressions**.

3 Enter a regular expression in the **Find What** box by doing one of the following:
- Click the regular expression menu icon, click a regular expression, and then type any additional text in the **Find what** box.
- Type a regular expression directly in the **Find what** box. Regular expression codes are listed in the table *Regular expression codes* on page 161.

4 To replace the item, enter the replacement text in the **Replace with** box.

5 Click **Find Next**, **Replace**, or **Replace All**.

When the **Use Regular Expressions** check box is selected, use the following codes to find character patterns.

Table 23-4 Regular expression codes

To find	Do this	Example
Any single character except line break characters	Enter a period (.)	***a.cd*** finds ***abcd*** ***a..d*** finds ***abcd*** ***[a.cd]*** finds ***a***, c, d or a period (.)
Zero or more copies of the previous expression	Enter an asterisk (*).	***go*gol*** finds ***ggol***, ***gogol***, ***googol*** or ***gooogol***.
At least one copy of the previous expression.	Enter a plus sign (+).	***go+gol*** finds ***gogol***, ***googol***, ***gooogo***, but not ***ggol***.
The beginning of a line	Enter a caret (^).	***^f*** finds any word that starts with f at the beginning of a line.
One or more characters at the end of a line	Enter a dollar sign ($).	***f$*** finds any word that ends with f at the end of a line.
To find one or more characters at the beginning or end of a word	Enter \b	***\bsub*** finds any word that starts with ***sub*** such as ***subroutine*** ***sub\b*** finds words that end in ***sub*** such as ***GOSUB***
To find one or more characters preceded or followed by white space	Enter \s	***\ssub*** finds any word that has white space before ***subsub\s*** finds words that are followed by white space.
The end of a line	Enter \n	Finds the end of program lines.
Any one of several characters	Type the characters within brackets ([and]).	***[rml]ate*** finds ***rate***, ***mate***, and late, but not ***gate***.
Any one character not in a specified group of characters	Precede the characters in brackets with a caret (^).	***[^rml]ate*** finds ***fate***, ***gate***, and ***date***, but not ***rate*** or ***late***.
Either of two or more groups of characters	Enter a vertical bar (|) between character groups	***AV10|BV10*** finds both ***AV10*** and ***BV10***

Related topics

- *About Control Basic programs* on page 165
- *Writing Control Basic programs in controllers* on page 152
- *Writing programs offline* on page 154
- *Automatically numbering program lines* on page 156
- *Keyword prompting* on page 156
- *Using keyboard shortcuts* on page 157
- *Changing Code Editor options* on page 162

Changing Code Editor options

The Code Editor options are located under three separate tabs.

- The General Options tab controls Intellisense typing and line numbering options.
- The Fonts & Colors tab includes settings for the color and other text properties in the Code Editor window.
- Settings under Tabs control program formatting to enhance the readability of Control Basic.

To change the Code Editor appearance or options, do the following:

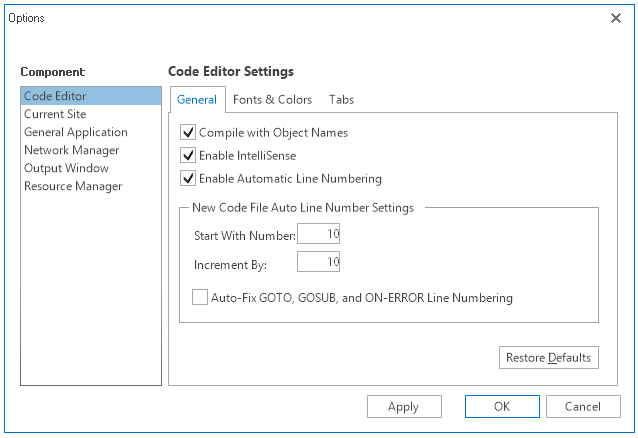
6 On the ribbon click the **File** tab and then click **Options**.

7 From the **Component** list choose **Code Editor**.

8 Choose a tab to make changes in the **Code Editor Settings** dialog.

9 Click **Apply** or **Ok** when finished.

Illustration 23–8 Code Editor Options dialog



General Options tab

Settings under the General Options tab controls Intellisense typing and line numbering options.

Code By Device Object Name When selected, object names my be used instead of mnemonics in Control Basic.

Turn Intellisense Keyword Filter On When selected, keywords are verified as they are entered from the keyboard.

Turn Intellisense Auto Line Numbering On When selected, automatically adds a line number at the start of a new line. The new line number is incremented by the value in **Increment By**.

Start With Number When new pages open, the first line number is automatically set to the value in **Start With Number**.

Increment By When **Turn Auto Line Number On** is selected, Code Editor adds a line number every time a new line is started. The value of the line number is the previous line number plus the value of **Increment By**.

Auto Fix When selected, references to lines by a *GOTO*, *GOSUB*, or *ON-ERROR* are automatically updated when new line numbers are inserted.

Restore to Default Click to return the Code Editor options to original settings.

Fonts and Colors tab

The colors and fonts for the text in the Code Editor window can be changed under the Fonts & Colors tab. Options are typical choices for text appearance in Windows programs. The text default colors are listed in the table, *Code Editor text colors*.

Table 23-5 Code Editor text colors

Text function	Default color
Plain Text	Black
Comments	Green
Keywords	Blue
Strings	Maroon

Tabs tab

Settings under Tabs changes the way Code Editor adds tab characters at the beginning of a new line. Tabs are invisible characters that span the space of five space characters. Using tabs enhances the readability of Control Basic but does not change the way a program compiles or runs.

Indenting When **Smart** is selected, the number of tab characters in **Tab Size** are automatically added to the beginning of a new line when ENTER is pressed.

Tab Size Specifies the number of tab characters to add to the beginning of a new program line. The minimum number of tabs is 4; the maximum is 80.

Related topics

- *About Control Basic programs* on page 165
- *Writing Control Basic programs in controllers* on page 152
- *Writing programs offline* on page 154
- *Automatically numbering program lines* on page 156
- *Keyword prompting* on page 156
- *Using keyboard shortcuts* on page 157
- *Finding and replacing text* on page 158

Section 24: About Control Basic programs

Control Basic is the process that creates the automation in KMC controllers. Topics in this section cover the rules for writing Control Basic programs.

Every KMC controller includes space for Control Basic programs. Within each controller a program continuously evaluates input data from the building automation system. Then, based upon the instructions in the program, the controller takes action to keep one or more pieces of equipment operating within required parameters.

The instructions within the program object are written in Control Basic, a programming language that is similar to BASIC (Beginner's All-purpose Symbolic Instruction Code). In addition to standard BASIC programming functions, it includes specialized functions specific for the building automation controls industry.

Each of the following topics cover a key aspect of Control Basic.

- *About Control Basic scans* on page 165
- *Programming format and notation* on page 169
- *Labels and line numbers* on page 167
- *Real numbers* on page 169
- *Hierarchy of operators* on page 170
- *Relational operators* on page 171
- *Using arithmetic operators* on page 172
- *Using Boolean logic* on page 172
- *Programming with variables* on page 173
- *Transferring values between BACnet controllers* on page 173
- *Programming with mnemonics* on page 174

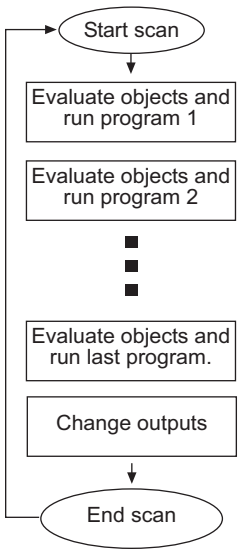
About Control Basic scans

Control Basic is the process that creates the automation in a KMC controller. Each controller has several program objects for storing and executing Control Basic instructions. When running Control Basic programs, the microprocessor in the controller does the following:

1. Evaluates the state of each object.
2. Executes the Control Basic programs.
3. Changes the state of all outputs when all programs have been executed.

This process—referred to as a scan—is normally performed several times a second. See the illustration *The scan process* on page 166 for an example of the scanning process.

Illustration 24–1 The scan process



The processor evaluates all program areas before making changes. For example, if programs 1, 3 and the last program includes instructions for Lights ON, and programs 2 and 4 had instructions Lights OFF, the lights will not flash, they will only be set to ON at the end of the scan.

Tip: Program the most important events in the highest numbered program area. This prevents programs with less importance from overriding critical actions.

Writing Control Basic statements

Use the Code Editor to write Control Basic programs.

- To change the program in a controller, see *Writing Control Basic programs in controllers* on page 152.
- To write a program without being connected to a controller, see *Writing programs offline* on page 154.
- To open a file in a BACnet controller, see the topic *Program object* on page 377.

Statements

A statement is the simplest instruction in a Control Basic program. Keywords such as INTERVAL, START, or GOTO are examples of statements. Expressions and functions are specific type of statements.

Multiple statements

Multiple statements can be used on the same program line, but must be separated by a colon.

```
A = B : GOTO 80 : START S
```

Functions

A function is a keyword that—when evaluated by Control Basic—returns a value. This returned value is the result computed by the function. Functions save time for complex calculations such as calculating square roots or averaging a set of values. They may also be used to retrieve common system data such using TIME to retrieve the time of day.

Functions are defined by either a keyword or by a user defined function as described in the topic *User defined procedures and functions* on page 176.

Expressions

A Control Basic expression describes a symbol or combination of symbols that represent a numeric value. Expressions may take the form of a simple equality such as `A=7` or a comparison between symbols such as `X < Y`. Expressions can be derived also from a function such as ***TIME***, a controller point or object such as Analog Input object AI2, or by the result of a series of calculations such as `A * B - AI2 - 2 / 9`.

An expression must evaluate to a real number.

Table 24-1 Examples of expressions

Expression	Example
Functions	TIME, DOW, DOY, etc.
Local Variables	A through Z
BACnet objects	AI1, BI1, AO2, BO4, etc.
KMD Controller Points	OUT1, IN3, VAR16, etc.

Labels and line numbers

Depending on the model of the controller, Control Basic programs will use either line numbers or labels. See the topic *Control Basic versions in controllers* on page 460 for a complete list of model numbers.

Standard Control Basic

- BAC-7000 series controllers
- BAC-5800 series controllers

Next Generation Control Basic

- BAC-A1616 Building controller
- BAC-10000 FlexStat controllers
- Conquest series controllers

Line numbers–Standard Control Basic only

When writing programs for controllers that support Standard Control Basic, enter a line number at the beginning of each line. Each Control Basic program line must include a line number and at least one function or statement.

```
10 A = 2 + 3
20 P = PI
30 REM Program starts here
```

Programs written for controllers with Next Generation Control Basic do not use program lines.

Labels in Next Generation Control Basic

Labels are used instead of line numbers with Next Generation Control Basic. Use labels when redirecting program flow with the following statements:

- *GOSUB* on page 196
- *GOTO* on page 197
- *ONERROR* on page 212
- *ON GOSUB* on page 210
- *ON GOTO* on page 211

Declare a label by typing a name followed immediately by a colon (:).

- A label can be any combination of letters (A-Z or a-z), numbers (0-9) or the underscore (_).
- Labels are not case sensitive.
- Labels are unique to the program in which they are declared.
- A label cannot duplicate a keyword, constant, local variable or alias.

In the following program example, `CoolMode` and `HeatMode` are program labels.

```
IF T > 55 THEN GOTO CoolMode
IF T <= 55 THEN GOTO HeatMode
END
CoolMode:
REM Cooling sequence runs here
END
HeatMode:
REM Heating sequence runs here
END
```

Programming format and notation

Control Basic programs consist of a series of numbered lines. On each line there are one or more statements.

Throughout these instructions the following notations are used to describe formats:

Table 24-2 Typographic conventions

CAPS	Words in capital letters are key words and must be entered as shown.
lowercase	Items in lowercase letters represent information such as expressions that you supply.
...	An ellipsis (...) indicates that an item may be repeated as many times as necessary.
spaces (_)	Required spaces in syntax are illustrated with an underline (_).
:	A colon (:) separates statements on the same line.
[ ]	Optional items are shown in brackets [ ].
	All other punctuation, including commas, are part of the syntax and must be included as shown in each example.

Real numbers

Real numbers are any logical number between $-3.4 * 10^{38}$ and $3.4 * 10^{38}$. Notation of the number is recognized in any of the following formats:

- Whole numbers (100)
- Decimal format (.0000123)
- Engineering notation (7.879 E-12)

Hierarchy of operators

Control Basic arithmetic operators have an order of precedence. When several operation take place in the same program statement, some operations are performed before others. Control Basic uses the operator-precedence shown in the illustration *Order of operator precedence* on page 170 when performing operations on an expression. Operations at the same level of precedence are evaluated from left to right.

Illustration 24–2 Order of operator precedence

Operator	Type	Precedence
( )	Expression in parenthesis	Highest (performed first)
NOT	Logical NOT	
^	Exponentiation	
*, /	Multiplication and division	
\	Integer division	
MOD	Modulus (remainder)	
+, -		
<, >, <=, >=	Relational	
=, <>		
AND	Logical	
OR		
XOR		Lowest (performed last)

Related topics

- *Using arithmetic operators* on page 172
- *Relational operators* on page 171
- *Using Boolean logic* on page 172

Relational operators

Relational operators are used to compare two values. The result is *TRUE* if the comparison is *TRUE*; otherwise, the result equals *FALSE*. This result can then be used to make a decision regarding program flow. All relational operators have the same level of precedence.

Table 24-3 Relational operators

Operator	Relation Tested	Example	Result
=	Equality	5 = 2	False
<>	Inequality	5 <> 2	True
<	Less than	5 < 2	False
>	Greater than	5 > 2	True
<=	Less than or equal to	5 <= 2	False
>=	Greater than or equal to	5 >= 2	True

Related topics

- *Using arithmetic operators* on page 172
- *Using Boolean logic* on page 172
- *Hierarchy of operators* on page 170

Using arithmetic operators

Operators are listed in their order of priority. The formats for using operators are listed in the table *Arithmetic order of precedence* on page 172.

Table 24-4 Arithmetic order of precedence

Symbol	Operation	Example
*	Multiplication	2*4=8
/	Division	10/4 =2.5
\	The integer portion of a division	13\5= 2
MOD	The remainder of a division	13 MOD 5=3
+	Addition	2+2=4
-	Subtraction	4-3=1
^	Exponentiation Raises a value to a power	A = AI1 ^ AV1

Related topics

- *Relational operators* on page 171
- *Using Boolean logic* on page 172
- *Hierarchy of operators* on page 170

Using Boolean logic

Control Basic recognizes four logical operators. The operators are listed in their order of precedence.

NOT NOT is a Boolean operator that performs a logical NOT operation on an expression. If the expression is 0, the result is 1. If the expression is non-zero, the result is 0.

For additional details on this operator, see the keyword *NOT* on page 210.

AND AND performs the logical *AND* of the two expressions. The result is *TRUE* if both expressions are non-zero; otherwise, the result is *FALSE* .

For additional details on this operator, see the keyword *AND* on page 184.

OR OR performs the logical *OR* of the two expressions. The result is *TRUE* if either expression is *TRUE*. The result is *FALSE* if both expressions are *FALSE*.

For additional details on this operator, see the keyword *OR* on page 213.

XOR XOR performs the logical ***exclusive or*** of the two expressions. The result is ***TRUE*** if the two expressions are different; otherwise, the result is ***FALSE***.

For additional details on this operator, see the keyword ***XOR*** on page 232.

Related topics

- *Using arithmetic operators* on page 172
- *Relational operators* on page 171
- *Hierarchy of operators* on page 170

Programming with variables

Variables are place holders for information such as setpoints, time delays, and operating modes. Control Basic uses two types of variables, value objects, and local variables.

BACnet value objects as variables

Analog, binary, and multistate value objects are used as program variables in BACnet controllers..

Local variables

Local variables can only be used within the Control Basic program that refers to them. The values they represent cannot be directly transferred to other Control Basic programs. Local variables are useful for counters or to store the results of local calculations.

Standard Control Basic local variables Within each Control Basic program there are 26 local variables. These variables are assigned the letters of the alphabet (A-Z).

Next Generation and Generation 5 Local variables in controllers that use Next Generation or Generation 5 Control Basic can be either a single-letter variable (A-Z) or a declared local variable. Variables are declared with the commands INTEGER, LOCALS, REAL, or STRING and must be declared within each Control Basic program. All single letter local variables are automatically declared unless a local variable is declared in which single letter variables must also be declared also.

For details on using and declaring local variables, see the following keywords.

- *LOCALS* on page 204
- *INTEGER* on page 201
- *REAL* on page 217
- *STRING* on page 226

Transferring values between BACnet controllers

BACnet controllers from KMC Controls can read from and write to the present values in objects in other controllers on the internetwork.

Reading properties from other BACnet devices

With Control Basic, a KMC BACnet controller can request the present value from any other controller on the BACnet internetwork. However, the following limitations must be observed.

- Each controller can request a present value from no more than 32 other devices.
- Each controller can request up to a total of 64 present values from the 32 devices.

For example, controller 50 can request two values each from controllers 1-32, four values each from controllers 1-16 or eight values from panels 1,5,6,8 and thirty-two values from panel 9.

To read a value from another panel, you must know the panel's device instance. The instance is separated from the value by a period (.).

Caution

In the following example, the WAIT statement is required. Do not delete it or the program will not run correctly.

Syntax: *device instance.object*

```
REM * POINTS TRANSFERRED FROM DEVICE 1213
REM * OUTSIDE AIR TEMPERATURE *
AV24 = 1213.AI7
WAIT 0:05:00
END
```

Writing values to other controllers

To change the value in another BACnet device, you must know the devices instance number. The instance is separated from the value by a period(.).

Syntax: *device instance.object*

```
201.AO1 = AV1
```

Programming with mnemonics

Mnemonics are a short, easy to remember abbreviations in Control Basic that refer to objects. For example, a physical input is entered as ***AI1*** or ***BI1*** in BACnet controllers instead of typing ***Input1***.

Control Basic mnemonics for BACnet objects are listed in the table *Control Basic mnemonics for BACnet objects* on page 175. The following line of Control Basic is an example of using mnemonics to refer to an analog input object and a binary output object.

```
If AI08 > 10 Then Start BO2
```

- Mnemonics listed as ***Read Only*** can read a value— such as its value or status— from that object.
- Mnemonics listed as ***Read and Write*** describes a property—such as its value or status—that may be changed through programming or by direct access.
- Mnemonics are reserved words and cannot be used for object names or descriptions.

Caution

If a Control Basic program uses a mnemonic to refer to an invalid local object or property within an object, the program will compile but it will halt execution. The reason for the halt is listed in the Description of Halt column of the Program Objects dialog.

Table 24-5 Control Basic mnemonics for BACnet objects

Object type	Mnemonic	Property	Action
Accumulator	ACC#	Present Value	Read Only
Accumulator	ACC#.PR	Pulse Rate	Read Only
Analog Input	AI#	Present value	Read and Write
Analog Input	AI#.LL	Low Limit	Read and Write
Analog Input	AI#.HL	High Limit	Read and Write
Analog Output	AO#	Present value	Read and Write
Analog Output	AO#.LL	Low Limit	Read and Write
Analog Output	AO#.HL	High Limit	Read and Write
Analog Value	AV#	Present value	Read and Write
Analog Value	AV#.LL	Low Limit	Read and Write
Analog Value	AV#.HL	High Limit	Read and Write
Binary Input	BI#	Present value	Read and Write
Binary Output	BO#	Present value	Read and Write
Binary Value	BV#	Present value	Read and Write
Loop	LOOP#	Present value	Read Only
Loop	LOOP#.B	Bias value	Read and Write
Loop	LOOP#.D	Derivative constant	Read and Write
Loop	LOOP#.P	Proportional constant	Read and Write
Loop	LOOP#.I	Integral constant	Read and Write

Table 24-5 Control Basic mnemonics for BACnet objects (continued)

Object type	Mnemonic	Property	Action
Loop	LOOP#.SP	Setpoint	Read and Write
Multi-State Input	MSI#	Present value	Read and Write
Multi-State Output	MSO#	Present value	Read and Write
Multi-State Value	MSV#	Present value	Read and Write
Schedule	SCHED#	Present value	Read Only
Trend	TL#.EN	Log enable	Read and Write

User defined procedures and functions

User defined functions and procedures are programming techniques that can improve programs in the following ways.

- They divide long and complicated programs into smaller, more manageable modules.
- They are a method to avoid repeated calculations with the same program steps within a Control Basic program.

Note: Procedures and functions apply only to controllers with Generation 5 Control Basic.

See the related keywords ***PROCEDURE*** on page 215 and ***FUNCTION*** on page 195.

Rules for functions and procedures.

- All functions and procedures must have unique names. The names cannot duplicate Control Basic keywords or object names or descriptions.
- If you use a function or procedure, then all Control Basic statements in that program object must be part of a function or procedure.
- If a program object includes a function or procedure it must also include one procedure named ***main***. The main procedure represents the starting point of the program.
- Function and procedure declarations cannot be nested inside other functions or procedures.
- Functions and procedures can call other functions and procedures.
- A function or procedure ***cannot*** use a GOTO, ON GOTO, GOSUB, or ON GOSUB that branches to a location outside of the function or procedure.
- A function or procedure cannot have an IF or FOR that crosses the boundaries of a function or procedure.
- Local variables declared inside of a procedure or function are not accessible outside of the procedure.
- The number of procedures and functions in a program object is limited only by available memory.

Functions A function is a self-contained block of statements that return a single value to the point from where it was called or referenced.

Functions are called from within an expression in place of a variable or constant by specifying the keyword *FUNCTION* followed by a list of arguments. The arguments are enclosed in parentheses and separated by commas. When a function is called, the values of the arguments are passed to the local variables in the function in the same order that they occupy in the list. The names of the arguments in the function reference need not be the same as those in the function definition. The arguments can be expressed as constants, variables, or expressions. However, the number of arguments must be the same and the data types of the arguments must match.

When all of the statements have been executed, control is returned to the statement immediately following the point from where the function was called.

Example of a function In the following example, local variable DialSp will always fall within the range set by constants LoCoolingSp and HiHeatingSp.

```
CONST LoCoolingSp = 68
CONST HiHeatingSp = 75
REM AI1 is the septoint on a sensor dial.

FUNCTION validateSp(s, l, h)
  s = MAX(s, l)
  s = MIN(s, h)
  FRETURN s
ENDFUNC

PROCEDURE main()
  LOCALS DialSp
  DialSP = validateSp(AI1, LoCoolingSP, HiHeatingSP)
ENDPROC
```

Procedures A procedure is a self-contained block of statements that perform a task. A procedure is called by name from any location except from within an expression.

Procedures are defined with the keyword *PROCEDURE*, followed by a name for the procedure. The procedure definition may also include one or more arguments which are enclosed in parentheses and separated by commas. When a procedure is called, the values of the arguments are passed to the local variables in the procedure in the same order that they occupy in the list. The names of the arguments in the procedure reference need not be the same as those in the procedure definition. The arguments can be expressed as constants, variables, or expressions. However, the number of arguments must be the same and the data types of the arguments must match.

When all of the statements have been executed, control is returned to the statement immediately following the point from where the procedure was called.

Example of a procedure In the following example, the lowest of two temperatures is saved in Analog Value object AV1 to use as the outside air temperature.

```
LOCALS oat1, oat2
PROCEDURE lowOAT(a,b)
  AV1 = MIN(a,b)
ENDPROC

PROCEDURE main()
  oat1 = AI1
  oat2 = AI2
  lowOAT(oat1,oat2)
ENDPROC
```

Arrays and dynamic access

Controllers that support Generation 5 Control Basic support arrays and dynamic access. For a list of controllers, see the topic *Control Basic versions in controllers* on page 460.

Arrays An array is a set of values—such as the highest daily temperature—that are logically related to each other. The individual values are called the elements of the array. You can then refer to individual elements by the array name and a number—the index—to tell them apart. The elements are numbered starting at zero(0) up to the size of the array.

Dynamic access Dynamic access is similar to using an array but uses an index to point to either a BACnet object or device instead of pointing to an array element.

In the following example, Control Basic stores the last seven readings of Analog Input object AI3 in a seven element array and Analog Value objects 1-7. The oldest reading is stored in array element zero(0); the newest is in array element 6.

- A seven element array (`seventemps`) and a local variable (`x`) are declared.
- A FOR TO NEXT loop moves array elements 1-6 to elements 0-5
- The newest temperature reading from Analog Input object AI3 is stored in element 6.
- A second FOR TO NEXT loop moves the values from array elements 0-6 to Analog Value objects 1-7.
- Finally, the program waits for 30 seconds before repeating the process.

```
REAL seventemps[7]
INTEGER x

FOR x = 0 TO 5
  seventemps[x] = seventemps[x + 1]
NEXT x

seventemps[6] = AI3

FOR x = 0 TO 6
  AV[x+1] = seventemps[x]
NEXT x

WAIT(00:00:30)
```

To declare arrays, see the following topics.

- *LOCALS* on page 204
- *INTEGER* on page 201
- *REAL* on page 217
- *STRING* on page 226

Data types

A data type is a classification identifying a type of data, such as integer, real, or string that determines the possible values for that type. It also defines the operations that can be done on values of that type, the meaning of the data, and the way values of that type can be stored and used by properties in controllers.

Integer Holds a signed 32-bit (4-byte) integer in value from -2,147,483,648 through 2,147,483,647.

Real Holds a signed IEEE 32-bit (4-byte) single-precision floating-point number ranging in value from -3.4028235E+38 through -1.401298E-45 for negative values and from 1.401298E-45 through 3.4028235E+38 for positive values.

String Holds sequences of the ASCII letters and symbols on a standard U.S. keyboard.

See the following topics for a complete description and the procedure to declare variables as the following data types.

- *INTEGER* on page 201
- *STRING* on page 226
- *REAL* on page 217

Section 25: Keywords for Control Basic

This section covers the keywords for the Control Basic programming language.

The Control Basic keywords for operators, commands, and functions are reserved by Control Basic. They may not be used for descriptors, labels or names of objects, variables, or procedures.

To refer to BACnet objects in Control Basic, see the topic *Programming with mnemonics* on page 174.

Using example programs from help

You can use example programs listed in this document. Highlight the example and then copy the example and paste it into a Control Basic program.

Syntax for commands and functions

Required spaces are shown with underscore marks (_) and indicate that a space must be included for proper syntax. Optional items are shown in brackets [ ].

ABS

This function returns the absolute value of the expression. The expression can be a single number or the result of a calculation.

Syntax: *ABS(_expression_)*

Standard Control Basic example

Returns ***2.3***, the absolute value of -2.3.

```
A = ABS( -2.3 )
```

Returns the absolute value from the result of the calculation.

```
C = ABS( AV1 - AI1 )
```

Next Generation Control Basic example

Returns ***2.3***, the absolute value of -2.3.

```
A=ABS(-2.3)
```

Returns the absolute value from the result of the calculation.

```
C=ABS(AV1 - AI1)
```

ALIAS

This statement declares a local variable and dynamically binds the value of a property to that variable. ALIAS is slightly different between Next Generation and Conquest compatible controllers.

- For the KMC Controls Building Controller and FlexStat, see *ALIAS–Next Generation*.
- For Conquest controllers, see *ALIAS–Generation 5* on page 183.

See also the related topic *FLUSH* on page 194.

Note: For accessing off-panel points with Generation 5 Control Basic compatible controllers, *NETPOINT* is the preferred function.

ALIAS–Next Generation

ALIAS declares a local variable and dynamically binds the value of a property to that variable. It also sets two intervals at which Control Basic will read from or write to the property bound to the variable.

Syntax: *ALIAS(device, object, property, local, read interval, write interval)*

Note: Next generation Control Basic only.

The same point may be bound to an ALIAS in more than one program. However, the lowest read or write interval of all ALIAS statements within the device is used in all programs.

Table 25-1 Parameter for Next Generation ALIAS

Parameter	Description	Comments
device	The device instance number	Enclose the name of a device with quotation marks (“). Names are case sensitive.
object	A valid mnemonic	See *Programming with mnemonics* on page 174.
property	The property and priority for writing	Priority is ignored for read only objects such as inputs.
local	The local name to use within the program	Use as a local variable within the Control Basic program in which the alias is declared.
read interval	The interval, in seconds, at which Control Basic will read the property	To never read from the object, use ***NONE***. The default value is 60 seconds.
write interval	The interval, in seconds, at which Control Basic will write to the property	To never write to the object, use ***NONE***. The default value is ***NONE***.

In the following example Control Basic binds the present value of Binary Output object BO1 in device 1212 to the local name ***Lights***. Control Basic reads the value of BO1 once an hour and writes the value every 60 seconds.

```
ALIAS(1212, BO1, PV@4, Lights, 1:00:00, 60)
```

The last example binds the value of the Analog Input object AI2 to the local variable ***OutsideAirTemp***. When the device argument is omitted, Control Basic binds the local variable to the device with the lowest device instance that contains an Analog Input object AI2.

```
ALIAS("", AI2, PV, OutsideAirTemp, 100, NONE)
```

ALIAS–Generation 5

This statement declares a local variable and dynamically binds the value of a property to that variable.

Syntax: *ALIAS(device, object, property, local)*

Note: Generation 5 Control Basic only.

The same point may be bound to an ALIAS in more than one program. See the related topic *FLUSH* on page 194.

Table 25-2 Parameters for Generation 5 ALIAS

Parameter	Description	Comments
device	The device instance number	Enclose the name of a device with quotation marks (“). Names are case sensitive.
object	A valid mnemonic	See *Programming with mnemonics* on page 174.
property	The property and priority for writing	Priority is ignored for read only objects such as inputs.
local	The local name to use within the program	Use as a local variable within the Control Basic program in which the alias is declared.

In the following example Control Basic binds the present value of Binary Output object BO1 in device 1212 to the local name ***Lights***.

```
ALIAS(1212, BO1, PV@4, Lights)
```

The last example binds the value of the Analog Input object AI2 to the local variable ***OutsideAirTemp***. When the device argument is omitted, Control Basic binds the local variable to the device with the lowest device instance that contains an Analog Input object AI2.

```
ALIAS("", AI2, PV, OutsideAirTemp)
```

AND

AND is a Boolean operator that performs the logical ***AND*** of two expressions. The result is ***true*** if both expressions are non-zero; otherwise, the result is ***false***.

Syntax: *result = expression1 AND expression2*

In the following example, local variable ***C*** will always equal 1 as long as both local variables ***A*** and ***B*** = 1

```
A = 1 : B = 1 : C = A AND B
```

See the related topic *Using Boolean logic* on page 172.

ARCCOS

Returns the arccosine of the specified angle. ***Angle*** is expressed in radians.

Syntax: *ARCCOS(angle)*

```
A = ARCCOS( AV1 )
```

See the related topic *COS* on page 187.

ARCSIN

Returns the arcsine of the specified angle. ***Angle*** is expressed in radians.

Syntax: *ARCSIN(angle)*

```
A = ARCSIN( AI8 )
```

See the related topic *SIN* on page 224.

ARCTAN

Returns the arctangent of the specified angle. ***Angle*** is expressed in radians.

Syntax: *ARCTAN(angle)*

```
A = ARCTAN( AV12 )
```

See the related topic *TAN* on page 228.

AVG

This statement returns the average value of the items enclosed in parenthesis. In the following example, local variable ***D*** equals the average of analog inputs 1, 3 and 6.

Syntax: *AVG(_expression_,_expression_...)*

Standard BACnet Control Basic example

```
D = AVG( AI1 , AI3 , AI6 )
```

Next Generation Control Basic example

```
D=AVG(AI1, AI3, AI6)
```

BIND

Binds a BACnet device instance to a physical network address. This is typically used to bind an MS/TP slave device to a master device.

Syntax: *BIND (device, network, mac, option)*

Note: For Next generation Control Basic only.

Table 25-3 BIND parameters

Parameter	Description	Comments
device	The instance number of the device.	
network	The number of the BACnet network on which the device resides	May be expressed as decimal or hexadecimal notation. Use zero (0) as the local network.
mac	The MAC address of the device.	

Table 25-4 BIND options

Option	Description
Hint	Sets the default address but uses whatever can be found by the controller. This is the default state.
Locked	Forces the default address back to this every time it changes.

Examples:

```
BIND (550013,1,13)
BIND(123456, 678, 0x24 )
BIND(123456, 0x44, 09:88:77:55:44:55 )
BIND(123456, 0x4, 10.1.2.3:678 )
Bind(123456, 0x4, 10.1.2.3:678 , LOCKED)
BIND(123456, 0, 10.1.2.3:678 , HINT )
```

CLEAR

Resets the value of all local variables—variables labeled A-Z and declared variables—to zero.

```
CLEAR
```

CLOSE

Sets the value of a named point, binary output or value object to ***off***.

Syntax: *CLOSE_point*

Standard BACnet Control Basic example

```
CLOSE BO2
CLOSE A
```

Next Generation Control Basic example

```
CLOSE BO2
CLOSE A
```

Related topics

- *OPEN* on page 212
- *START* on page 225
- *STOP* on page 226

CONST

Use to declare a variable and assign to it a fixed value. Do not use with variables that change with subsequent steps in the program.

Syntax: *CONST, variable[, variable, ...]*

Note: For Next generation Control Basic only.

- Constants must be declared before they are used in a program. A constant may be declared anywhere in the program but typically it is at the beginning of the program.
- Must start with a letter A-Z, a-z, or an underscore (_). Constants are not case sensitive.
- Can be any combination of letters (A-Z or a-z), numbers (0-9) or the underscore (_).
- A constant may be used only within the program in which it is declared.
- A constant cannot duplicate a keyword, local variable, label or alias.

```
CONST Freeze = 32
CONST Boiling = 212
```

To declare local variables or arrays, see the following topics:

- *INTEGER* on page 201
- *LOCALS* on page 204
- *REAL* on page 217
- *STRING* on page 226

COS-1

Returns the arccosine of the specified angle. ***Angle*** is expressed in radians.

Syntax: *COS-1(_angle_)*

Note: Deprecated for BACnet controllers. See the keyword *ARCCOS* on page 184.

COS

Returns the cosine value of a specified angle. ***Angle*** is expressed in radians.

Syntax: *COS(_angle_)*

BACnet examples

Standard BACnet Control Basic example

```
10 A = COS( AV1 )
```

Next Generation Control Basic example

```
A = COS(AV1)
```

COV

This command sets up an object and property for a COV (Change of Value) subscription. The object must be in a device that supports COV subscriptions. It applies to all read accesses in a program regardless of location.

Syntax: *COV(device.object.property, increment , confirmed, lifetime)*

Note: See the topic *NETPOINTCACHE* on page 209 to enable or disable reading by COV.

Note: The smallest read interval of all COV statements within the device is used in all programs within the same controller.

```
COV(5.AV1.PV,200,4,1)
a = 5.AV1
```

Table 25-5 NETPOINT parameters

Parameter	Description	Comments
device	The device instance number of the device that contains the subscribed object.	
object	A valid Control Basic object mnemonic.	See *Programming with mnemonics* on page 174.
property	Present value (PV) is typically the property.	Priority is ignored for read only objects such as inputs.
increment	The increment of change that the subscribed object will report.	
confirmed	Enter 0 (zero) for a confirmed COV and 1 (one) for an unconfirmed subscription.	
lifetime	The time, in seconds, for the lifetime of the COV subscription.	

DEC

Decrements the value of ***point*** by the value of ***step***. If ***step*** is omitted, the ***step*** value is 1.

Syntax: *DEC(_point_,_step_) DEC(_point_)*

See the related topic *INC* on page 200.

Standard BACnet Control Basic example

```
10 DEC( AV1 , A + B )
20 DEC( AV2 )
```

Next Generation Control Basic example

```
DEC( AV1 , A + B )
DEC( AV2 )
```

DEWPOINT

Returns the dew point in degrees Fahrenheit based on Outside Air Humidity (OAH) and Outside Air Temperature (OAT). OAT is in degrees Fahrenheit.

Syntax: *DEWPOINT(_OAH_,_OAT_)*

See the related topic *DEWPOINTSI* on page 189 to express temperature in degrees Celsius.

Standard Control Basic example

```
D = DEWPOINT( AI1 , AI2 )
```

Next Generation Control Basic example

```
D=DEWPOINT(AI1, AI2)
```

DEW-POINT

Returns the dew point in degrees Fahrenheit based on Outside Air Humidity (OAH) and Outside Air Temperature (OAT). OAT is in degrees Fahrenheit.

Syntax: *DEW-POINT(_OAH_,_OAT_)*

Note: Deprecate for BACnet controllers. See the keyword *DEWPOINT* on page 189.

DEWPOINTSI

Returns the dew point in degrees Celsius based on Outside Air Humidity (OAH) and Temperature (OAT). OAT is in degrees Celsius.

Syntax: *DEWPOINTSI(OAH , OAT)*

See the related topic *DEWPOINT* on page 189 to express temperature in degrees Fahrenheit.

```
D=DEWPOINT(AI1, AI2)
```

DEW-POINT-SI

Returns the dew point in degrees Celsius based on Outside Air Humidity (OAH) and Temperature (OAT). OAT is in degrees Celsius.

Syntax: *DEW-POINT-SI(_OAH_,_OAT_)*

Note: Deprecated for BACnet controllers. See the keyword *DEWPOINTSI* on page 189.

```
D = DEW-POINT-SI( VAR1 , VAR2 )
```

DISABLE

DISABLE sets the value of a point, which can be the present value of an input, output or value object, to *off*.

Syntax: *DISABLE_point*

Standard BACnet Control Basic example

```
DISABLE AO1
DISABLE A
```

Next Generation Control Basic example

```
DISABLE AO1
DISABLE A
```

Related topics

- *ENABLE* on page 191
- *START* on page 225
- *STOP* on page 226

DOM

Returns the current day of the month.

Standard BACnet Control Basic example

```
IF+ DOM = 15 THEN 20 ELSE END
REM Continue program execution
```

Next Generation Control Basic example

```
IF+ DOM=15 THEN GOTO Continue ELSE END
Continue:
REM Continue program execution
```

DOW

Returns a numerical value for the day of the week.

BACnet example

In BACnet controllers the days of the week are numbered 1-7.

- Monday is day 1.
- Sunday is day 7.
- The day can also be identified by the first three letters (SUN, MON, etc.).

```
IF DOW = MON THEN START BO1
```

DOY

Returns the day of the year.

- The year always begins on January 1.
- December 31st is day 366.
- February is always counted as having 29 days which means March 1 is always day 61.
- On non-leap years, February 29 (day 60) is skipped.

The day of the year may be expressed as either a number or the first three letters of the month and the day of the month.

Standard BACnet Control Basic example

```
IF DOY = 92 THEN START BO1
```

Next Generation Control Basic example

```
IF DOY=92 THEN START BO1
```

ENABLE

ENABLE sets the value of an object, which can be the present value of an input, output or value object to 1 or ***on***.

Syntax: *ENABLE_point*

BACnet example

```
ENABLE AO1
ENABLE A
```

Related topics

- *DISABLE* on page 190
- *START* on page 225
- *STOP* on page 226

END

Terminates the execution of a program. When the END statement is encountered, the program stops reading lines and exits the program. All program lines that follow an encountered END statement are ***not*** executed.

In the following example, the last line is ignored and the analog output will always equal 10.

BACnet example

```
AO1 = 10
END
AO1 = 7
```

KMD example

```
OUT1 = 10
END
OUT1 = 7
```

ENDFUNC

Use ENDFUNC to mark the end of the definition of a Control Basic function.

Note: Procedures and functions apply only to controllers with Generation 5 Control Basic.

```
FUNCTION validateSp(s, l, h)
  s = MAX(s, l)
  s = MIN(s, h)
  FRETURN s
ENDFUNC
```

See also the topics *FUNCTION* on page 195 and *User defined procedures and functions* on page 176.

ENDPROC

Marks the end of the definition of a Control Basic user defined procedure.

Note: Procedures and functions apply only to controllers with Generation 5 Control Basic.

```
PROCEDURE lowOAT(a,b)
  AV1 = MIN(a,b)
ENDPROC
```

See also the topics *PROCEDURE* on page 215 and *User defined procedures and functions* on page 176.

ENTHALPY

Calculates enthalpy based on Outside Air Temperature (OAT) and Outside Air Humidity (OAH). The value returned is expressed as BTUs per pound of air. OAT is in degrees Fahrenheit.

Syntax: *ENTHALPY(_OAH_,_OAT_)*

For BACnet controllers, see the topic *ENTHALPYSI* on page 193 to enter OAT in degrees Celsius.

BACnet example

```
E = ENTHALPY( AI1 , AI2 )
```

ENTHALPYSI

Calculates enthalpy based on Outside Air Temperature (OAT) and Outside Air Humidity (OAH). The value returned is expressed as kilojoules per kilogram of air. OAT is in degrees Celsius.

Syntax: *ENTHALPYSI(_OAH_,_OAT_)*

See the topic *ENTHALPY* on page 193 to enter OAT in degrees Fahrenheit.

Standard Control Basic example

```
10 E = ENTHALPY-SI( AI1 , AI2 )
```

Next Generation Control Basic example

```
E=ENTHALPYSI(AI1, AI2)
```

ENTHALPY-SI

Calculates enthalpy based on Outside Air Temperature (OAT) and Outside Air Humidity (OAH). The value returned is expressed as kilojoules per kilogram of air. OAT is in degrees Celsius.

Syntax: *ENTHALPY-SI(_OAH_,_OAT_)*

Note: Deprecated for BACnet controllers. See *ENTHALPYSI* on page 193.

FLUSH

When a FLUSH statement runs, Control Basic immediately reads from or writes to the property bound to the local variable declared by ALIAS.

Syntax: *Flush (LocalAlias1)*

Note: Next generation Control Basic only.

```
ALIAS(1212, BO1, PV@4, Lights, 1:00:00, 60)
FLUSH(Lights)
```

See the related topic *ALIAS* on page 182.

FOR TO NEXT

The FOR TO NEXT loop repeats a set of instructions a specific number of times.

Syntax: *FOR_ConrtrolVariable_=_StartValue_to_EndValue(_Step_Increment_)*

- ***ControlVariable*** is the variable that ***FOR*** increments each time the loop repeats. It controls whether or not Control Basic repeats the loop. ***ControlVariable*** must be local to the controller in which the Control Basic program is running.
- ***StartValue*** is the initial value that Control Basic assigns to ***ControlVariable***.
- ***EndValue*** is the value that the ***ControlVariable*** must equal before the loop ends.
- ***Increment*** is the amount that Control Basic adds to ***ControlVariable*** with each iteration of the loop. ***Increment*** can be a positive or negative value. If ***STEP*** and ***Increment*** are omitted, the default value is 1.
- NEXT ends FOR TO statements. It directs Control Basic to increment ***ControlVariable*** and to test whether it is greater than ***EndValue***. If it is not, the loop continues at the first statement within the loop; if not, the program continues at the first statement following NEXT.

In the following examples, the value of ***A*** increases from 0 to the value of ***AV2*** in 0.1 increments, pausing 10 seconds between steps.

Standard Control Basic example

```
FOR A = 0 TO AV2 STEP .1
AO1 = A
WAIT 0:00:10
NEXT A
END
```

Next Generation Control Basic example

```
FOR A = 0 TO AV2 STEP .1
    AO1 = A
    WAIT 0:00:10
NEXT A
END
```

FRETURN

Use FRETURN to designate the value that is returned by the function. A function can include multiple FRETURN statements but only the first one executed designates the returned a value.

Note: Procedures and functions apply only to controllers with Generation 5 Control Basic.

```
FUNCTION validateSp(s, l, h)
  s = MAX(s, l)
  s = MIN(s, h)
  FRETURN s
ENDFUNC
```

See also the topics *FUNCTION* on page 195 and *User defined procedures and functions* on page 176.

FUNCTION

This keyword marks the beginning of a user defined function. A function is a self-contained block of statements that return a single value.

Note: Procedures and functions apply only to controllers with Generation 5 Control Basic.

Syntax: *FUNCTION name (argument[, argument, ...])*

Rules specific to functions:

- All procedures must have unique names. The names cannot duplicate Control Basic keywords, object names or descriptions, or mnemonics.
- A function must be called from an expression.
- Functions are declared with the keyword FUNCTION( ).
- Arguments passed to the function are enclosed in parentheses and separated by commas. The parenthesis ( ) are required even if no arguments are passed.
- The body of a function ***must*** end with the keyword ENDFUNC.
- The keyword FRETURN must be used to return from a function and must be followed by a value to return.
- All branches created by an IF statement must end with a FRETURN statement.

```
FUNCTION validateSp(s, l, h)
  s = MAX(s, l)
  s = MIN(s, h)
  FRETURN s
ENDFUNC
```

Related topics

User defined procedures and functions on page 176

FRETURN on page 195

ENDFUNC on page 192

PROCEDURE on page 215

GOSUB

GOSUB is the preferred way of branching to a subroutine in a program and then returning to the original point and continuing execution. When Control Basic encounters a GOSUB statement, the program jumps to the location specified and continues reading program lines until a RETURN statement is encountered. At that point the program returns to the line following the GOSUB statement.

Syntax: *GOSUB_line#*

In the following examples, the program reads the first line, jumps to the third line and then to the fourth line. The RETURN statement on the fourth line sends the program back to the second line and the program ends.

See the related topics *GOTO* on page 197 and *RETURN* on page 218.

BACnet example

```
GOSUB 30
END
REM
RETURN
```

Next generation Control Basic

```
      GOSUB DoSubRoutine
      END
DoSubRoutine:
      RETURN
```

GOTO

This statement redirects the program to a new location in the program.

In the following examples, the program does not run the second line and output 1 is never changed.

See the related topic *GOSUB* on page 196.

Syntax: *GOTO_line#*

BACnet example

```
GOTO 30
START BO1
REM Program continues here
END
```

Next generation Control Basic

```
      GOTO JumpToEnd
      START BO1
JumpToEnd:
      END
```

HALT

Stops the program from running and sets the ***Program State*** property to ***Halted***. The string ***Message*** is displayed in the property ***Description of Halt***.

Syntax: *HALT "Message"*

Note: For Next generation Control Basic only.

Once stopped the program cannot be restarted from Control Basic. It can be restarted only by doing one of the following:

- Performing a warm start or cold start
- Cycling controller power
- Changing the ***Program Change*** property on the program object to ***Run***.

```
HALT "Shutting down the program"
```

HSEL

Selects the highest (second highest, etc.) value of the expressions listed. The value for ***N*** defines whether it selects the highest (1) or the second highest (2) etc. The expressions can be variables, inputs, outputs, calculations, etc.

Syntax: *HSEL(_N_,_expression_,_expression...)*

This example returns the local variable ***A*** equal to the second highest value of the items listed.

BACnet Control Basic example

```
10 A = HSEL( 2 , AI1 , AI2 , AI3 , AV1 )
```

Next Generation Control Basic example

```
A=HSEL(2, AI1, AI2, AI3, AV1)
```

IF THEN (ELSE)

IF THEN is a decision making statement. The ***expression*** parameter can be any expression capable of being true or false (high or low, on or off, etc.) If ***expression*** is ***true*** the THEN statement will be executed. If the expression is ***false*** (not true) the ELSE statement will be executed. The ELSE statement and associated clause are optional. If they are not included the program reads and executes the next program line.

Syntax: *IF_expression_THEN_clause(_ELSE_clause)*

Standard BACnet Control Basic example

In this example, the program stops analog output #5 if Analog Input AI11 is less than Analog Input AI2. If AI1 is not less than AI2, Analog Output AO5 will be turned on (started). If the ***ELSE START AO5*** statement was not included, the program will stop AO5 if AI1 is less than

AI2. Otherwise, it will do nothing and end the program.

```
IF AI1 < AI2 THEN STOP AO5 ELSE START AO5
```

Note: Use commas to separate multiple commands in an IF statement.

```
IF T > S THEN START BO1 , STOP BO2
IF T > S THEN START BO1, STOP BO2 ELSE STOP BO1
```

Next generation Control Basic

By using ENDIF, Next Generation Basic supports block and nested IF THEN statements.

```
IF TIME > 7:00:00 THEN
    a=b
ENDIF


Locals ChilledWaterSetpoint
AV24 = ChilledWaterSetpoint
IF BV258 THEN
    ChilledWaterSetpoint=52
ELSE
    Chilledwatersetpoint=48
ENDIF


IF TIME > 7:00:00 THEN
    IF TIME < 9:00:00 THEN
        B=C
    ENDIF
ENDIF
```

IF+ THEN

IF+ is similar to IF THEN, except that it detects the first time a condition changes from ***false*** to ***true***. If the expression is true and on the previous scan it was not true, the THEN clause will be executed.

Syntax: *IF+_expression_THEN_clause(_ELSE_clause)*

The ELSE statement and associated clause are optional. If they are not included the program reads and executes the next program line.

When a button closes the circuit in the sensor analog input 1 to which it is connected, the program will branch down to line 30, which increases the setpoint (AV13 or VAR13) by one degree. This will happen only once for each time the button is pressed and released. Even if the button is held for several minutes it will only increment the setpoint by one degree.

See the related topic *IF THEN (ELSE)* on page 198 and *IF- THEN* on page 200.

BACnet example

```
IF+ SENSOR-ON( AI1 ) THEN GOSUB 30
END
AV13 = AV13 + 1 : REM Line 30 starts here
RETURN
```

Next generation Control Basic

```
IF+ SENSORON( AI1 ) THEN GOSUB 30
END
30: : AV13 = AV13 + 1
RETURN
```

IF- THEN

IF- is similar to IF THEN except that it detects the first time a condition changes from ***true*** to ***false***. In this case the THEN clause would only be executed if the expression is ***false*** and on the previous scan it was ***true***.

Syntax: *IF-_expression_THEN_clause(_ELSE_clause)*

Note: The *ELSE* and associated clause is optional.

See the related topic *IF THEN (ELSE)* on page 198 and *IF+ THEN* on page 199.

INC

Increments the value of the argument ***point*** by the value of the argument ***step***. If ***step*** is omitted, the step value is 1. ***Point*** may be the present value of any analog object.

Syntax: *INC(_point_, step_) INC(_point_)*

See the related topic *DEC* on page 189.

Standard BACnet Control Basic example

```
INC( AV1 , A + B )
INC( AV2 )
```

Next Generation Control Basic example

```
INC( AV1 , A + B )
Inc( AV2 )
```

INTEGER

Use the command INTEGER to declare a local variable or array of data type ***INTEGER***. For a description of data types, see the topic, *Data types* on page 179.

Note: Procedures and functions apply only to controllers with Generation 5 Control Basic.

Syntax: *INTEGER variable[, variable, ...]*

- A local variable may be used only within the program in which it is declared.
- Local variables declared with LOCALS, REAL, STRING, or INTEGER must be declared before they are used in a program. Typically local variables are declared at the beginning of the program.
- Must start with a letter A-Z, a-z, or an underscore (_). They are not case sensitive.
- Can be any combination of letters (A-Z or a-z), numbers (0-9) or the underscore (_).
- Variables A-Z are automatically declared unless LOCALS, REAL, STRING, or INTEGER declares another variable.
- A local variable cannot duplicate a keyword, constant, label or alias.

```
INTEGER fanSpeed
fanSpeed = 50
```

The command INTEGER can also be used to declare an array.

Syntax: *INTEGER arrayName[size]*

- When declaring an array, enclose the size of the array in brackets [ ].
- Each element in the array counts as one of the 256 local variables.
- The rules for naming and declaring an array are the same as the rules for declaring variables.

```
INTEGER rooms[25]
AV1 = rooms[3]
```

Related topics

- *LOCALS* on page 204
- *REAL* on page 217
- *STRING* on page 226
- *Programming with variables* on page 173.
- *Arrays and dynamic access* on page 178

INTERVAL

The INTERVAL command performs an operation at a regular time interval. The statement is ***true*** at each expression time; otherwise it is ***false***. The time format is in ***hh:mm:ss*** format.

Syntax: *INTERVAL(_expression_)*

The program sequence in this example increases the setpoint temperature—stored in value object AV1—by 0.1° every 45 seconds.

Standard BACnet Control Basic example

```
IF INTERVAL( 00:00:45 ) THEN AV1 = AV1 + .1
END
```

Next Generation Control Basic example

```
IF INTERVAL(00:00:45) THEN AV1 = AV1 + .1
END
```

INT

INT returns the integer portion of the numeric value ***expression***. The value returned is the greatest integer that is less than or equal to the value of ***expression***.

Syntax: *INT(_expression_)*

The following examples calculate the hour of the day (0-23) without minutes or seconds. The result is stored in analog value object AV1.

Standard BACnet Control Basic example

```
AV1 = INT( TIME / 100 )
```

Next Generation Control Basic example

```
AV1=INT(TIME/100)
```

INVLN

The function INVLN returns the inverse natural logarithm of the numeric expression.

Note: Next generation Control Basic only.

Syntax: *INVLN(_expression_)*

Standard BACnet Control Basic example

```
B = INVLN( AI4 * 125 )
```

Next Generation Control Basic example

```
B = INVLN( AI4 * 125 )
```

See the related topics ***LN*** on page 204.

ISNAN

ISNAN tests the value of ***expression*** to determine if it is a valid number. If the value of ***expression*** is equal to NAN (Not A Number), then ISNAN returns ***true***.

Syntax: *ISNAN(_expression_)*

Note: Next generation Control Basic only.

A typical use of ***ISNAN*** is to test the present value property of an object in a remote device.

Note: If the remote device goes offline, the last good value is held until the controller is reset with a cold start, warm start, or power cycle. After the reset, the value in the remote property becomes NAN until it is read by another controller.

In the following example the program tests the present value of analog input 4 in device instance 4410 once every minute. If the value is a usable number then the remote value is stored in value object AV503. If the remote value is not valid, the value object is set equal to 55, the default value.

```
IF INTERVAL ( 00:01:00 ) THEN
  REM Verify that the value is good
  IF ISNAN( 4410.AI4 ) THEN
    REM Set a default value
    AV503 = 55
  ELSE
    REM Use the received value
    AV503 = 4410.AI4
  ENDIF
ENDIF
```

LET

The LET function assigns ***expression1*** to equal ***expression2***. Use this function to assign values to inputs, outputs, variables, PID control loops or schedule.

Syntax: *LET_expression1_=_expression2*

```
LET OUT1 = CON1
LET A = OUT1
```

The LET function is optional. Both of the following examples will produce the same results.

```
VAR3 = IN2 - 23
LET VAR3 = IN2 - 23
```

LN-1

LN-1 returns the inverse natural logarithm of the numeric expression.

Syntax: *LN-1(_expression_)*

Note: Deprecated for BACnet controllers. See the keyword *INVLN* on page 202.

LN

The function ***LN( )*** returns the natural logarithm of the numeric expression.

Syntax: *LN(_expression_)*

Standard BACnet Control Basic example

```
B = INVLN( AI4 * 125 )
```

Next Generation Control Basic example

```
B = INVLN( AI4 * 125 )
```

LOCALS

Use to declare local variables or arrays of data type REAL.

Note: Next generation Control Basic only.

Syntax: *LOCALS variable[, variable, ...]*

```
LOCALS chilledWaterSetpoint, a, b
```

- A local variable may be used only within the program in which it is declared.
- Local variables declared with LOCALS, REAL, STRING, or INTEGER must be declared before they are used in a program. Typically local variables are declared at the beginning of the program.
- Must start with a letter A-Z, a-z, or an underscore (_). They are not case sensitive.
- Can be any combination of letters (A-Z or a-z), numbers (0-9) or the underscore (_).
- Variables A-Z are automatically declared unless LOCALS, REAL, STRING, or INTEGER declares another variable.
- A local variable cannot duplicate a keyword, constant, label or alias.

The command LOCALS can also be used to declare an array of data type REAL.

Syntax: *LOCALS arrayName[size]*

- When declaring an array, enclose the size of the array in brackets [].
- Each element in the array counts as one of the 256 local variables.
- The rules for naming and declaring an array are the same as the rules for declaring variables.

```
LOCALS roomTemps[10]
```

Related topics

- *INTEGER* on page 201
- *REAL* on page 217
- *STRING* on page 226
- *Programming with variables* on page 173
- *Arrays and dynamic access* on page 178

LSEL

LSEL returns the lowest, second lowest, etc. value of the expression listed. The value ***N*** defines whether it selects the lowest (1) or second lowest (2) etc. Expressions can be variables, inputs, outputs, calculations, etc.

Syntax: *LSEL(_N_,_expression_,_expression_…)*

In the examples local variable ***A*** will be set equal the second lowest value of the items listed.

Standard BACnet Control Basic example

```
A = LSEL( 2 , BI1 , BI2 , BI3 , BV1 )
```

Next Generation Control Basic example

```
A=LSEL(2, BI1, BI2, BI3, BV1)
```

MAX

MAX returns the maximum value of the expression listed. Expressions can be the present value of an input, output, or value object or the result of a calculation.

Syntax: *MAX(_expression_,_expression_…)*

Standard BACnet Control Basic example

```
A = MAX( AI1 , AI2 , AI3 , AV1 )
```

Next Generation Control Basic example

```
A=MAX(AI1, AI2, AI3, AV1)
```

MIN

MIN returns the minimum value of those expression listed. Expressions can be the present value of an input, output, or value object or the result of a calculation.

Syntax: *MIN(_expression_,_expression_…)*

Standard BACnet Control Basic example

```
B = MIN( AI1 , AI2 , AI3 )
```

Next Generation Control Basic example

```
B = MIN( AI1 , AI2 , AI3 )
```

MOD

MOD is an arithmetic operator returns the remainder of a division operation.

Syntax: *Dividend MOD Divisor*

Standard BACnet Control Basic example

```
IF AV1 MOD 5 = 0 THEN START BO2 ELSE STOP BO2
```

Next Generation Control Basic example

```
IF AV1 MOD 5=0 THEN START BO2 ELSE STOP BO2
```

The following example uses MOD to calculate leap year. If the year in the controller's internal clock is a leap year, local variable ***L*** is set to ***true***. For other years the variable ***L*** is set to ***false***.

```
IF YEAR MOD 4 = 0 AND YEAR MOD 100 <> 0 OR YEAR MOD 400
= 0 THEN L = 1 ELSE L = 0
```

See the related topic *Using arithmetic operators* on page 172.

MODELNUMBER

Returns the numerical portion of the model number of the controller.

Syntax: *MODELNUMBER*

```
AV1=MODELNUMBER
```

MODEL-NUMBER

Returns the numerical portion of the model number of the controller.

Note: Deprecated for BACnet controllers. See the keyword *MODELNUMBER* on page 206.

MONTH

Returns the current month of the year.

Standard BACnet Control Basic example

```
M = MONTH
```

Next Generation Control Basic example

```
M=MONTH
```

NAN

Use NAN to set a variable or property to a ***Not A Number*** constant or to test if the variable or property is equal to ***Not A Number***.

Note: Next generation Control Basic only.

```
IF A <> NAN THEN GOTO CONTINUE
B = 55
CONTINUE:
B = A
```

See the related topic *ISNAN* on page 203.

NEG_INFINITY

Use the NEG-INFINITY command to set a variable or property to a very large negative number.

Note: Procedures and functions apply only to controllers with Generation 5 Control Basic.

Note: The underscore(_) is required for this keyword.

```
a = NEG-INFINITY
AV21.LL = NEG-INFINITY
```

See also the topic *POS_INFINITY* on page 214.

NETPOINT

Use the NETPOINT function to set the interval for reading from and writing to BACnet off-panel objects.

Syntax: *NETPOINT(device.object.property, read interval, write interval)*

The lowest read or write interval of all NETPOINT statements within the device is used in all programs within the same device.

Note: NETPOINT and COV functions can be disabled with the command NETPOINTCACHE. See the topic *NETPOINTCACHE* on page 209 for details.

In the following example, the local variable ***A*** is updated every 180 seconds (3 minutes) and the value of local variable ***B*** writes to Analog Value object 1 in device 5 every 60 seconds.

```
NETPOINT(5.AV1.PV,180,60)
A = 5.AV1
5.AV1@8 = B
```

Table 25-6 NETPOINT parameters

Parameter	Description	Comments
device	The device instance number	
object	A valid Control Basic object mnemonic	See *Programming with mnemonics* on page 174.
property	The property for reading or writing.	
read interval	The interval, in seconds, at which Control Basic will read the property.	The default value is 60 seconds. Enter ***NONE*** for a write-only function.
write interval	The interval, in seconds, at which Control Basic will write to the property.	The default value is ***NONE*** which sets the function to read-only.

See also the topics *NETPOINTCACHE* on page 209 and *Transferring values between BACnet controllers* on page 173.

NETPOINTCACHE

Use NETPOINTCACHE to enable and disable the NETPOINT and COV functions.

Syntax: *NETPOINTCACHE(expression)*

- Enter one (1) for ***expression*** to enable NETPOINT and COV functions. Uses the read and write intervals specified by NETPOINT and COV functions.
- Enter zero (0) for ***expression*** to disable NETPOINT and COV functions. Use this condition to read a property at the end of a scan when the property must be updated as soon as possible.

```
NETPOINTCACHE(0)
NETPOINTCACHE(TRUE)
```

The following example reads the value of Binary Output object BO1 from device 12 only when the Binary Value object BV1 is True(1).

```
NETPOINTCACHE(1)
REM Normal Program operation
REM goes here
IF BV1 Then
  NETPOINTCACHE(0)
  BV1 = 12.BO1
  WAIT 0:05:00
  NETPOINTCACHE(1) REM Resume Caching
ENDIF
```

See also the topics and *NETPOINT* on page 208.

NETSENSORSTATUS

Returns the connection status of a NetSensor with which the program can take appropriate action. The function returns ***true*** if a functional NetSensor is connected to the controller and ***false*** if the controller does not detect a NetSensor.

Standard Control Basic example

```
F NOT NETSENSORSTATUS THEN STOP BV1
```

Next Generation Control Basic example

```
IF NOT NETSENSORSTATUS THEN STOP BV1
```

NETSENSOR-STATUS

Returns the connection status of a NetSensor so the program can take appropriate action. The function returns ***true*** if a functional NetSensor is connected to the controller and ***false*** if the controller does not detect a NetSensor.

Note: Deprecate for BACnet controllers. See the keyword *NETSENSORSTATUS* on page 209.

NOT

NOT is a Boolean operator that performs a logical negation operation on an expression. If the expression is 0, the result is 1. If the expression is non-zero, the result is 0.

Syntax: *result = NOT expression*

```
IF NOT BV1 THEN STOP BO2
```

See the related topic *Using Boolean logic* on page 172.

ON GOSUB

ON GOSUB is a control statement. The program branches to the location from the list passed by the statement. The value of ***expression*** determines the location in the list to which Control Basic will continue. ***Expression*** is rounded to an integer. For example, if ***expression*** = 3 the program will branch to the location in the list. If the value of ***expression*** is greater than the number of locations listed or if ***expression*** is less than 1, no branch will occur.

Syntax: *ON_expression_GOSUB_location1[_location2_location3_...]*

See the related topic *RETURN* on page 218.

Standard BACnet Control Basic example

In this example Value Object ***AV1*** is equal to ***3*** which will cause the program to branch to Line 80. If AV1 were equal to 2, the program would branch to Line 60, etc.

```
AV1 = 3
ON AV1 GOSUB 40 , 60 , 80
END
RETURN
RETURN
RETURN
```

Next generation Control Basic

In this example Value Object ***AV1*** is equal to ***3*** which will cause the program to branch to label ***80***. If ***AV1*** equals ***2***, the program will branch to label ***60***, etc.

```
AV1 = 3
ON AV1 GOSUB 40 , 60 , 80
END
40:
RETURN
60:
RETURN
80:
RETURN
```

ON GOTO

ON GOTO is a control statement. The program branches to the locations from the list passed by the statement. The value of ***expression*** determines the location in the list to which the program will branch. ***Expression*** is rounded to an integer. For example, if ***expression*** = 3 the program will branch to the third location in the list. If the value of ***expression*** is greater than the number of locations listed or if ***expression*** is less than 1, no branch will occur.

Syntax: *ON_expression_GOTO_location1[_location2_location3_…]*

Standard BACnet Control Basic example

In this example Value Object ***AV1*** is equal to ***3*** which will cause the program to branch to Line 60. If AV1 were equal to 2, the program would branch to Line 60, etc.

```
AV1 = 3
ON AV1 GOTO 40 , 50 , 60
END
REM Program continues here
REM Program continues here
REM Program continues here
```

Next generation Control Basic

In this example Value Object ***AV1*** is equal to ***3*** which will cause the program to branch to label ***60***. If ***AV1*** equals ***2***, the program will branch to label ***50***, etc.

```
AV1 = 3
ON AV1 GOTO Forty, Fifty, Sixty
END
Forty:
Fifty:
Sixty:
```

ONERROR

When an error is detected on the line previous to the line containing ONERROR, the program continues at the line specified by ***location***. In the following examples, the program attempts to read an off-panel object and if the object is not found it substitutes the value ***70***.

Syntax: *ONERROR location*

Standard BACnet Control Basic example

```
AV16 = 101-AV1
ONERROR 40
GOTO 60 : REM Jump around error recovery
REM Error recovery
AV16 = 70
REM Continue program
```

Next generation Control Basic

```
AV16 = 101.AV1
ONERROR 40
GOTO 60 : REM Jump around error recovery
40:
REM Error recovery
AV16 = 70
60:
REM Continue program
```

OPEN

Use OPEN to set the present value of an object or point to ***on*** or ***true***.

Syntax: *OPEN_point*

Standard BACnet Control Basic example

```
OPEN V
20 OPEN A
30 OPEN BO1
```

Next Generation Control Basic example

```
OPEN V
OPEN A
OPEN BO1
```

Related topics

- *CLOSE* on page 186
- *START* on page 225
- *STOP* on page 226

OR

OR is a Boolean operator that performs the logical ***OR*** of the two expressions. The result is ***true*** if either expression is ***true***. The result is ***false*** if both expressions are ***false***.

Syntax: *result = expression1 OR expression2*

In the following example, local variable ***C*** will equal 1 if either of the variables ***A*** and ***B*** are equal to 1.

```
A = 1 : B = 0 : C = A OR B
```

See the related topic *Using Boolean logic* on page 172.

OUTPUTOVERRIDE

Returns the switch position of an optional HPO-6700 series output board installed in the controller in which Control Basic is running.

Syntax: *OUTPUTOVERRIDE(output)*

The argument ***output*** is returned ***FALSE*** if the switch is in ***AUTO*** and ***TRUE*** if the switch is set to either the ***OFF*** or ***HAND*** position. ***Output*** can be expressed as either of the following:

- The instance number of the output.
- A local variable whose value represents the number of an output object.

Standard BACnet Control Basic example

```
BV20 = OUTPUTOVERRIDE( 2 )
```

Next Generation Control Basic example

```
BV20=OUTPUTOVERRIDE(2)
```

OUTPUT-OVERRIDE

Returns the switch position of an optional HPO-6700 series output board installed in the controller in which Control Basic is running.

Syntax: *OUTPUT-OVERRIDE(_expression_)*

Note: Deprecated for BACnet controllers. See the keyword *OUTPUTOVERRIDE* on page 213.

PANELADDRESS

Returns the device instance number of the controller on which the Control Basic program is running.

Standard Control Basic example

```
P = PANELADDRESS
```

Next Generation Control Basic example

```
P=PANELADDRESS
```

PANEL-ADDRESS

Returns the KMD network address of the controller on which Control Basic is running.

Note: Deprecated for BACnet controllers. For BACnet controllers see *PANELADDRESS* on page 214

PI

Inserts the value of pi. The following examples convert angle ***D*** from degrees to radians.

Standard Control Basic example

```
A = PI * ( D / 180 )
```

Next Generation Control Basic example

```
A=PI*(D/180)
```

POS_INFINITY

Use the POS-INFINITY command to set a variable or property to a very large positive number.

Note: Applies only to controllers with Next Generation of Control Basic.

Note: The underscore(_) is required for this keyword.

```
a = POS_INFINITY
AV21.HL = POS_INFINITY
```

See also the topic *NEG_INFINITY* on page 207.

POWERLOSS

Use POWERLOSS to detect loss of power to the controller. It will also detect any other condition that causes the controller to run its restart sequence. This function returns ***true*** on the first scan of all Control Basic programs after power is restored. After the first scan, it returns as ***false*** .

The following examples are useful for monitoring intermittent power failures at a controller. The Analog Value object AV32 increments by 1 each time power is restored.

Standard Control Basic example

```
IF POWERLOSS THEN AV32 = AV32 + 1
END
```

Next Generation Control Basic example

```
IF POWERLOSS THEN AV32 = AV32+1
END
```

POWER-LOSS

Use POWER-LOSS to detect loss of power to the controller or any condition that forced the controller to reset. This function returns ***true*** on the first scan of all Control Basic programs after power is restored. After the first scan, it returns as ***false***.

Note: Deprecated for BACnet controllers. See *POWERLOSS* on page 215.

The following example is useful for monitoring intermittent power failures at a controller. The KMD variable point ***VAR32*** increments by 1 each time power is restored. ***POWER-LOSS*** may also be used to detect any other condition that causes the controller to perform its restart sequence.

```
IF POWER-LOSS THEN VAR32 = VAR32 + 1
END
```

PRETURN

Use PRETURN to return from a procedure before reaching ENDPROC. Control Basic returns program control to the next statement that called the procedure. PRETURN is optional.

Note: Procedures and functions apply only to controllers with Generation 5 Control Basic.

See also, *PROCEDURE* on page 215 and *User defined procedures and functions* on page 176.

PROCEDURE

This keyword marks the beginning of a user defined procedure. A procedure is a self-contained block of Control Basic statements that perform a task.

Note: Procedures and functions apply only to controllers with Generation 5 Control Basic.

Syntax: *PROCEDURE name([argument, argument, argument…])*

Rules specific to procedures:

- All procedures must have unique names. The names cannot duplicate Control Basic keywords, object names or descriptions, or mnemonics.
- A procedure cannot be called from an expression.
- Procedures are declared with the keyword PROCEDURE( ). The parenthesis ( ) are required even if no arguments are passed.
- The body of a procedure ***must*** end with the keyword ENDPROC.
- The keyword PRETURN is optional and can be used to return from a procedure before the ENDPROC statement is reached.

```
PROCEDURE lowOAT(a,b)
  AV1 = MIN(a,b)
ENDPROC
```

Related topics

- *User defined procedures and functions* on page 176
- *PRETURN* on page 215
- *FUNCTION* on page 195

READ-CONBIAS

Returns the bias value of a PID control loop. The value range is 0–100.

Syntax: *READ-CONBIAS(_expression_)*

```
B = READ-CONBIAS( 1 )
```

READ-CONPROP

Returns the proportional band value of PID control loop. The value range is 0–4000.

Syntax: *READ-CONPROP(_expression_)*

```
P = READ-CONPROP( 1 )
```

READ-CONRATE

Returns rate (derivative) value of PID controller. The value range is 0–2.00.

Syntax: *READ-CONRATE(_expression_)*

```
R = READ-CONRATE( 1 )
```

READ-CONRESET

Returns the reset (integral) value of a PID control loop. The value range is 0–255.

Syntax: *READ-CONRESET(_expression_)*

```
S = READ-CONRESET( 1 )
```

REAL

Use the command REAL to declare a local variable or array of data type ***REAL***. For a description of data types, see the topic, *Data types* on page 179.

Note: Procedures and functions apply only to controllers with Generation 5 Control Basic.

Syntax: *REAL variable[, variable, ...]*

- A local variable may be used only within the program in which it is declared.
- Local variables declared with LOCALS, REAL, STRING, or INTEGER must be declared before they are used in a program. Typically local variables are declared at the beginning of the program.
- Must start with a letter A-Z, a-z, or an underscore (_). They are not case sensitive.
- Can be any combination of letters (A-Z or a-z), numbers (0-9) or the underscore (_).
- Variables A-Z are automatically declared unless LOCALS, REAL, STRING, or INTEGER declares another variable.
- A local variable cannot duplicate a keyword, constant, label or alias.

```
REAL hotWaterSetpoint, outsideAirTemp
```

The command REAL can also be used to declare an array.

Syntax: *REAL arrayName[size]*

- When declaring an array, enclose the size of the array in brackets [].
- Each element in the array counts as one of the 256 local variables.
- The rules for naming and declaring an array are the same as the rules for declaring variables.

```
REAL roomTemps[10]
AV1 = roomTemps[5]
```

Related topics

- *INTEGER* on page 201
- *LOCALS* on page 204
- *STRING* on page 226
- *Programming with variables* on page 173.
- *Arrays and dynamic access* on page 178

REM

Place a REM statement at the beginning of a program line to insert explanatory comments or remarks. REM is a method to document the use of a subroutine or to explain a formula used in a calculation.

Syntax: *REM_string*

Standard BACnet Control Basic example

```
REM ** Step temperature every minute by 1 degree **
IF INTERVAL( 0:01:00 ) THEN AV1 = AV1 + 1
REM **calculation for velocity (FPM)**
AV1 = 4004.4 * SQR( AI1 )
END
```

Next Generation Control Basic example

```
REM ** Step temperature every minute by 1 degree **
IF INTERVAL(0:01:00) THEN AV1=AV1+1
REM **calculation for velocity (FPM)**
AV1=4004.4*SQR(AI1)
END
```

RETURN

This command returns control from a subroutine that was called with a GOSUB or ON GOSUB statement.

See the related topics *GOSUB* on page 196 and *ON GOSUB* on page 210.

RLQ

Relinquishes the priority level of a BACnet output or value object.

Syntax: *RLQ_object@priority*

Standard Control Basic example

```
RLQ AO1@7
```

Next Generation Control Basic example

```
RLQ AO1@7
```

RND

RND is a numeric function which returns a random number between 0 and ***expression***-1. It is useful for applications such as security lighting.

Syntax: *RND(_expression_)*

Standard BACnet Control Basic example

```
IF TIME = 20:00:00 + RND( 10:00:00 ) THEN START AO1
```

Next Generation Control Basic example

```
IF TIME=20:00:00+RND(10:00:00) THEN START AO1
```

SCANS

SCANS returns the rate a controller is processing all Control Basic programs. The value returned is expressed in scans per second. As the complexity or length of a program increases it takes longer to process and the number of scans per second decreases.

A useful application for SCANS is to create a time-based counter similar to those used for time-proportioning relays. If you use the INTERVAL or WAIT statements you are limited to a time division no smaller than one second. By programming a counter based on SCANS, the smallest time increment can range between 1/5 of a second to 1/50 of a second depending on how busy the controller is.

If a time proportioning relay sequence is based on a 5 second cycle for example, having time increments in only 1 second divisions would likely not be sufficient.

See the related topic *About Control Basic scans* on page 165.

Standard BACnet Control Basic example

```
A = 1 / SCANS
B = A + B : REM B Will increment by 1 every second
(based on scan rate)
IF B > 10 THEN B = 0 : REM B counts 0-10 in 10 seconds
END
```

Next Generation Control Basic example

```
A=1/SCANS
B=A+B:REM B Will increment by 1 every second (based on
scan rate)
IF B>10 THEN B=0: REM B counts 0-10 in 10 seconds
END
```

SCHEDOFF

Use this function to determine the time of day that a schedule object will set the present value of the reference object to ***Inactive*** or a value of zero (0).

Syntax: *SCHEDOFF(_schedule object #_,_time_)*

- The returned value is the difference—in seconds—from the time specified to the time that the schedule's present value will be ***Inactive*** (or zero).
- The schedule object must be within the same device.
- The parameter ***time*** is entered in 24-hour format.
- The value returned is based on the current day of the week.
- A returned value of 0 indicates that the schedule is already set to off.
- A return of 86,400 indicates that there are no more scheduled Off times for the current day.

Note: KMC Controls recommends that, because it is computationally intensive, Control Basic does not continuously run the SCHEDOFF function.

See ***SCHEDON*** on page 221 for calculating the time when a schedule becomes active.

SCHED-OFF

Deprecated for BACnet controllers. See ***SCHEDOFF*** on page 220.

SCHEDON

Use this function to determine the time of day that a schedule object will set its reference object to ***Active*** or a non-zero value.

Syntax: *SCHEDON(_schedule object #_,_time_)*

- The returned value is the difference—in seconds—from the time specified to the time that the schedule's present value will become ***Active*** (or non-zero).
- The schedule object must be within the same device.
- The schedule object at ***a*** must be within the same device. It can be from a Get block or the number of the schedule object entered in ***a***.
- The parameter ***time*** is entered in 24-hour format.
- The value returned is based on the current day of the week.
- A returned value of 0 indicates that the schedule is already on.
- A return of 86,400 indicates that there are no more scheduled On times for the current day.

Note: KMC Controls recommends that, because it is computationally intensive, Control Basic does not continuously run the SCHEDON function.

In the following example, line 10 calculates a value once every five minutes. The time parameter of 0:00:00 indicates when time the schedule will be on. If the schedule is set to change to On at 5:00 AM, when line 10 executes, value object AV1 will return a value of 18,000 seconds.

See ***SCHEDOFF*** on page 220 for calculating the time when a schedule becomes inactive.

Standard Control Basic example

```
10 IF INTERVAL( 0:05:00 ) THEN AV1 = SCHEDON( 1 ,
0:00:00 )
```

Next Generation Control Basic example

```
IF INTERVAL(0:05:0) THEN AV1=SCHEDON(1,0:00:00)
```

SCHED-ON

Deprecated for BACnet controllers. See the keyword ***SCHEDON*** on page 221.

SENSOROFF

Use SENSOROFF to detect an open-circuit condition on an input that is configured as an analog input. A typical application is to detect momentary conditions such as a pressed button. If the opened contact condition lasts longer than two minutes the function will be disabled. After three minutes, the object will change ***Out Of Service*** to ***true*** but the commands will still execute.

Syntax: *SENSOROFF(_IN#_)*

When used with *SENSORON* on page 223 and *IF THEN (ELSE)* on page 198, *IF+ THEN* on page 199, or *IF- THEN* on page 200 you can determine three separate conditions from one input:

- A temperature or other analog reading.
- A sensor with open contacts (SENSOROFF).
- A sensor with closed contacts (SENSORON).

SENSOROFF can also be used with inputs using a table if the minimum value in the table is set to a value greater than zero and its maximum value is less than 5.00 volts.

Table 25-7 Example table for SENSOROFF in BACnet controllers

Input Voltage	Detected condition
0	Closed circuit
0.4	Temperature-55 degrees
4.9	Temperature-95 degrees
5.0 or greater	Open circuit

In the table *Example table for SENSOROFF in BACnet controllers* on page 222, the input voltage under normal temperature conditions will never fall below 0.4 volts. When a sensor is shorted to ground, the input voltage will fall to zero, which is a condition SENSORON can detect. Similarly, if the circuit is opened, the controller will read the open circuit voltage, which is higher than the maximum 4.9 volts in the table which will be detected by SENSOROFF.

Standard Control Basic example

```
IF- SENSOROFF( AI1 ) THEN AV11 = 02:00:00
```

Next Generation Control Basic example

```
IF- SENSOROFF(AI) THEN AV11=02:00:0
```

SENSOR-OFF

Use SENSOR-OFF to detect an open-circuit condition on an input that is configured as an analog input.

Syntax: *SENSOR-OFF(_IN#_)*

Note: Deprecated in BACnet controllers. See the keyword *SENSOROFF* on page 222.

SENSORON

Use SENSORON to detect 0 volts (closed-circuit) condition on an input that is configured as an analog input. A typical application is to detect momentary conditions such as a pressed button. If the opened contact condition lasts longer than two minutes the function will be disabled. After three minutes, the object will change ***Out Of Service*** to ***true*** but the commands will still execute.

Syntax: *SENSORON(_IN#_)*

When used with *SENSOROFF* on page 222 and *IF THEN (ELSE)* on page 198, *IF+ THEN* on page 199, or *IF- THEN* on page 200 you can determine three separate conditions from one input:

- A temperature or other analog reading.
- A sensor with open contacts (SENSOROFF).
- A sensor with closed contacts (SENSORON).

SENSORON can also be used with analog inputs using a table if the minimum value in the table is set to a value greater than zero and its maximum value is less than 5.00 volts.

Table 25-8 Input conditions for SENSORON for BACnet controllers

Input Voltage	Detected condition
0	Closed circuit
0.4	Temperature-55 degrees
4.9	Temperature-95 degrees
5.0 or greater	Open circuit

In the table *Input conditions for SENSORON for BACnet controllers* on page 223, the input voltage under normal temperature conditions would never fall below 0.4 volts. When a sensor is shorted to ground, the input voltage will fall to zero, which is a condition SENSORON can detect. Similarly, if the circuit is opened, the controller will read 5.00 volts, which is higher than the maximum 4.9 volts in the table which will be detected by SENSOROFF.

Standard Control Basic example

```
IF+ SENSORON( AI1 ) THEN AV11 = 02:00:00
```

Next Generation Control Basic example

```
IF+ SENSORON(AI1) THEN AV1=02:00:00
```

SENSOR-ON

Use SENSOR-ON to detect 0 volts (closed-circuit) condition on an input that is configured as an analog input.

Syntax: *SENSOR-ON(_IN#_)*

Note: Deprecated for BACnet controllers. See *SENSORON* on page 223.

SIN-1

Returns the arcsine of the specified angle. The value ***angle*** is expressed in radians.

Syntax: *SIN-1(_angle_)*

Note: Deprecated for BACnet controllers. See the keyword *ARCSIN* on page 184.

SIN

Returns the sine of the specified angle. ***Angle*** is expressed in radians.

Syntax: *SIN(_angle_)*

Standard Control Basic example

```
A = SIN( AI1 )
```

Next Generation Control Basic example

```
A=SIN(AI1)
```

SQR

The SQR function returns a value equal to the square-root of the value ***expression***.

Syntax: *SQR(_expression_)*

Standard BACnet Control Basic example

```
A = SQR( AI1 )
```

Next Generation Control Basic example

```
A=SQR(AI1)
```

START

START sets the value of a point to ***on***.

Syntax: *START_point*

Standard BACnet Control Basic example

```
START AO1
START F
START A
```

Next Generation Control Basic example

```
START AO1
START F
START A
```

Related topics

- *STOP* on page 226
- *DISABLE* on page 190
- *ENABLE* on page 191

STOP

STOP sets the value of a ***point*** to ***Off***.

Syntax: *STOP_point*

Standard BACnet Control Basic example

```
STOP AO1
STOP F
```

Next Generation Control Basic example

```
STOP AO1
STOP F
```

Related topics

- *START* on page 225
- *DISABLE* on page 190
- *ENABLE* on page 191

STRING

Use the command STRING to declare a local variable or array of data type ***STRING***. For a description of data types, see the topic, ***Data types*** on page 179.

Note: Procedures and functions apply only to controllers with Generation 5 Control Basic.

Syntax: *STRING variable[, variable, ...]*

When using strings:

- A string is limited to 64 characters and consists of the ASCII letters and symbols on a standard U.S. keyboard.
- Strings can be assigned to controller properties that are of type string or to other variables of type string.
- Strings can be concatenate (joined) with the plus(+) sign.
- Strings can be compared to each other with arithmetic comparison.

Rules for declaring local variables.

- A local variable may be used only within the program in which it is declared.
- Local variables declared with LOCALS, REAL, STRING, or INTEGER must be declared before they are used in a program. Typically local variables are declared at the beginning of the program.
- Must start with a letter A-Z, a-z, or an underscore (_). They are not case sensitive.
- Can be any combination of letters (A-Z or a-z), numbers (0-9) or the underscore (_).
- Variables A-Z are automatically declared unless LOCALS, REAL, STRING, or INTEGER declares another variable.
- A local variable cannot duplicate a keyword, constant, label or alias.

```
STRING alarmMessageHi, alarmMessageLo
```

The command STRING can also be used to declare an array.

Syntax: *STRING arrayName[size]*

- When declaring an array, enclose the size of the array in brackets [].
- Each element in the array counts as one of the 256 local variables.
- The rules for naming and declaring an array are the same as the rules for declaring variables.

```
STRING messages[5]
```

Related topics

- *INTEGER* on page 201
- *LOCALS* on page 204
- *REAL* on page 217
- *Programming with variables* on page 173
- *Arrays and dynamic access* on page 178

TAN-1

A function that Rreturns the arctangent of the specified angle.The value of ***angle*** is expressed in radians.

Syntax: *TAN-1(_angle_)*

Note: Deprecated for BACnet controllers. See the keyword *ARCTAN* on page 184.

TAN

A function that returns the tangent of the specified angle. The value ***angle*** is expressed in radians.

Syntax: *TAN(_angle_)*

Standard Control Basic example

```
A = TAN( AV10 )
```

Next Generation Control Basic example

```
A=TAN(AV10)
```

TBL

Use the TBL function to calculate value of an expression in a custom created table. Use the function when the value of the expression is nonlinear or requires a complicated calculation to arrive at the proper value.

Syntax: *TBL(_expression_,_table_)*

- ***Expression*** is an analog input, value object, or variable.
- The value for ***table*** must be a whole number.
- In BACnet controllers use a Cbasic table not an input table. KMD controllers have only one type of table.

Standard Control Basic example

```
AV1 = TBL( AI3 , 2 )
```

Next generation Control Basic

```
AV1 = TBL( AI3 , 2 )
```

Example for TBL

The following example uses a table to calculate the value for a boiler hot water reset function as shown in the following graph.

1. Enter values into a table.
   - For BACnet controllers use Cbasic table 1.
   - For KMD controllers use Table 1
2. Set up an analog object or variable for the outside air temperature and with the name "OAT".
3. Set up a second analog object or variable with the name "boilerReset".

4 Add the following line to a Control Basic program.

```
boilerReset = TBL(OAT,1)
```

Illustration 25–1 Graph for boiler reset function

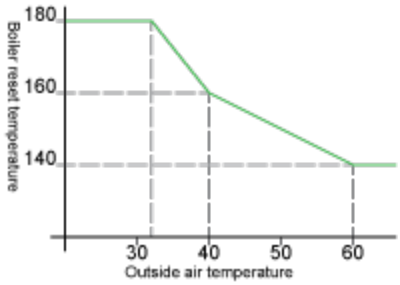


Table 25-9 Cbasic table values for boiler reset

Outside air temperature X Value	Reset temperature Y Value
32	180
40	160
60	140

As Control Basic runs the program, the value of the object boilerReset automatically updates as the outside air temperature changes.

- For any OAT value less than 32, the reset temperature is 180.
- For any OAT value greater than 60, the reset temperature is 140.
- For an OAT value between 32 and 60, the reset temperature is calculated as shown in *Graph for boiler reset function* on page 229.

TIME

A function that returns a value based on the time of day as maintained in the controller running Control Basic.

Tip: For the most accurate results, read the time only once in a program . If time is required in more than one place, read time and then assign the value to a local variable.

BACnet time format KMC BACnet controllers return a value for system time as the number of seconds after midnight.

KMD time format The TIME function in KMD controllers returns a number in the 24-hour format.

BACnet example

The following program returns hours, minutes and seconds in local variables H, M, and S.

```
t = TIME
h = t \ 3600 : REM Hours
m = INT( ( t - h * 3600 ) / 60 ) : REM Minutes
s = TIME - 60 * m - 3600 * H : REM Seconds
```

TIMEOFF

Use this statements to determine if the present value of object has been in the ***off*** state for a specific period of time. If ***point*** is a value object, it must be configured as a unit of time.

Syntax: *TIMEOFF(_point_)*

Standard Control Basic example

```
IF TIMEOFF(BO1)>0:10 THEN STOP BO2
REM BO2 will turn off if BO1 has been off for longer
than 10 minutes
```

Next Generation Control Basic example

```
IF TIMEOFF(BO1)>0:10 THEN STOP BO2
REM BO2 will turn off if BO1 has been off for longer
than 10 minutes
```

See the related keyword topic *TIMEON* on page 231.

TIME-OFF

Use TIME-OFF to determine if a point has been in an ***off*** state for a specific period of time.

Syntax: *TIME-OFF(_point_)*

Note: Deprecated for BACnet controllers. See the topic *TIMEOFF* on page 230.

TIMEON

Use this statements to determine if the present value of object has been in the ***on*** state for a specific period of time.

Syntax: *TIMEON(_point_)*

Caution

TIMEON responds to the time a BACnet property is set to *On* as maintained by the controller running the program. This time may not be the same as the actual time if the object containing the property is in a different controller.

In the following example, analog output BO2 will be set to ***On*** if output BO1 has been set to ***On*** for more than 10 minutes.

```
IF TIMEON( BO1 ) > 0:10 THEN START BO2
```

See the related keyword topic *TIMEOFF* on page 230.

TIME-ON

Use this statements to determine if the present value in a point t has been ***on*** for a specific period of time.

Syntax: *TIME-ON(_point_)*

Note: Deprecated for BACnet controllers. See the keyword *TIMEON* on page 231.

TOSTRING

Use TOSTRING to convert a numerical value to a string.

Syntax: *TOSTRING(argument1, argument2)*

See also the topic *STRING* on page 226.

WAIT

Use WAIT to control timed events. The program waits for the time period specified before reading the next program line. Other programs in the controller will not be affected as WAIT applies only to the program in which it is listed.

Syntax: *WAIT_period*

Tip: The value for ***period*** can be expressed in 24-hour format (14:15) or converted to decimal format (1425). See the related topic *TIME* on page 229.

Note: Plan carefully when using WAIT before a conditional branch such as with IF-THEN. Conditions within a controller may change the value of points or properties during the waiting period.

Standard BACnet Control Basic example

```
START AO2
REM Program continues here
WAIT 00:00:10 : REM ** Waits 10 seconds **
END
```

Next Generation Control Basic example

```
START AO2
REM Program continues here
WAIT 00:00:10 : REM ** Waits 10 seconds **
END
```

XOR

XOR performs a logical exclusion on two Boolean expressions. The result is ***true*** if the two expressions are different; otherwise, the result is ***false***.

Syntax: *result_=_expression1_XOR_expression2*

In the following example, local variable ***C*** will equal 1 as long as variables ***A*** and ***B*** are not equal to each other.

```
A = 1 : B = 0 : C = A XOR B
```

See the related topic *Using Boolean logic* on page 172.

YEAR

Returns the four-place value of the current year.

```
Y = YEAR
```

Section 26: Writing block programs

This section is an overview of block programming in KMC BACnet controllers.

The topics in this section describe the parts of the Control Basic Block editor and how to use it to build programs for KMC BACnet Controllers.

Topics in this section

Applicable controllers on page 233

Starting the Block Editor on page 234

A tour of the Block Editor on page 236

Get and Set blocks on page 238

Adding and editing routines on page 240

Defining local variables on page 239

Adding and editing routines on page 240

Macro blocks on page 241

Limitations and importing on page 245

Applicable controllers

Block program is available only in the Conquest series of controllers.

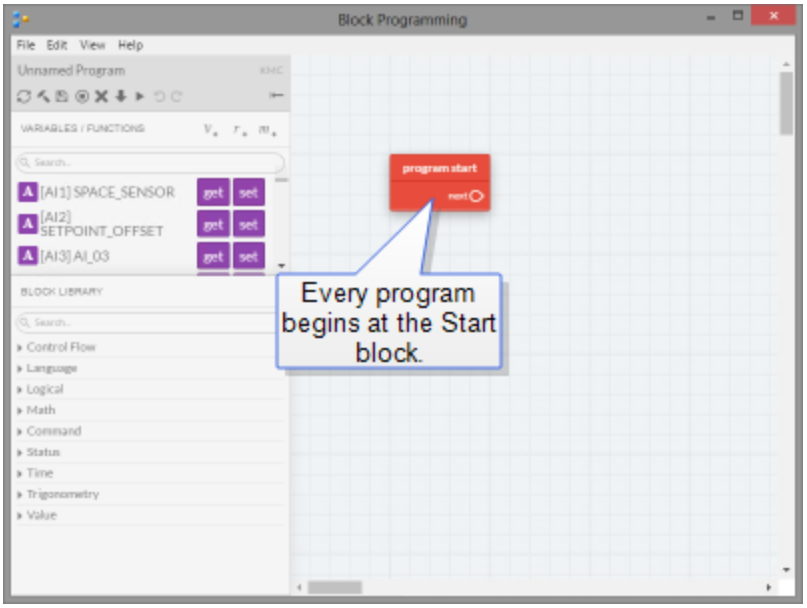
- BAC-5901
- BAC-9001
- BAC-9021
- BAC-9301
- BAC-9311

Starting the Block Editor

To start block programing do one of the following:

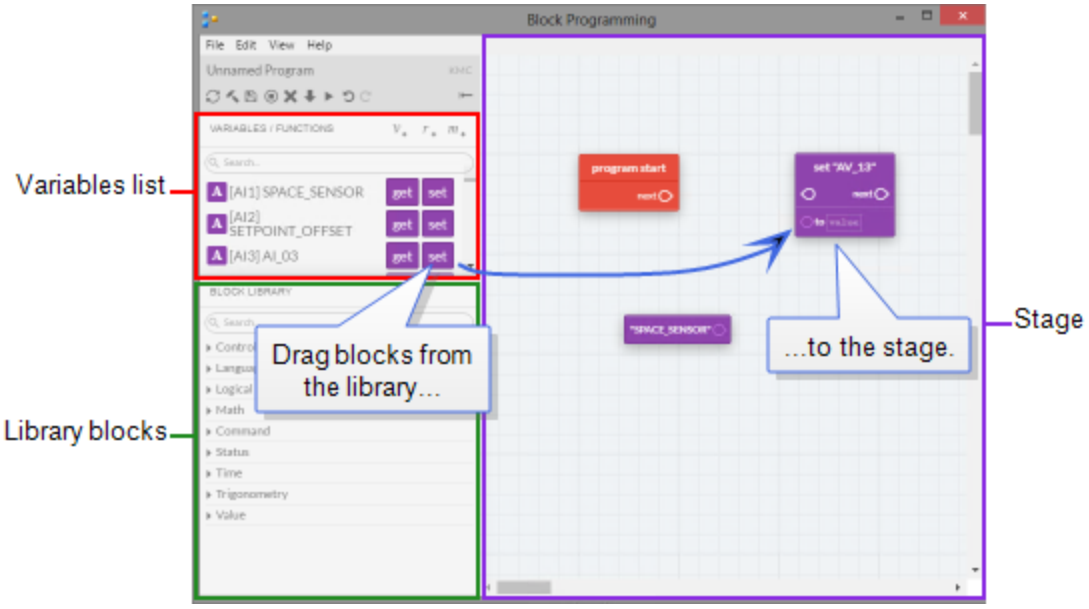
- In the Network Manager list, right-click a program object icon and then choose **Block Editor** from the shortcut menu.
- Open a program object and then choose **Block Editor** from the Control Basic Program group.

The block editor always includes a red program start block. The start block is added automatically and cannot be deleted.

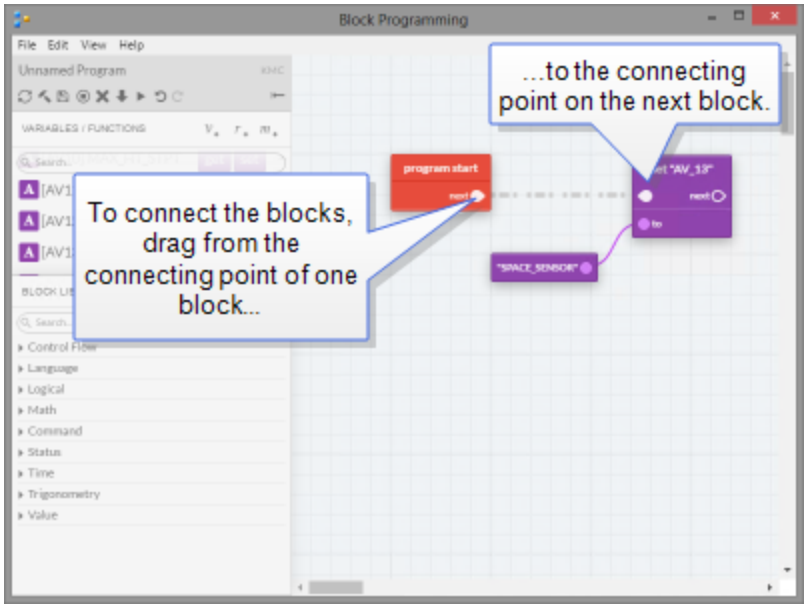


Add program blocks to the stage by dragging them from the variables and objects list or from one of the block library groups.

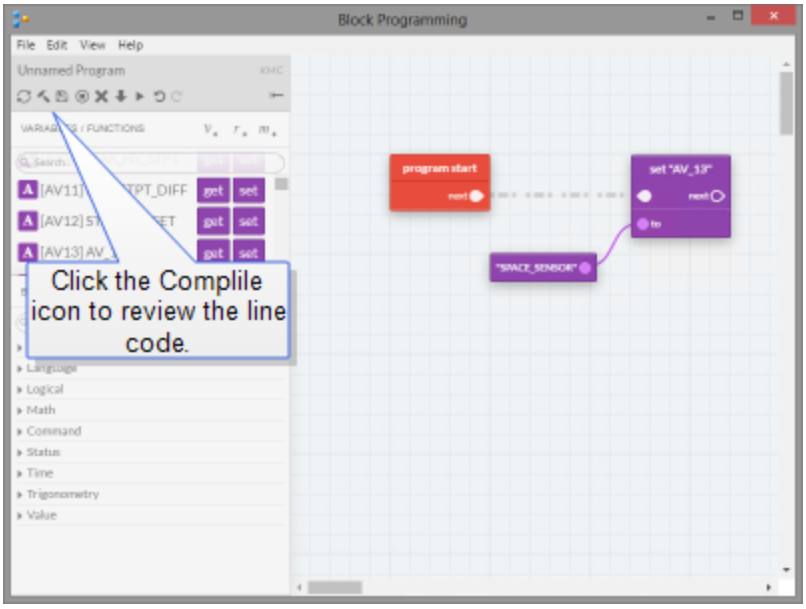
- Blocks in the library are programming functions and commands.
- The Get and Set blocks in the variables list transfer values to a program step.



To build a program, connect the blocks together to create the program logic. In the following illustration, the Get and Set block are connected to create the operation `AV13="SPACE_SENSOR"`.



Once all blocks are in place and connected, click the compile icon on the tool bar.



A tour of the Block Editor

This topic covers the major parts of the Control Basic Block Editor.

The parts of the editor

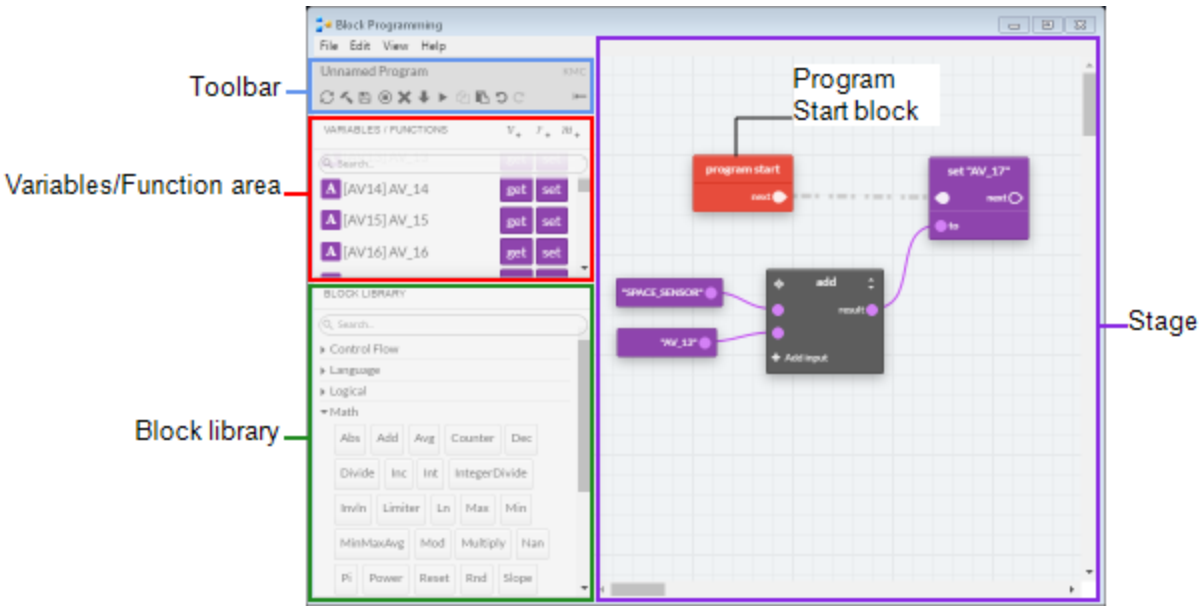
Toolbar Click the icons on the toolbar to compile, save, and run block programs.

Variables/Function The Variables/Function list has two parts. The three buttons in the top part of the list for adding local variables, routines, and macros. The blocks in the variables list represent objects in the controllers and local variables. If defined, subroutine and macro blocks may also be in the Variables/Function area.

Stage The stage is the work area of the Block Editor. Blocks are added to the stage to build the program.

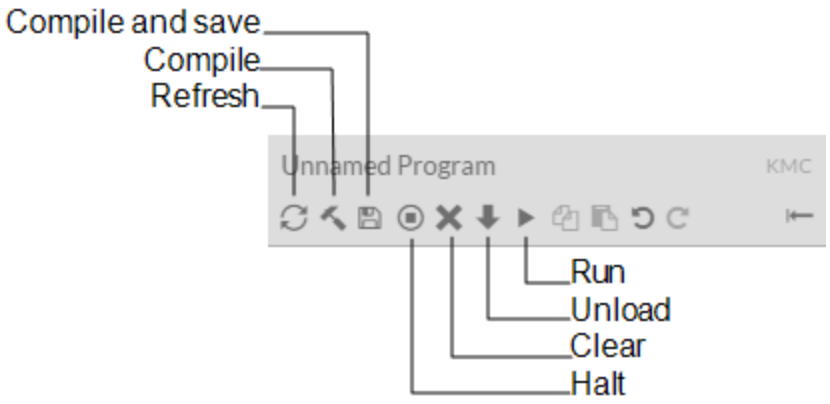
Block Library The blocks in the library are the commands and functions that make up the program. Command and function blocks are described in the section *Reference to Control Basic blocks* on page 247.

Program start block The Program Start block is automatically added when a new block program is started. The block is the beginning of the program and cannot be removed. Other blocks are added as needed to build the program.



The Block Editor toolbar

The icons on the toolbar compile, save, clear, and run the block programs in the program object.



Compile and Save Converts the block program into Control Basic line code and saves it in the program object.

Compile Converts the block program into Control Basic line code but does not save it in the program object.

Refresh Refreshes the Block Editor workspace.

Halt Stops the program from running in the program object.

Clear Deletes the program from the program object.

Unload Deletes the program from the program object. In KMC controllers this is similar to Clear.

Run Compiles the program, saves it to memory, and runs the program in the program object.

Get and Set blocks

Get and Set blocks represent objects and local variables within the controller. They transfer values between the program and local variables or objects in the controller.

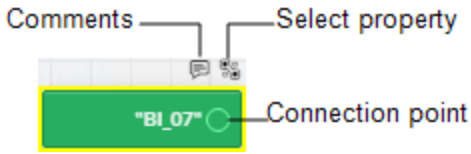
Get blocks

A Get block transfers a value to a program statement from local variables or objects in the controller.

Comments Comments to save notes or remarks for block in the comments area.

Select property Selects the Present Value or Out Of Service property value.

Connection point The connection point connects to a Set block ***to*** connector or to an input on other types of blocks.



Set blocks

A Set block transfers a value to local variables or objects from a program statement.

Comments Save notes or remarks for the block in the comments area.

Select property Selects the Present Value or Out Of Service property value.

Select priority Selects the BACnet priority for writing to the priority array. The default level is priority nine.

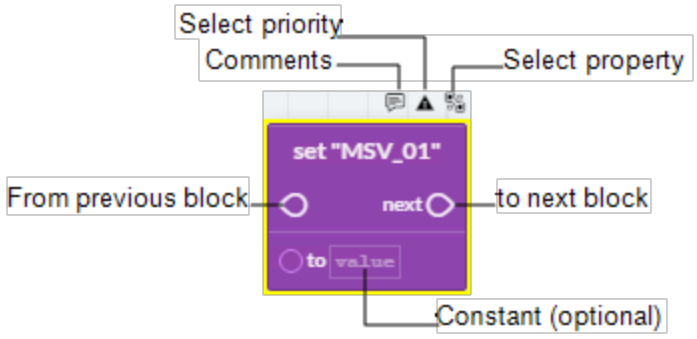
Connection point The connection point connects to Set blocks or an input on other types of blocks.

From previous block Connect to the Program Start block or the previous block in the program flow.

To next block Connects to the next block in the program flow.

value Enter a constant in the value field to set the object or variable to a fixed number.

to Connect to a Get block or the ***result*** connect from another block such as a math block. The ***to*** connector is disabled when there is an entry in ***value***.

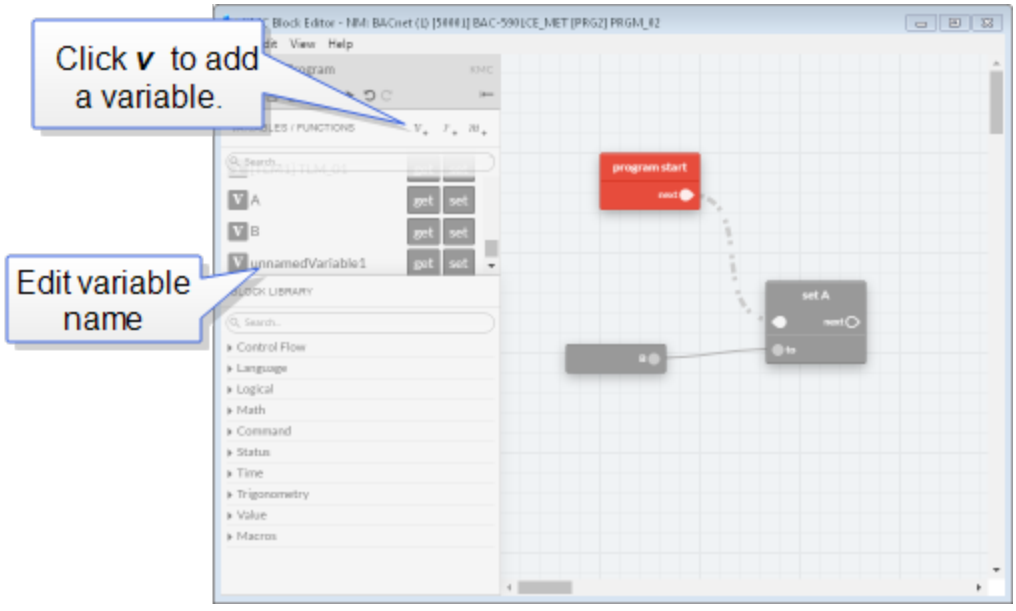


Defining local variables

Local variables are place holders for information. They can only be used within the Program object that defines them and cannot be directly transferred to other Control Basic programs. Local variables are often used for counters or to store the results of local calculations.

1. Click the variable button  in the Variables/Function area. Variable Set and Get blocks are added to the ***bottom*** of the variables list.
2. Give the variable a unique, descriptive name.
3. Use the variable Get and Set blocks the same as you would use other Get and Set blocks.

Tip: When the Block Editor is closed and then reopened, the local variables move to the top of the list.

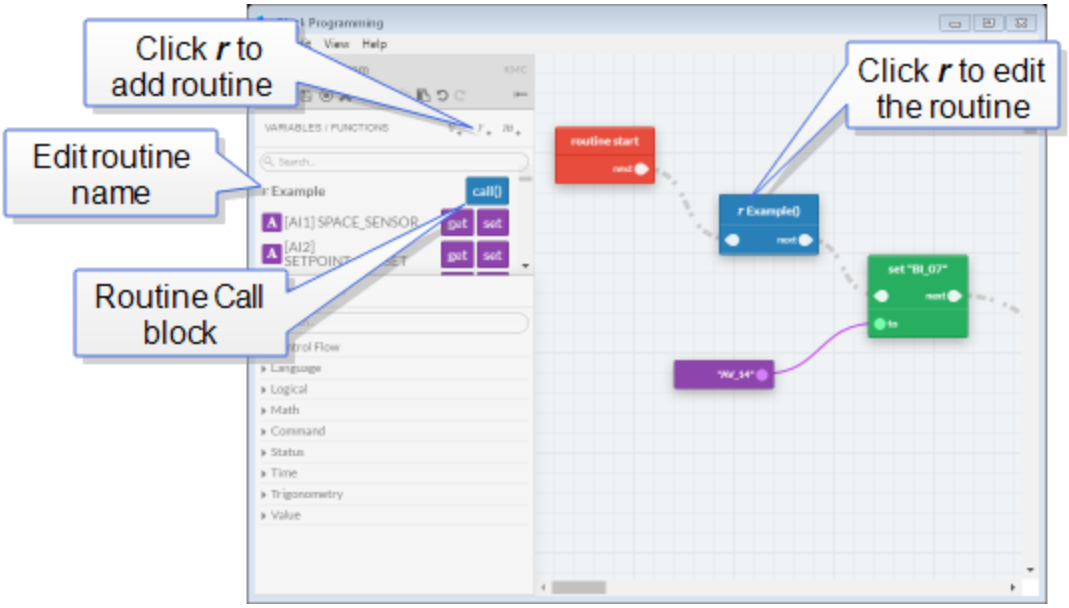


Adding and editing routines

A routine—often referred to as a subroutine— is a packaged unit of program blocks that perform a specific task. The unit can then be added to a program wherever that particular task should be performed. Routines are added as Call blocks in the Block Editor.

To add a routine, do the following:

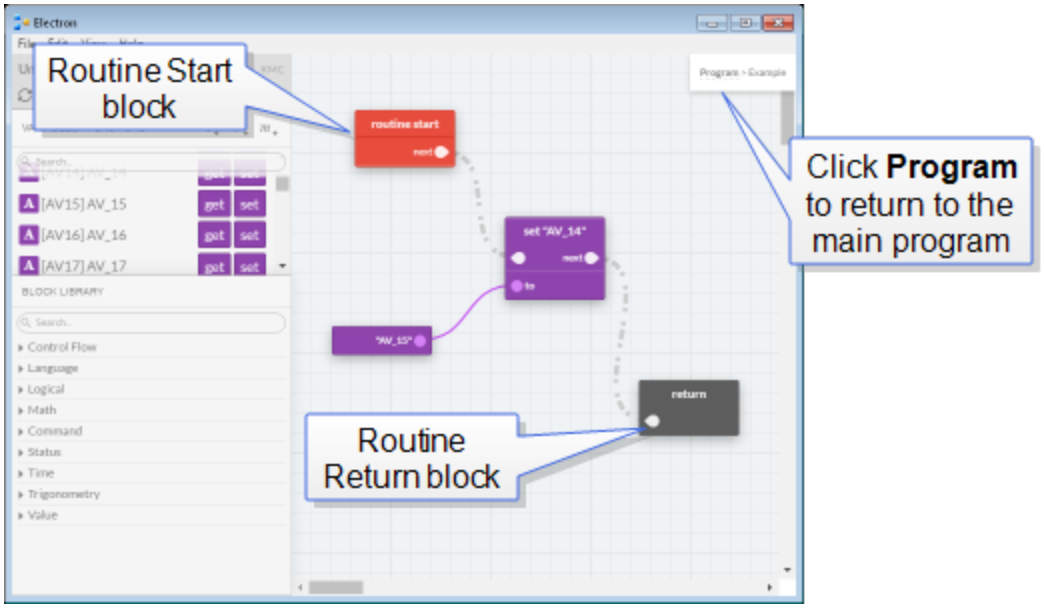
1. Click the routine button  in the Variables/Function area. A Call block is added to the variables list.
2. Give this block a unique, descriptive name.
3. Drag the Call block to the stage.
4. Connect the Call block to other blocks as required by program logic.



To edit a routine, do the following:

1. On a Call block, click the **r** next to the program name.
2. On the return stage, add blocks as needed between the Routine Start block and the Return block. The Routine Start and Return blocks are added automatically and cannot be removed.

3 When finished, click **Program** in the upper right corner to return to the main program stage.



Macro blocks

A Macro block represents a set of Control Basic instructions that are inserted into a program with a single block. That single block can be reused in the program without rewriting complicated or long sequences.

Macro block ports are the inputs and outputs to the set of instructions. The ports represents the arguments or parameters used within the macro.

- Input ports transfer values into the macro.
- Output ports transfer results from the macro to variables.
- The ports appear as the connector points on the Macro block.
- Ports are optional and can be added, deleted, or renamed as needed when setting up the macro.
- When the macro block is compiled, the reference to the port is replaced by the actual parameter, such as the results from a previous block, a variable, or constant.

Macro blocks can be saved on the local computer and reused in other programs, either in the same controller and other controllers in the Network Manager list.

Adding a macro block

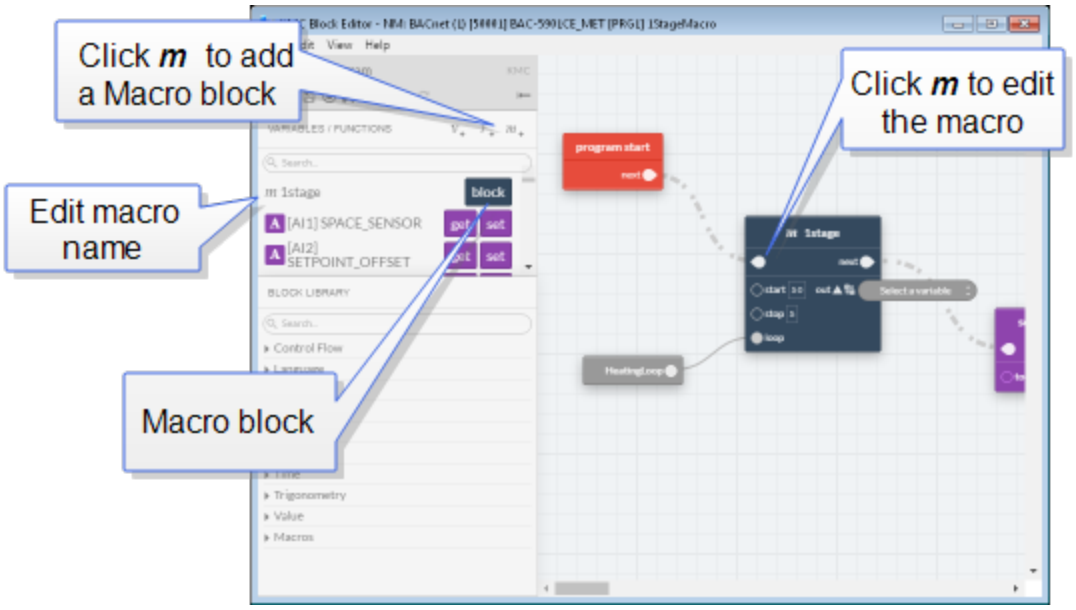
To add a macro, do the following:

1 Click the macro button  in the Variables/Function area. A Macro block is added above the variables list.

2 Give this block a unique, descriptive name.

3 Drag the Macro block to the stage.

4 Connect the Macro block to other blocks as required by program logic.

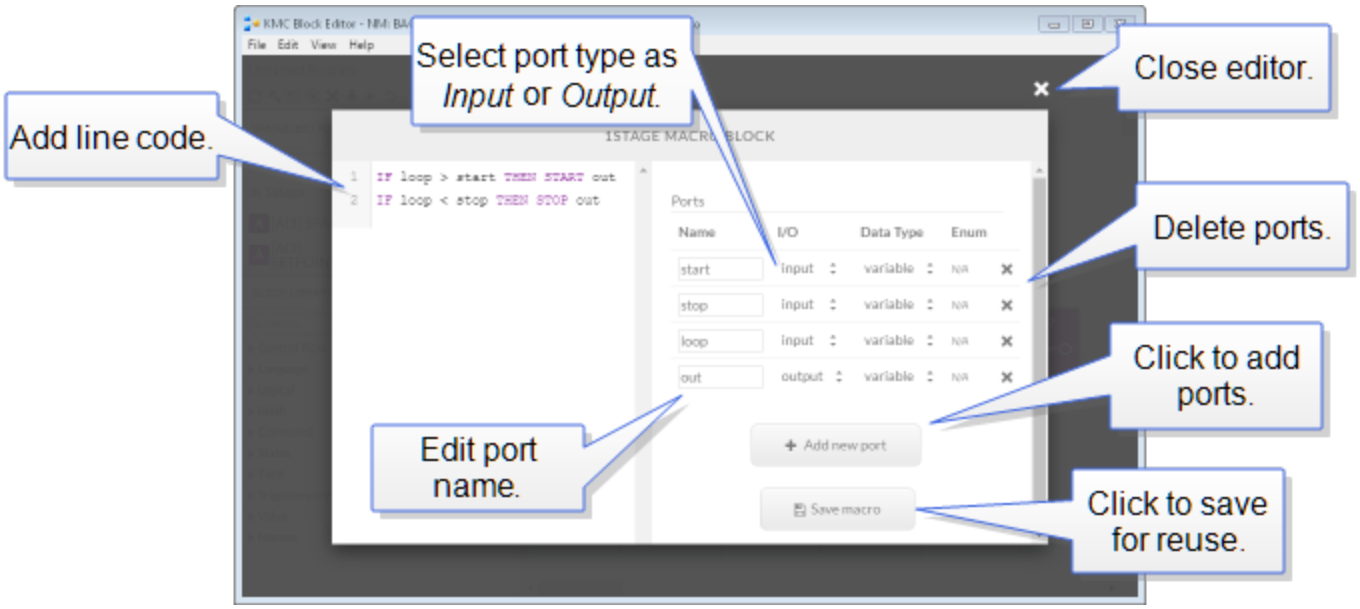


Editing and saving macros

To edit a macro, do the following:

1. On a Macro block, click the **m** next to the macro name.
2. In the Macro Block Editor, add or delete ports as needed.
   - Input ports add input connectors to the macro block and transfer parameters into the line code.
   - Output ports add a variable selector to the macro block and transfer results from the line code to a variable or object.
   - Change the port name to be descriptive of the port function.
3. Add the Control Basic line code using the names of the ports instead of mnemonics or object names. Control Basic can be entered directly in the editor or copied and pasted from another program text file.

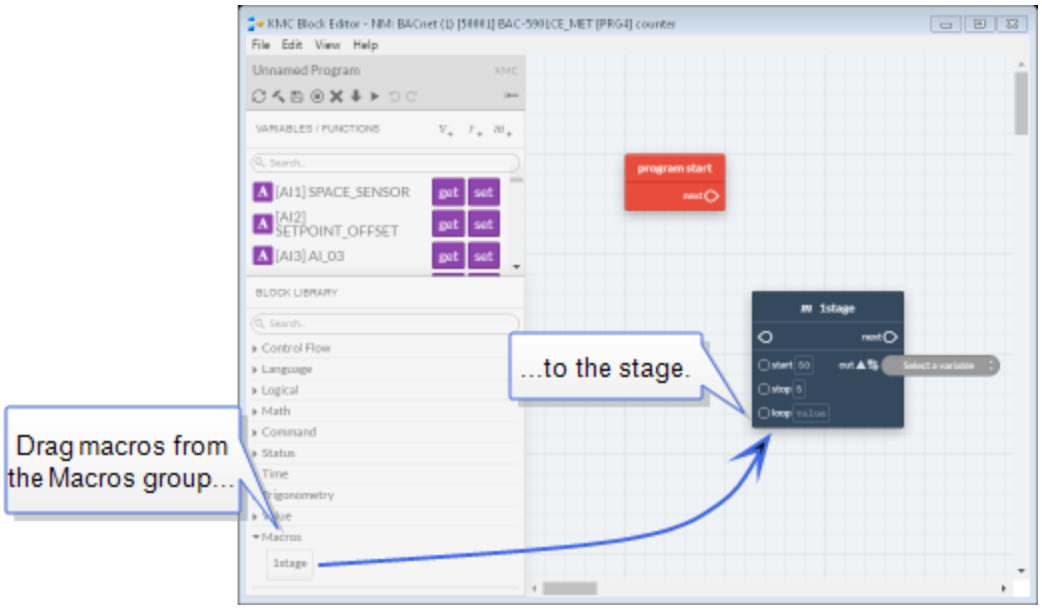
   Note: To write to an output port at a BACnet priority other than 9, add an at sign (@) followed by the priority number after the name of the output port. For example "result@8".
4. To save the macro for reuse in other programs, click **Save macro**.
5. When finished, click the close editor button ⊠.



Reusing macros

Saved macro blocks can be reused in other programs in either the same controller or programs in other controllers.

To reuse a saved macro, drag it from the Macros group in the Blocks Library to the stage.



To move a Macro block to another computer, copy the file in Windows Explorer. Macro block files are located at `C:\ProgramData\KMC Controls\Block Program Blocks`.

Limitations and importing

The block programming editor will attempt to import programs from the line editor. However some programs may be too complex and will require revision before they can be imported.

Capitalization All keywords in the line program must be capitalized before importing.

Unsuported keywords The following keywords are not supported with blocks in block programming.

- ALIAS
- FUNCTION and the associated keywords ENDFUNC and FRETURN
- GOTO
- HALT
- NETPOINT
- NETPOINTCACHE
- ON GOTO
- ON GOSUB
- ONERROR
- PROCEDURE and the associated keywords ENDPROC and PRETURN keywords
- STRING
- WAIT

REM Remark statements are not imported. They can be added in the Block Editor once the program is imported.

Multiple states on one line Statements on the same line separated by a colon (:) are not supported.

Off panel points Reading from and writing to objects that are not in the same controller must be done with line programming. KMC Controls recommends using just one program for all off-panel reading and writing and then storing the results in value objects.

Section 27: Reference to Control Basic blocks

This section covers the blocks for the Control Basic block programming.

Abs block

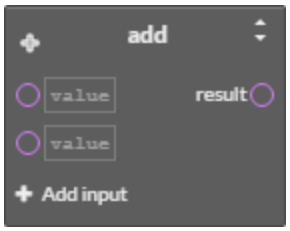
Block type: Math

The Abs block returns the absolute value of the input. The input at ***a*** can be a Get block or the output of a math block. Connect the ***result*** connector to another math block or a Set block.

Add block

Block type: Math

The Add block performs addition on all of the input values. The inputs at the ***value*** connectors can be from a constant, a Get block, or the result of a math block. Connect the ***result*** connector to the input of another logical block or a Set block.



The Add block can be changed to any of the following types of blocks by clicking on the double-arrows and choosing a new function.

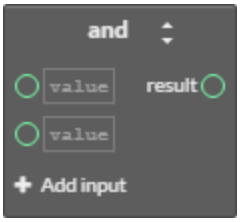
- Multiply
- Divide
- Integer divide
- Mod
- Subtract
- Power

Add additional inputs to the block by clicking **+ Add Input**.

And block

Block type: Logical

The And block performs the logical ***AND*** of two expressions. The result is ***TRUE*** if both expressions are non-zero; otherwise, the result is ***FALSE***. The inputs at the ***value*** connectors can be from constants or other blocks.



The And block can be changed to an Or or an Xor block by clicking on the double-arrows and choosing the new function.

Add additional inputs to the block by clicking **+ Add Input**.

See also the topics ***Or block*** on page 275 and ***Xor block*** on page 287.

AntiCycle block

Block type: Command

Use the AntiCycle macro block to prevent an output from rapidly cycling. A typical use for this block is a timed lockout for compressors or motors.

- The value at ***MinOnTime*** sets the minium time that ***output*** will be ***TRUE*** (1) after ***input1*** is ***FALSE*** (0).
- The value at ***MinOffTime*** sets the maximum time that ***output*** must be ***FALSE*** (1) after ***input1*** is ***TRUE*** (0).
- The values for ***MinOnTime*** and ***MinOffTime*** can be from blocks or constants.
- The variable for ***output*** must be a binary object or a local variable.

Arccos block

Block type: Trigonometry

The Arccos block calculates the arccosine of the value at ***a***. The value at ***result*** is expressed in radians. The value at ***a*** can be from a constant, a Get block, or the result of a math block. The block is typically used with a math block or connected to a Set block.

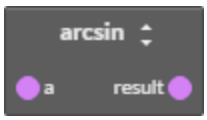
The Arccos block can be changed to any of the following types of blocks by clicking on the double-arrows and choosing a new function.

- Arcsin
- Arctan
- Cos
- Sine
- Tan

Arcsin block

Block type: Trigonometry

The Arcsin block calculates the arcsine of the value at ***a***. The value at ***result*** is expressed in radians. The value at ***a*** can be from a constant, a Get block, or the result of a math block. The block is typically used with a math block or connected to a Set block.



The Arcsin block can be changed to any of the following types of by clicking on the double-arrows and choosing a new function.

- Arcos
- Arctan
- Cos
- Sine
- Tan

Arctan block

Block type: Trigonometry

The Arctan block calculates the arctangent of the value at ***a***. The value at ***result*** is expressed in radians. The value at ***a*** can be from a constant, a Get block, or the result of a math block. The block is typically used with a math block or connected to a Set block.

The Arctan block can be changed to any of the following types of by clicking on the double-arrows and choosing a new function.

- Arcos
- Arcsin
- Cos
- Sine
- Tan

Avg block

Block type: Math

The Avg block returns the average value of the values connected to the inputs. Connect the ***result*** connector to another math block or a Set block.

The block can be changed to a Max or Min block by clicking on the double-arrows and choosing a new function.

Add additional inputs to the block by clicking **+ Add Input**.

An average can also be calculated with the *MinMaxAvg block* on page 271.

Clear block

Block type: Command

The Clear block resets the value of all local variables to zero.

Close bock

Block type: Command

The Close block sets the value of the input block to zero (0) for analog objects and variables and ***inactive*** for binary objects and variables.

The Close block can be changed to any of the following types of blocks by clicking on the double-arrows and choosing a new command.

- Open
- Start
- Stop
- Enable
- Disable

Compare block

Block type: Logical

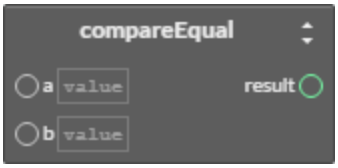
The Compare block is a macro block that includes an optional deadband. The input and deadband (***HI-DB, LO-DB***) values can be from Get blocks, math blocks, or constants.

- If ***input1*** is greater than ***input2*** plus the value at ***HI-DB***, the output variable is ***TRUE***.
- If ***input1*** is less than ***input2*** minus the value at ***LO-DB***, the output variable is ***FALSE***.
- Values for ***HI-DB*** and ***LO-DB*** are required. Enter zero (0) if no value is required.
- Only binary objects or local variables can be selected for ***output***.

CompareEqual block

Block type: Logical

The CompareEqual block tests whether one value is greater than the other. If the value at ***a*** is greater than the value at ***b***, the result is logically true. The values at ***a*** and ***b*** can be from a Get block, the result of a math block, or a constant entered in ***value***. The block is typically used as the ***condition*** for an *If block* or connected to a Set block.



The CompareEqual block can be changed to any of the following types of blocks by clicking on the double-arrows and choosing a new function.

- CompareGreater
- CompareGreaterEqual
- CompareLess
- CompareLessEqual
- CompareNoEqual

CompareGreater block

Block type: Logical

The CompareGreater block tests if one value is greater than a second value. If the value at ***a*** is greater than the value at ***b***, the result is logically true. The values at ***a*** and ***b*** can be from a Get block, the result of a math block, or a constant entered in ***value***. The block is typically used as the ***condition*** for an *If block* or connected to a Set block.

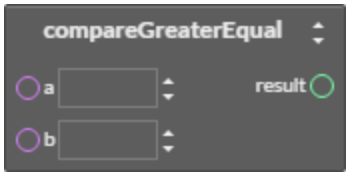
The CompareGreater block can be changed to any of the following types of blocks by clicking on the double-arrows and choosing a new function.

- CompareEqual
- CompareGreaterEqual
- CompareLess
- CompareLessEqual
- CompareNotEqual

CompareGreaterEqual block

Block type: Logical

The CompareGreaterEqual block is a logical block that tests if one value is greater than or equal to a second value. If the value at ***a*** is greater than or equal to the value at ***b***, the result is logically true. The values at ***a*** and ***b*** can be from a Get block, the result of a math block, or a constant entered in ***value***. The block is typically used as the ***condition*** for an *If block* or connected to a Set block.



The CompareGreater block can be changed to any of the following types of blocks by clicking on the double-arrows and choosing a new function.

- CompareEqual
- CompareGreaterEqual
- CompareLess
- CompareLessEqual
- CompareNotEqual

CompareLess block

Block type: Logical

The CompareLess block tests if one value is less than a second value. If the value at ***a*** is less than the value at ***b***, the result is logically true. The values at ***a*** and ***b*** can be from a Get block, the result of a math block, or a constant entered in ***value***. The block is typically used as the ***condition*** for an *If block* or connected to a Set block.

The CompareLess block can be changed to any of the following types of blocks by clicking on the double-arrows and choosing a new function.

- CompareEqual
- CompareGreater
- CompareGreaterEqual
- CompareLessEqual
- CompareNotEqual

CompareLessEqual block

Block type: Logical

The CompareLessEqual block tests if one value is less than or equal to a second value. The block is typically used as the ***condition*** for an *If block* or connected to a Set block.

If the value at ***a*** is less than or equal to the value at ***b***, the result is logically true. The values at ***a*** and ***b*** can be from a Get block, the result of a math block, or a constant entered in ***value***.

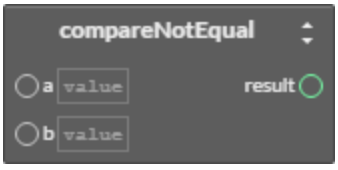
The CompareLessEqual block can be changed to any of the following types of blocks by clicking on the double-arrows and choosing a new function.

- CompareEqual
- CompareGreater
- CompareGreaterEqual
- CompareLess
- CompareNotEqual

CompareNotEqual block

Block type: Logical

The CompareNotEqual blocks tests whether one value is not equal to a second value. The block is typically used as the ***condition*** for an *If block* or connected to a Set block. If the values at ***a*** and ***b*** are not equal, the result is logically true. The values at ***a*** and ***b*** can be from a Get block, the result of a math block, or a constant entered in ***value***.



The CompareNotEqual block can be changed to any of the following types of blocks by clicking on the double-arrows and choosing a new function.

- CompareEqual
- CompareGreater
- CompareGreaterEqual
- CompareLess
- CompareLessEqual

Cos block

Block type: Trigonometry

The Cos block calculates the cosine of the value at ***a***. The value at ***result*** is expressed in radians. The block is typically used with a math block or connected to a Set block. The value at ***a*** can be from a constant, a Get block, or the result of a math block.

The Cos block can be changed to any of the following types of by clicking on the double-arrows and choosing a new function.

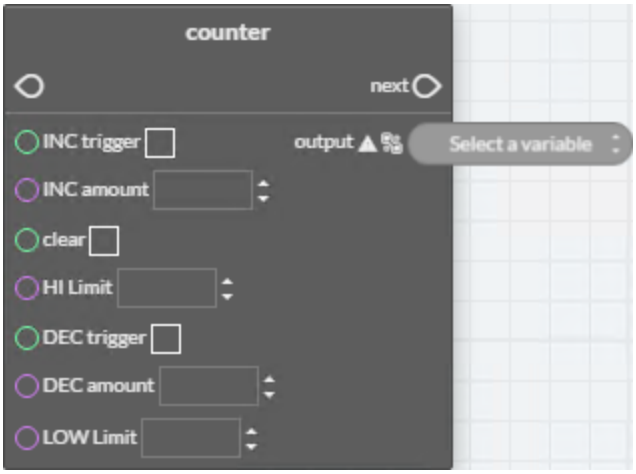
- Arccos
- Arcsin
- Arctan
- Sine
- Tan

Counter block

Block type: Math

Use the Counter macro math block to change the value of the output by a fixed amount every time an input is triggered. This can be useful for installations such as ramping a supply fan frequency drive during start up by triggering ***INC trigger*** once a minute until an operational static pressure is reached. After start up, a PID loop would then be used to modulate the frequency drive.

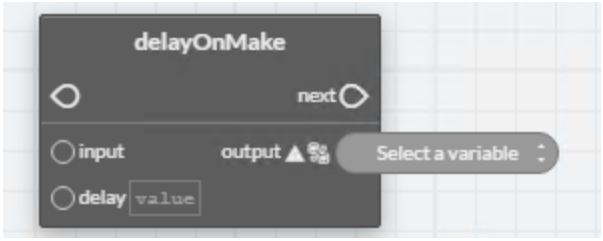
- A transition from ***FALSE*** (0) to ***TRUE*** (1) at ***INC trigger*** or ***DEC trigger*** will increase or decrease the value at ***output*** by the amount in ***INC amount*** or ***DEC amount***.
- The ***HI Limit*** and ***LOW Limit*** input sets the minimum and maximum value for output. These inputs are optional and can be from other blocks or constants.
- Setting ***clear*** to ***TRUE*** (1) resets and holds the output value to zero (0).
- The variable for ***output*** must be an analog object or local variable.



DelayOnMake block

Block type: Time

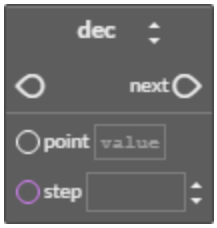
The DelayOnMake macro block sets ***output*** to ***TRUE*** when ***input*** changes to ***TRUE*** after the period set by ***delay***. The values for ***delay*** and ***input*** can be from Get blocks, other blocks, or constants. The variable for ***output*** must be a binary object or a local variable.



Dec block

Block type: Math

The Dec block decrements the value at ***point*** by the value at ***step***. The values at ***step*** and ***point*** can be from constants, Get blocks, or the results from math blocks.



The Dec block can be changed to an *Inc block* by clicking on the double-arrows and choosing the new function.

DelayOnBreak block

Block type: Time

The DelayOnBreak macro block extends the time that ***output*** is *TRUE* for the period set by ***delay*** after the ***input*** condition changes to *FALSE*. The values for ***delay*** and ***input*** can be from Get blocks, other blocks, or constants. The variable for ***output*** must be a binary object or a local variable.

See also the topic *DelayOnMake block* on page 256.

Dewpoint block

Block type: Value

The Dewpoint block calculates the dew point in degrees Fahrenheit based on air temperature–in degrees Fahrenheit–and humidity. Humidity at ***hum*** and temperature at ***temp*** can be from constants, Get blocks, or the results of math blocks. The block is typically used with a math block or connected to a Set block.

The Dewpoint block can be changed to any of the following types of blocks by clicking on the double-arrows and choosing a new function.

- Enthalpy
- Enthalpysi
- Dewpointsi

To calculate dew point in degrees Celsius, see the topic *Dewpointsi block* on page 258.

Dewpointsi block

Block type: Value

The Dewpointsi block returns the dew point in degrees Celsius based on outside air temperature, in degrees Celsius, and outside humidity. Humidity at ***hum*** and temperature at ***temp*** can be from can be from constants, Get blocks, or the results of math blocks. The block is typically used with a math block or connected to a Set block.

The Dewpointsi block can be changed to any of the following types of blocks by clicking on the double-arrows and choosing a new function.

- Dewpointsi
- Enthalpy
- Enthalpysi

To calculate dew point in degrees Fahrenheit, see the topic *Dewpoint block* on page 257.

Disable block

Block type: Command

The Disable block is a command block that sets the value of the input block to zero (0) for analog objects and variables and ***inactive*** for binary objects and variables.

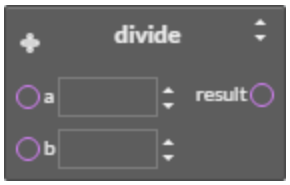
The Disable block can be changed to any of the following types of blocks by clicking on the double-arrows and choosing a new command.

- Close
- Open
- Start
- Stop
- Enable

Divide block

Block type: Math

The Divide block divides the value of ***a*** by the value of ***b***. The values at ***a*** and ***b*** can be from constants, Get blocks, or the results from math blocks. Connect the ***result*** connector to the input of another math block or a Set block.



The Divide block can be changed to any of the following types of blocks by clicking on the double-arrows and choosing a new function.

- Multiply
- Integer divide
- Mod
- Add
- Subtract
- Power

See the related topics *IntegerDivide block* on page 267 and *Mod block* on page 271.

Dom block

Block type: Time

The Dom block returns the current day of the month. The block is typically used with a logical block, or connected to a Set block.

The Dom block can be changed to any of the following types of blocks by clicking on the double-arrows and choosing a new function.

- Dow
- Doy
- Month
- Time
- Year

Dow block

Block type: Time

The Dow block returns a numerical value for the day of the week. In BACnet controllers Monday is day 1 and Sunday is day 7. The block is typically used with a logical block, or connected to a Set block.

The Dow block can be changed to any of the following types of blocks by clicking on the double-arrows and choosing a new function.

- Dom
- Doy
- Month
- Time
- Year

Doy block

Block type: Time

The Doy block returns the current day of the year. The block is typically used with a logical block, or connected to a Set block.

- The year always begins on January 1.
- December 31st is day 366.
- February is always counted as having 29 days which means March 1 is always day 61.
- On non-leap years, February 29 (day 60) is skipped.

The Doy block can be changed to any of the following types of blocks by clicking on the double-arrows and choosing a new function.

- Dom
- Dow
- Month
- Time
- Year

Enable block

Block type: Command

The Enable block sets the value of the input block to one (1) for analog objects and variables and ***active*** for binary objects and variables.

The Enable block can be changed to any of the following types of blocks by clicking on the double-arrows and choosing a new command.

- Close
- Open
- Start
- Stop
- Disable

End block

Block type: Language

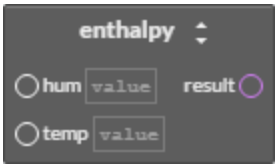
The End block terminates the execution of a program. When the block is encountered, the program stops reading blocks and exits the program.

The block programming editor automatically adds an End block. Other End blocks my be required by program design.

Enthalpy block

Block type: Value

The Enthalpy block calculates enthalpy based on air temperature–in degrees Fahrenheit–and humidity. The value at ***result*** is expressed as BTUs per pound of air. Humidity at ***hum*** and temperature at ***temp*** can be from can be from constants, Get blocks, or the results of math blocks. The block is typically used with a math block or connected to a Set block.



The Enthalpy block can be changed to any of the following types of blocks by clicking on the double-arrows and choosing a new function.

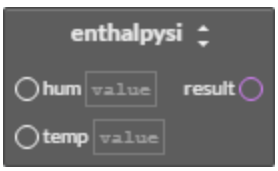
- Enthalpysi
- Dewpoint
- Dewpointsi

To calculate enthalpy as kilojoules per kilogram of air, see the topic ***Enthalpysi block*** on page 262.

Enthalpysi block

Block type: Value

The Enthalpysi block calculates enthalpy based on air temperature–in degrees Celsius–and humidity. The value at ***result*** is expressed as kilojoules per kilogram of air. Humidity at ***hum*** and temperature at ***temp*** can be from can be from constants, Get blocks, or the results of math blocks. The block is typically used with a math block or connected to a Set block.



The Enthalpysi block can be changed to any of the following types of blocks by clicking on the double-arrows and choosing a new function.

- Enthalpy
- Dewpoint
- Dewpointsi

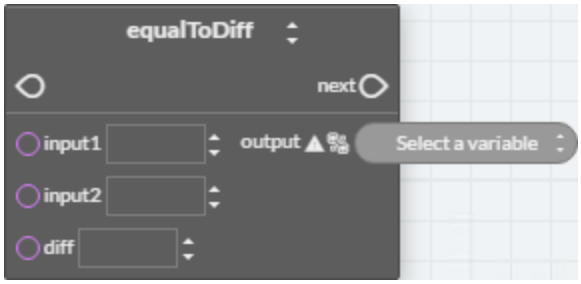
To calculate enthalpy as BTUs per pound of air, see the topic *Enthalpy block* on page 262.

EqualToDiff block

Block type: Logical

The EqualToDiff block is a comparison macro block. The values for ***input1***, ***input2***, and ***diff*** can be from Get blocks, math blocks, or constants.

- If ***input1*** is equal to ***input2***, the output variable is *TRUE*.
- If ***input1*** is greater than ***input2*** plus the value at ***diff*** or ***input1*** is less than ***input2*** minus the value at ***diff***the output variable is *FALSE*.
- An entry for ***diff*** is required. Enter zero (0) if no value is required.
- Only binary objects or local variables can be selected for ***output***.

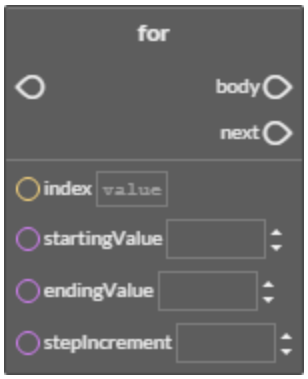


The EqualToDiff block can be changed to LessThanDiff or GreaterThanDiff blocks by clicking on the double-arrows and choosing a new function.

For block

Block type: Control flow

The For block adds a For To Next loop to the program. The loop repeats a set of instructions a specific number of times.



index is the variable or value that increments each time the loop repeats. It controls whether or not Control Basic repeats the loop. The value at ***index*** must be local to the controller in which the Control Basic program is running.

startingValue is the initial value that Control Basic assigns to ***index*** value.

endingValue is the value that the ***index*** value must equal before the loop ends.

stepIncrement is the amount that Control Basic adds to ***index*** with each iteration of the loop. The value at ***stepIncrement*** can be a positive or negative value.

body connects to the program blocks that are executed inside of the loop.

GreaterThanDiff block

Block type: Logical

The GreaterThanDiff block is a comparison macro block. The values for ***input1***, ***input2***, and ***diff*** can be from Get blocks, other blocks, or constants.

- If ***input1*** is greater than or equal to ***input2***, the output variable is *TRUE*.
- If ***input1*** is less than ***input2*** minus the value at ***diff***, the output variable is *FALSE*.
- An entry for ***diff*** is required. Enter zero (0) if no value is needed.
- Only binary objects or local variables can be selected for ***output***.

The GreaterThanDiff block can be changed to equalToDiff or lessThanDiff block by clicking on the double-arrows and choosing a new function.

Hsel block

Block type: Logical

The Hsel block selects the highest (second highest, etc.) from value of blocks connected to the inputs. The value at ***N*** defines whether it selects the highest (1) or the second highest (2) etc. An input block can be a Get block, the result of a math block, or a constant entered in *value*.

The Hsel block can be changed to an *Lsel block* by clicking on the double-arrows and choosing the new function.

Add additional inputs to the block by clicking **+ Add Input**.

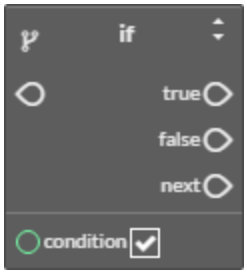
See also the topic *Lsel block* on page 269.

If block

Block type: Control flow

The If block is a decision making block based on the state of the input at the ***condition*** connector. The ***condition*** input can be from a Get block or the results of from another block with a value that is either true or false (high or low, on or off, etc.)

- If ***condition*** is ***true*** the block connected to the ***true*** connector is executed.
- If the ***condition*** is ***false*** (not true) the block connected to the ***false*** connector is executed.
- A block connected to the ***next*** connector is the next block executed in the program flow after the blocks connected to either the ***true*** or ***false*** connectors are executed.



The If block can be changed to an IfPlus or IfMinus block by clicking on the double-arrows and choosing the new function.

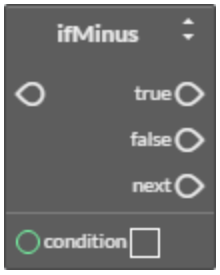
See also the topics *IfMinus block* on page 265 and *IfPlus block* on page 266.

IfMinus block

Block type: Control flow

TheIfMinus block is a decision making block that detects the first time a condition changes from ***true*** to ***false***. If the condition is false and on the previous scan it was true, the block connected to the ***true*** connector will be executed otherwise the block connected to the ***false*** connecter will be executed.

A block connected to the ***next*** connector is the next block executed in the program flow.



The IfPlus block can be changed to an If or IfPlus block by clicking on the double-arrows and choosing the new function.

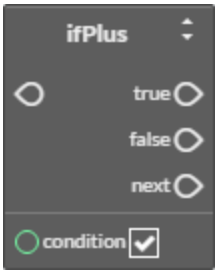
See also the topics *If block* on page 265 and *IfPlus block* on page 266.

IfPlus block

Block type: Control flow

The IfPlus block is a decision making block that detects the first time a condition changes from ***false*** to ***true***. If the condition is true and on the previous scan it was not true, the block connected to the ***true*** connector will be executed otherwise the block connected to the ***false*** connecter will be executed.

A block connected to the ***next*** connector is the next block executed in the program flow.



The IfPlus block can be changed to an If or IfMinus block by clicking on the double-arrows and choosing the new function.

See also the topics *If block* on page 265 and *IfMinus block* on page 265.

Inc block

Block type: Math

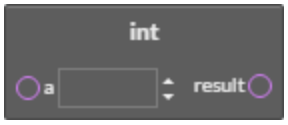
The Inc block increments the value at ***point*** by the value at ***step***. The values at ***step*** and ***point*** can be from constants, Get blocks, or the results from math blocks.

The Inc block can be changed to a *Dec block* by clicking on the double-arrows and choosing the new function.

Int block

Block type: Math

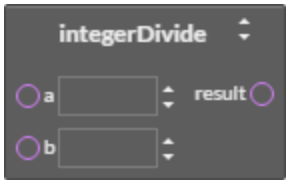
The Int block returns the integer portion of the block connected to ***a***. The value returned is the greatest integer that is less than or equal to the value of ***expression***. The value at ***a*** can be a constant, a Get block, or the result of a math block. Connect the ***result*** connector to another math block or a Set block.



IntegerDivide block

Block type: Math

The IntergerDivide block divides the value of ***a*** by the value of ***b*** and returns the only the integer portion. The values at ***a*** and ***b*** can be from constants, Get blocks, or the results of math blocks. Connect the ***result*** connector to the input of another math block or a Set block.



The IntegerDivide block can be changed to any of the following types of blocks by clicking on the double-arrows and choosing a new function.

- Multiply
- Divide
- Mod
- Add
- Subtract
- Power

Interval block

Block type: Time

The Interval block is a repeating count-up timer. When the timer reaches the value at ***a***, ***result*** is true otherwise ***result*** is false. The Interval block is typically used with Compare and If blocks.

InvIn block

Block type: Math

The InvIn block returns the inverse natural logarithm of the value at ***a***. The value at ***a*** can be a constant, a Get block, or the result of a math block. Connect the ***result*** connector to another math block or a Set block.

LessThanDiff block

Block type: Logical

The LessThanDiff block is a comparison macro block. The values for ***input1***, ***input2***, and ***diff*** can be from Get blocks, math blocks, or constants.

- If ***input1*** is less than or equal to ***input2***, the output variable is *TRUE*.
- If ***input1*** is greater than ***input2*** plus the value at ***diff***, the output variable is *FALSE*.
- An entry for ***diff*** is required. Enter zero (0) if no value is required.
- Only binary objects or local variables can be selected for ***output***.

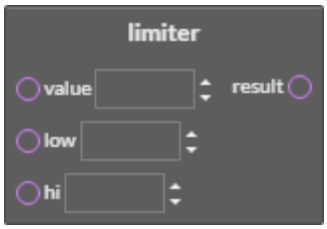
The LessThanDiff block can be changed to a GreaterThanDiff or EqualToDiff block by clicking on the double-arrows and choosing a new function.

Limiter block

Block type: Logical

The Limiter block is a macro block that limits the input at ***value*** to the range defined by ***low*** and ***hi***. The values at ***value***, ***low***, and ***hi*** can be from constants, Get blocks, or the results of other blocks. Connect the ***result*** connector to the input of another math block or a Set block.

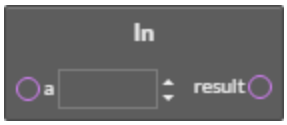
Tip: Connecting a Get block to ***value*** and a Set block to ***result*** from the same object tests and limits the objects value.



Ln block

Block type: Math

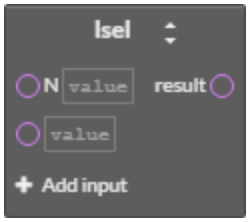
The Ln block returns the natural logarithm of the value at ***a***. The value at ***a*** can be a constant, a Get block, or the result of a math block. Connect the ***result*** connector to another math block or a Set block.



Lsel block

Block type: Logical

The Lsel block selects the lowest, second lowest, etc, value of blocks connected to the input. The value at ***N*** defines whether it selects the lowest (1) or the second lowest(2) etc. The input blocks can be value, inputs, or output objects, calculations, etc. An input block can be a Get block, the result of a math block, or a constant entered in ***value***.



The Lsel block can be changed to an *Hsel block* by clicking on the double-arrows and choosing the new function.

Add additional inputs to the block by clicking **+ Add Input**.

See also the topic *Hsel block* on page 264.

Max block

Block type: Logical

The Max block returns the maximum value of the blocks connected to the inputs. An input block can be a Get block, the result of a math block, or a constant entered in ***value***.

The Max block can be changed to an Avg or Min block by clicking on the double-arrows and choosing the new function.

Add additional inputs to the block by clicking **+ Add Input**.

Min block

Block type: Logical

The Min block returns the minimum value of the blocks connected to the inputs. An input block can be a Get block, the result of a math block, or a constant entered in ***value***.

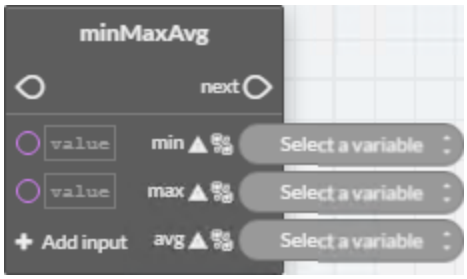
The Min block can be changed to an Avg or Max block by clicking on the double-arrows and choosing the new function.

Add additional inputs to the block by clicking **+ Add Input**.

MinMaxAvg block

Block type: Logical

The MinMaxAvg block is a macro block that returns the minimum, maximum, and average values of the values connected to the inputs. The values at the inputs can be from constants, Get blocks, or the results of math blocks. The ***min***, ***max***, and ***avg*** variables can be analog objects or local variables.



Add additional inputs to the block by clicking **+ Add Input**.

See also the following topics.

- *Avg block* on page 250
- *Max block* on page 270
- *Min block* on page 270

Mod block

Block type: Math

The Mod block returns the remainder of ***a*** divided by ***b***. The values at ***a*** and ***b*** can be from constants, Get blocks, or the results from math blocks. Connect the result connector to the input of another math block or a Set block.

The Mod block can be changed to any of the following types of blocks by clicking on the double-arrows and choosing a new function.

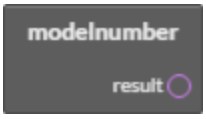
- Multiply
- Divide
- Integer divide
- Add
- Subtract
- Power

See the related topics *Divide block* on page 259 and *IntegerDivide block* on page 267.

ModelNumber block

Block type: Status

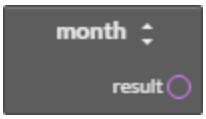
Use the ModelNumber block to set a variable or property to the numerical portion of the model number of the controller. The block is typically used with an IF block, a logical block, or connected to a Set block.



Month block

Block type: Time

The Month block returns the current month of the year. The block is typically used with a logical block, or connected to a Set block.



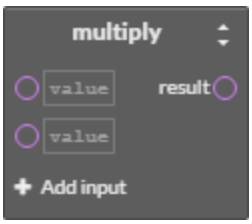
The Month block can be changed to any of the following types of blocks by clicking on the double-arrows and choosing a new function.

- Dom
- Dow
- Doy
- Time
- Year

Multiply block

Block type: Math

The Multiply block multiplies the input values. The input values can be from constants, Get blocks, or the results from math blocks. Connect the ***result*** connector to the input of another math block or a Set block.



The Multiply block can be changed to any of the following types of blocks by clicking on the double-arrows and choosing a new function.

- Divide
- Integer divide
- Mod
- Add
- Subtract
- Power

Add additional inputs to the block by clicking **+ Add Input**.

Nan block

Block type: Math

Use the Nan block to set a variable or property to a ***Not A Number*** constant or to test if the variable or property is equal to ***Not A Number***. The block is typically used as the ***condition*** for an *If block* or connected to a Set block.

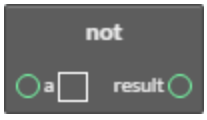
Netsensorstatus block

Block type: Status

The Netsensorstatus block returns ***TRUE*** if a functional NetSensor is connected to the controller and ***FALSE*** if the controller does not detect a NetSensor. The block is typically used with an IF block, a logical block, or connected to a Set block.

Not block

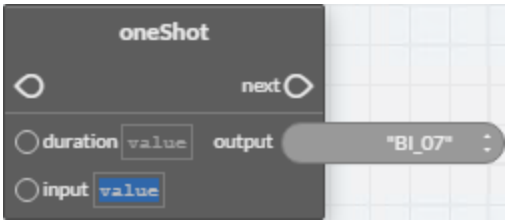
The Not block performs a logical negation operation on the input. If the input is 0, the result is 1. If the input is non-zero, the result is 0. The block at ***a*** can be from a Get block or the result of a math block.



OneShot block

Block type: Time

The OneShot block sets the variable at ***output*** to *TRUE* the first time ***input*** changes from *FALSE* to *TRUE*. The output variable remains *TRUE* for the period set by ***duration*** and then changes to *FALSE*. The values for ***duration*** and ***input*** can be from Get blocks, other blocks, or constants. The variable for ***output*** must be a binary object or a local variable.



See also the topic *IfPlus block* on page 266

Open block

Block type: Command

The Open block is a command block that sets the value of analog objects and variables to one (1) and binary objects and variables to ***active***.

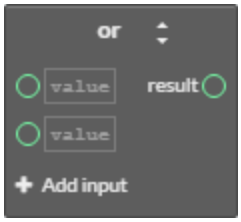
The Open block can be changed to any of the following types of blocks by clicking on the double-arrows and choosing a new command.

- Close
- Enable
- Start
- Stop
- Disable

Or block

Block type: Logical

The Or block performs a logical ***OR*** of the values of the input blocks. The result is ***true*** if any of the inputs are ***true***. The result is ***false*** if all inputs are ***false***. The values at ***a*** and ***b*** can be from constants or other blocks.



The Or block can be changed to an And or Xor block by clicking on the double-arrows and choosing the new function.

Add additional inputs to the block by clicking **+ Add Input**.

See also the topics *And block* on page 248 and *Xor block* on page 287.

Outputoverride block

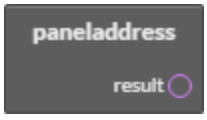
Block type: Status

The Outputoverride block returns the switch position of an optional HPO-6700 series output board installed in the controller in which Control Basic is running. The block returns *FALSE* if the switch is in *AUTO* and *TRUE* if the switch is set to either the *OFF* or *HAND* position. The output object instance can be a numerical constant entered at ***a*** or a Get block.

Paneladdress block

Block type: Status

Use the Paneladdress block to set a variable or property to the device instance of the controller. The block is typically used with an IF block, a logical block, or connected to a Set block.



Pi block

Block type: Math

Use the Pi block to set a variable or property to the value of the constant pi. The block is typically used with a math block or connected to a Set block.

Power block

Block type: Math

The Power block raises the value of ***a*** by the value of ***b***. The values at ***a*** and ***b*** can be from constants, Get blocks, or the results from math blocks. Connect the ***result*** connector to the input of another math block or a Set block.

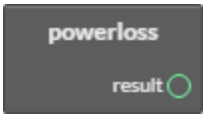
The Power block can be changed to any of the following types of blocks by clicking on the double-arrows and choosing a new function.

- Multiply
- Divide
- Integer divide
- Mod
- Add
- Subtract

Powerloss block

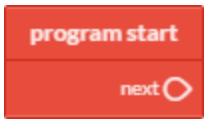
Block type: Status

The Powerloss block returns ***TRUE*** on the first scan of all Control Basic programs after power is restored. After the first scan, it returns as ***FALSE***. The block is typically used with an If block, a logical block, or connected to a Set block.



Program start block

The Program Start block is automatically added when a new block program is started. The block is the beginning of the program and cannot be removed. Other blocks are added as needed to build the program.



Rem block

Block type: Language

A Rem block inserts explanatory comments or remarks into the program. Use Rem blocks to document the use of a subroutine or to explain a calculation.

Reset block

Block type: Math

The Reset block is a macro block that calculates an output in one unit of measure—for example, a boiler hot water setpoint—from an input in another unit of measure such as outside air temperature in degrees Fahrenheit.

- Input values—***OUT_MAX, OUT_MIN, IN_MAX, IN_MIN***, and ***input***—can be from Get blocks, the results of other blocks, constants or local variables.
- The variables for ***slope*** and ***y-intercept*** are required and are usually local variables.
- The ***output*** variable can be an object within the controller or a local variable.
- The output value is limited to the range set by ***OUT_MIN*** and ***OUT_MAX***.

The Reset block can be changed to a *Slope block* by clicking on the double-arrows and choosing the new function.

Rlq block

Block type: Command

The Rlq bloc relinquishes the priority level of the BACnet output or value object connected to ***a***.

To set the priority level, do the following:

1. Select the block. A selected block has a yellow border.
2. Click the priority arrow above the box.
3. Click the up and down arrows in the priority list to choose the new priority value.

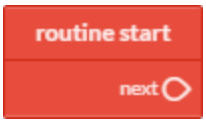
Rnd block

Block type: Math

The Rnd block calculates a random number between 0 and the value at ***a***-1. The value at ***a*** can be a constant, a Get block, or the result of a math block. Connect the ***result*** connector to another math block or a Set block.

Routine Start block

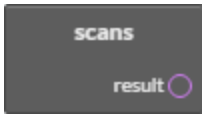
The Routine Start block is automatically added to new routines. The block is the beginning of the routine and cannot be removed. See the topic *Adding and editing routines* on page 240.



Scans block

Block type: Time

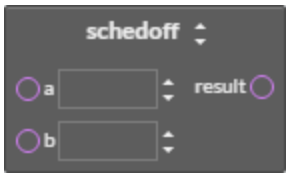
The Scan block returns the rate a controller is processing all Control Basic programs. As the complexity or length of a program increases it takes longer to process and the scan rate decreases. The value at ***result*** is expressed in scans per second.



Schedoff block

Block type: Time

The Schedoff block returns the time that the schedule object at ***a*** will set its reference object to ***Inactive*** or a value of zero(0).



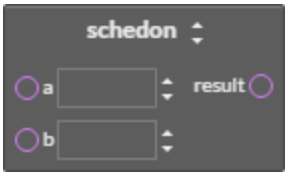
- The returned value is the difference—in seconds—from the time specified at ***b*** to the time that the schedule's present value will be ***Inactive*** or zero(0).
- The schedule object at ***a*** must be within the same device.
- Both the schedule object at ***a*** and the time at ***b*** may be expressed as either a local variable, value (12:00, 6) or value object (AV1).
- The time parameter at ***b*** is entered in 24-hour format.
- The value returned is based on the current day of the week.
- A returned value of 0 indicates that the schedule is already set to off.
- A return of 86,400 indicates that there are no more scheduled Off times for the current day.

See also the topic *Schedon block* on page 280.

Schedon block

Block type: Time

The Schedon block returns the time that the schedule object at ***a*** will set the present value of the reference object to ***Active*** or a non-zero value.



- The returned value is the difference—in seconds—from the time specified at ***b*** to the time that the schedule's present value will become ***Active*** (or non-zero).
- The schedule object at ***a*** must be within the same device. It can be from a Get block or the number of the schedule object entered in ***a***.
- The time parameter at ***b*** is entered in 24-hour format.
- The value returned is based on the current day of the week.
- A returned value of 0 indicates that the schedule is already set to off.
- A return of 86,400 indicates that there are no more scheduled Off times for the current day.

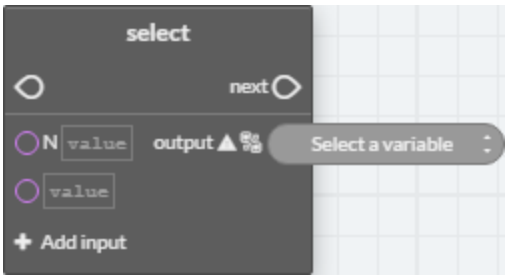
See also the topic *Schedoff block* on page 279.

Select block

Block type: Command

Use the Select macro block to choose a value from a list of values. This block is useful to choose a setpoint based on a system mode.

- The value at ***N*** points to the position in the list of input values to use as the output value.
- ***N*** must be an integer. Non integer values will not change the output value.
- The ***value*** inputs can be from blocks or a constants.
- The variable for ***output*** must be a analog object or a local variable.

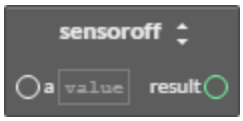


Add additional inputs to the block by clicking **+ Add Input**.

Sensoroff block

Block type: Value

Use the Sensoroff block to detect an open-circuit condition at an analog input. The input at ***a*** is typically a Get block for an analog input. The result is usually connected to an if, ifMinus, or ifPlus block.



The Sensoroff block can be changed to a *Sensoron block* on page 282 by clicking on the double-arrows and choosing the new function.

A typical application is to detect momentary conditions such as a pressed button. If the opened contact condition lasts longer than two minutes the function will be disabled. After three minutes, the object will change ***Out Of Service*** to ***true*** but the commands will still execute.

When used with *Sensoroff block* on page 281 and *If block* on page 265, *IfMinus block* on page 265, or *IfPlus block* on page 266 you can determine three separate conditions from one input:

- A temperature or other analog reading.
- A sensor with open contacts (SENSOROFF).
- A sensor with closed contacts (SENSORON).

SENSOROFF can also be used with inputs using a table if the minimum value in the table is set to a value greater than zero and its maximum value is less than 5.00 volts.

Table 27-1 Example table for SENSOROFF in BACnet controllers

Input Voltage	Detected condition
0	Closed circuit
0.4	Temperature-55 degrees
4.9	Temperature-95 degrees
5.0 or greater	Open circuit

In the table *Example table for SENSOROFF in BACnet controllers* on page 281, the input voltage under normal temperature conditions will never fall below 0.4 volts. When a sensor is shorted to ground, the input voltage will fall to zero, which is a condition SENSORON can detect. Similarly, if the circuit is opened, the controller will read the open circuit voltage, which is higher than the maximum 4.9 volts in the table which will be detected by a sensoroff block.

Sensoron block

Block type: Status

Use a Sensoron block to detect 0 volts (closed-circuit) condition on an analog input. The input at ***a*** is typically a Get block for an analog input. The result is usually connected to an if, ifMinus, or ifPlus block.

The Sensoron block can be changed to a *Sensoroff block* on page 281 by clicking on the double-arrows and choosing the new function.

A typical application is to detect momentary conditions such as a pressed button. If the opened contact condition lasts longer than two minutes the function will be disabled. After three minutes, the object will change ***Out Of Service*** to ***true*** but the commands will still execute.

When used with the *Sensoroff block* on page 281and *If block* on page 265, *IfPlus block* on page 266, or *IfMinus block* on page 265 you can determine three separate conditions from one input:

- A temperature or other analog reading.
- A sensor with open contacts (SENSOROFF).
- A sensor with closed contacts (SENSORON).

The sensoron block can also be used with analog inputs using a table if the minimum value in the table is set to a value greater than zero and its maximum value is less than 5.00 volts.

Table 27-2 Input conditions for SENSORON for BACnet controllers

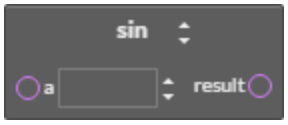
Input Voltage	Detected condition
0	Closed circuit
0.4	Temperature-55 degrees
4.9	Temperature-95 degrees
5.0 or greater	Open circuit

In the table *Input conditions for SENSORON for BACnet controllers*, the input voltage under normal temperature conditions would never fall below 0.4 volts. When a sensor is shorted to ground, the input voltage will fall to zero, which is a condition SENSORON can detect. Similarly, if the circuit is opened, the controller will read 5.00 volts, which is higher than the maximum 4.9 volts in the table which will be detected with a sensoroff block.

Sin block

Block type: Trigonometry

The Sin block calculates the sine of the value at ***a***. The value at ***result*** is expressed in radians. The block is typically used with a math block or connected to a Set block. The value at ***a*** can be from a constant, a Get block, or the result of a math block.



The Sin block can be changed to any of the following types of by clicking on the double-arrows and choosing a new function.

- Arccos
- Arcsin
- Arctan
- Cos
- Tan

Slope block

Block type: Logical

The Slope block is a macro block that calculates an output in one unit of measure—for example, a boiler hot water setpoint—from an input in another unit of measure such as outside air temperature in degrees Fahrenheit.

- Input values—***OUT_MAX, OUT_MIN, IN_MAX, IN_MIN***, and ***input***—can be from Get blocks, the results of other blocks, constants, or local variables.
- The variables for ***slope*** and ***y-intercept*** are required and are usually local variables.
- The ***output*** variable can be an object within the controller or a local variable.

The slope block can be changed to a *Reset block* by clicking on the double-arrows and choosing the new function.

Sqr block

Block type: Math

The Sqr block calculates the square root of the value at ***a***. The block is typically used with a math block or connected to a Set block. The value at ***a*** can be from a constant, a Get block, or the result of a math block.

Start block

Block type: Command

The Start block is a command block that sets the value of the input block to one (1) for analog objects and variables and ***active*** for binary objects and variables.

The Start block can be changed to any of the following types of blocks by clicking on the double-arrows and choosing a new command.

- Close
- Open
- Stop
- Enable
- Disable

Stop block

Block type: Command

The Stop block is a command block that sets the value of the input block to zero (0) for analog objects and variables and ***inactive*** for binary objects and variables.

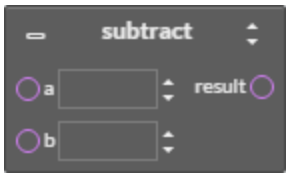
The Stop block can be changed to any of the following types of blocks by clicking on the double-arrows and choosing a new command.

- Close
- Open
- Start
- Disable
- Enable

Subtract block

Block type: Math

The Subtract block subtracts the value at ***b*** from the values at ***a***. The values at ***a*** and ***b*** can be from constants, Get blocks, or the results from math blocks. Connect the ***result*** connector to the input of another math block or a Set block.



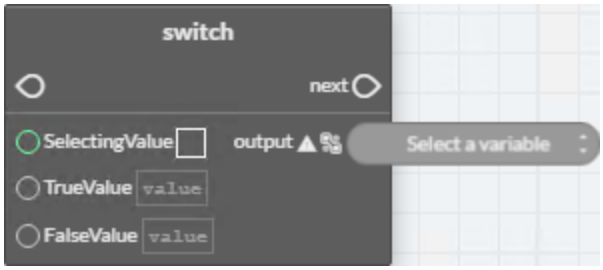
The subtract block can be changed to any of the following types of blocks by clicking on the double-arrows and choosing a new function.

- Multiply
- Divide
- Integer divide
- Mod
- Add
- Power

Switch block

Block type: Command

The Switch macro block switches the output between ***TrueValue*** when ***SelectingValue*** is ***TRUE***, and the ***FalseValue*** when ***SelectingValue*** is ***FALSE***. The values for ***SelectingValue***, ***TrueValue***, and ***FalseValue*** can be from Get blocks, other blocks, or constants. The variable for ***output*** can be any object within the controller or a local variable.



Tan block

Block type: Trigonometer

The Tan block calculates the tangent of the value at ***a***. The value at ***result*** is expressed in radians. The block is typically used with a math block or connected to a Set block. The value at ***a*** can be from a constant, a Get block, or the result of a math block.

The Tan block can be changed to any of the following types of by clicking on the double-arrows and choosing a new function.

- Arccos
- Arcsin
- Arctan
- Cos
- Sine

Tbl block

Block type: Value

Use the Tbl block to calculate a value based on the input at ***a*** and a Control Basic table. The input at ***a*** can be from a Get block for an analog object and must be an integer. The value at ***b*** designates the Control Basic table object within the controller.

Use the Tbl block when the value of the input is nonlinear or requires a complicated calculation to arrive at the proper value.

Time block

Block type: Time

The Time block returns the time of day—as the number of seconds after midnight—as maintained in the controller running Control Basic. The block is typically used with a logical block, or connected to a Set block.

The Time block can be changed to any of the following types of blocks by clicking on the double-arrows and choosing a new function.

- Dom
- Dow
- Doy
- Month
- Year

Within block

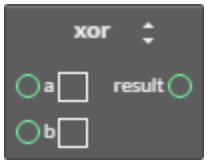
Block type: Logical

The Within block is a macro block that sets the output to *ACTIVE (1)* if the input at ***value*** is within the range set by ***HI*** and ***LO***. If value is not within the range, the output is set to *INACTIVE (0)*. The inputs at ***variable***, ***HI***, and ***LO*** can be from constants, a Get block, or the results from another block. The object selected by ***output*** is limited to binary objects or local variables.

Xor block

Block type: Math

The Xor block performs a logical exclusion on the two Boolean expressions at ***a*** and ***b***. The result is ***true*** if the two expressions are different; otherwise, the result is ***false***. The values at ***a*** and ***b*** can be from constants or other blocks.

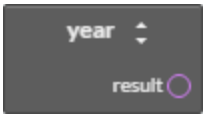


The Xor block can be changed to an And or Or block by clicking on the double-arrows and choosing the new function.

Year block

Block type: Time

The Year block returns the four-place value of the current year. The block is typically used with a logical block, or connected to a Set block.



The Year block can be changed to any of the following types of blocks by clicking on the double-arrows and choosing a new function.

- Dom
- Dow
- Doy
- Month
- Time

KMC Connect

Part VII: BACnet objects

Section 28: BACnet overview

The topics in this section are an overview of the BACnet objects, properties and services. It discusses also priority arrays and alarm basics.

The ASHRE BACnet standard defines a controller or other BACnet device by describing the key characteristics of objects and properties within the device. The standard requires that objects and properties respond the same way regardless of the manufacturer. When connected to the same BACnet internetwork, devices from various manufactures can share data because of this standardized operation.

- *BACnet objects and properties* on page 291
- *BACnet services* on page 292
- *Priority arrays* on page 292
- *Configuring devices for BACnet alarms* on page 105
- *Reference to BACnet objects* on page 315

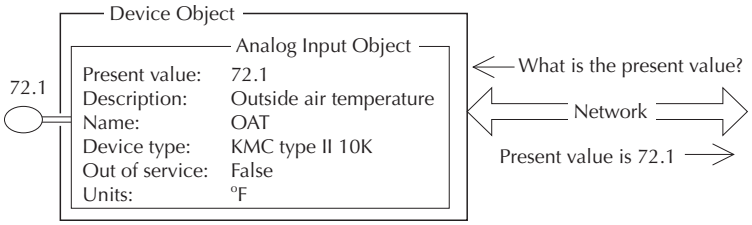
BACnet objects and properties

Objects are the means by which a BACnet device represents information that can be observed or changed. An object may represent a physical point such as an input or output or a logical grouping of data such as a PID loop, schedule or variable. Objects are a method of organizing and accessing data in a way that corresponds to real-world inputs and values.

An example BACnet object

The example object in the illustration *Object model* on page 291.

Illustration 28–1 Object model



BACnet properties

The BACnet standard strictly defines available objects, their properties, and the acceptable values for each property. Because each type of object has the same set of required properties, and the properties follow the same rules about what values they can be, the data that the property represents is generally accessible to any BACnet process that requests it.

BACnet services

BACnet devices use services to acquire information from another device, command another device to perform certain actions, or announces to one or more devices that some event has taken place. Examples of services include scheduled commands and alarms between BACnet devices. Some services read or write properties of objects in the receiving device. Other services convey notification of alarms or other special events, still others read and write files. The services provided by a BACnet device are described by the device's PIC. statement.

In the object model shown in the illustration ***Object model*** on page 291 , a read property service is shown as the question “What is the present value?” The analog input object responds with “Present value is 72.1”.

Priority arrays

BACnet devices use the priority array to control the Present Value property in output and value objects. The priority array property maintains order when several commands are simultaneously issued to change the Present Value property. For example, an operator may enter a command to stop a fan when a schedule is commanding it to run. By programming the command from the operator at a higher priority, the priority array property permits the operator command to take precedence over the schedule.

Priority array properties have 16 levels associated with them. Priority 1 is the highest; priority 16 is the lowest. When a command is issued for a present value property of an output or value object, rather than directly affecting the present value, the object stores the value in its priority array property at the appropriate priority level. The command with the highest priority sets the present value of the object.

An example priority array

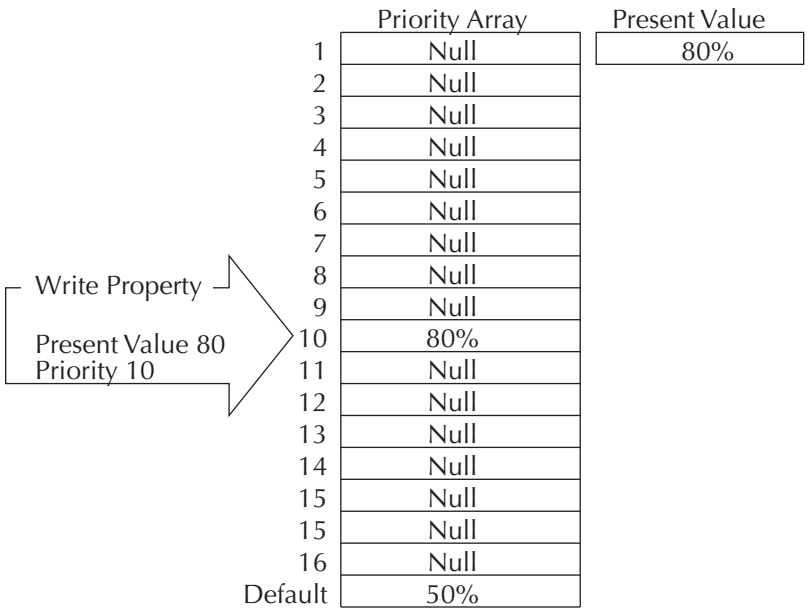
Initially, all levels of a priority array are filled with a ***NULL*** value. The default value is entered in Relinquish Default in the object menu. In this example Relinquish Default equals 50%.

Illustration 28–2 Initial state

	Priority Array	Present Value
1	Null	50%
2	Null	
3	Null	
4	Null	
5	Null	
6	Null	
7	Null	
8	Null	
9	Null	
10	Null	
11	Null	
12	Null	
13	Null	
14	Null	
15	Null	
15	Null	
16	Null	
Default	50%	

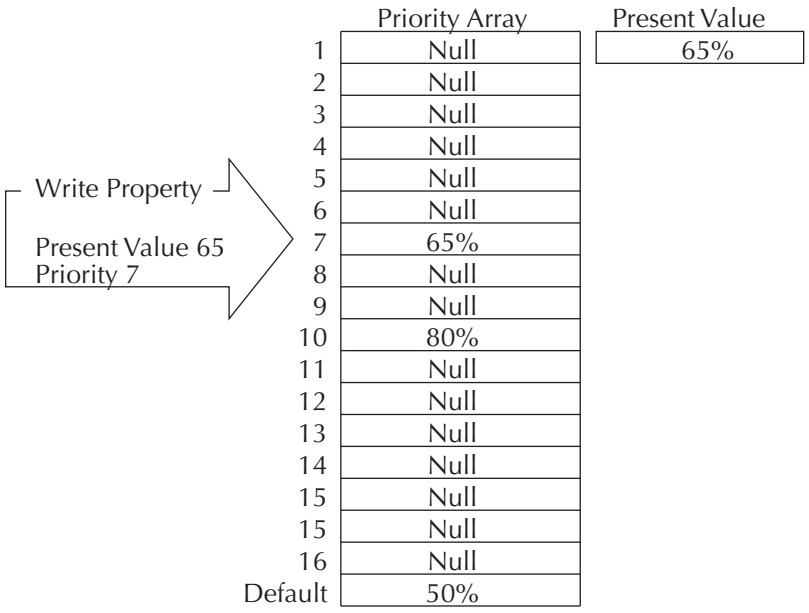
Then, a write property command with a value of 80% at a priority level of 10 is sent to the object. Because this new priority is a higher priority than the default level, the present value changes to 80%. Note that the array default value remains in the array.

Illustration 28–3 First write property command



Next, a write property command with a value of 65% at a priority level 7 is sent to the object. Since this new command has a higher priority than the previous level 10 priority, the present value becomes 65%. As before, the previous values remain in the priority array.

Illustration 28–4 Second write property command



At this point, if a write property command is sent to the object with a ***NULL*** value at priority 7, this relinquishes the priority 7 control at that priority. The output reverts to the next highest priority, which in this example is the 80% value at level 10.

The same principles for controlling analog objects hold true for binary objects, the only difference is that the values for binary objects are ***INACTIVE (0)*** or ***ACTIVE (1)*** and are referred to as numerical values.

To manipulate the write priority of an object with Control Basic, see the keyword ***RLQ*** on page 218.

Standard BACnet priority levels

Some priorities are designated by the BACnet standard. For example, Priority 1 is reserved for use by life/safety systems and Priority 8 is reserved for manual operator commands. The intent of standardizing the meanings of various priority levels encourages a consistent application of those priorities by various vendors across a multitude of facility types where the objectives of the programming cannot be predicted in advance.

Table 28-1 BACnet standard priorities

Priority Level	BACnet Standard Priority
P1	Manual-Life Safety
P2	Automatic-Life Safety
P3	
P4	
P5	Critical Equipment Control
P6	Reserved for minimum on and off time

Table 28-1 BACnet standard priorities (continued)

Priority Level	BACnet Standard Priority
P7	
P8	Manual Operator
P9	Default Control Basic
P10–P16	

Even with these designations, the standard meanings are subject to interpretation and the result of a local decision as to how priority levels are applied. The assignment of specific meanings to the priorities are site dependent and represents the objectives of the building's owner and management. To maintain interoperability, system programmers should apply priorities consistently across all controllers within the same facility.

Related topics

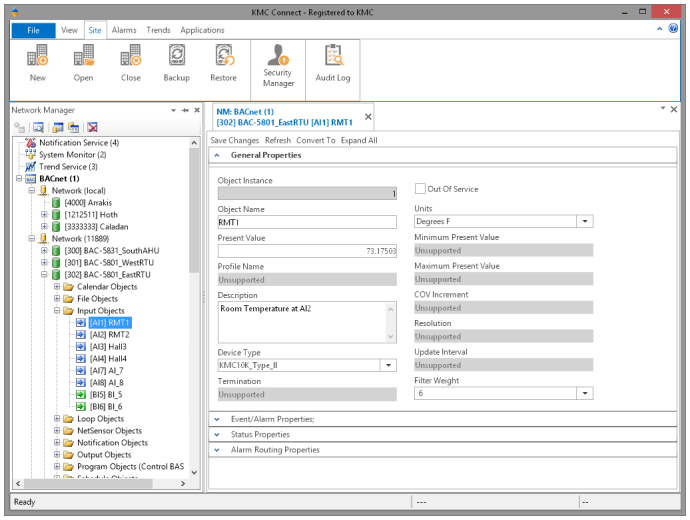
- *BACnet objects and properties* on page 291
- *BACnet services* on page 292

Section 29: Configuring and connecting to BACnet devices and networks

The topics in this section cover connecting to well as configuring BACnet objects and the BACnet service.

The following topics are basic procedures for working with BACnet networks, devices and objects in the Network Manager and configuration pages.

Illustration 29–1 Network Manager and Input Configuration tab



- *Adding BACnet devices to the Network Manager list* on page 298
- *Configuring BACnet devices and objects* on page 300
- *Regenerate networks and devices* on page 304
- *Restart a BACnet device* on page 306
- *Setting BACnet system time* on page 307
- *Automatic MAC Addressing commands* on page 308

Adding BACnet devices to the Network Manager list

To add BACnet devices and networks to the Network Manger list use the **Find Devices** command.

- Use **Find Devices** rather than **Regenerate Networks** to initially populate the Network Manager list.
- When new devices or networks are added, select only the device or network to add to the Network Manager list. Choosing the **All Devices** or **All Networks** options may take several minutes and will unnecessarily increase network traffic.

To add BACnet devices to the Network Manager list, do the following:

1. From the Network Manager list, open the Find Devices dialog box by doing one of the following:
   - On the Network Manager toolbar, click the Find Device icon .
   - In the Network Manager list, right-click a BACnet service icon or network icon and then choose **Find Devices** from the shortcut menu.
2. Choose an option from the **Network** and **Devices** properties.
   - From **Network** choose a specific network, the local network or all of the networks.
   - From **Devices** choose **All Devices**, a specific device or a range of BACnet devices specified by the BACnet instance numbers.
3. Click **Find**. The Network Manager list will fill with BACnet networks and devices. See the table *Network Manager BACnet device icons* on page 298 for a description of the BACnet icons.

Table 29-1 Network Manager BACnet device icons

Icon	Status	Action
	All objects in the device are known to KMC Connect.	No action required. The device is operating normally.
	A BACnet device that has one or more GPL compliant programs.	No action required. The device is operating normally.
	The device has been discovered under limited discovery.	No action required. The device is operating normally.
	The device has been detected but not all objects have been placed in the database.	If the device icon does not change to green, an exception for a port in a firewall may be required.

Table 29-1 Network Manager BACnet device icons (continued)

Icon	Status	Action
	KMC Connect has located the device and is in the process of detecting objects.	No action required. This icon will change to green once all objects are detected.
	The device is known to KMC Connect but is no longer available.	The device may be disconnected, device power is turned off or there is a network problem. Use Regenerate Networks when the problem is corrected.
	Changes have been made to this device that requires a cold start.	See *Restart a BACnet device* on page 306.

Related topics

- *Configuring BACnet devices and objects* on page 300
- *Configuring a BACnet service* on page 311
- *Regenerate networks and devices* on page 304
- *Restart a BACnet device* on page 306
- *Setting BACnet system time* on page 307
- *Automatic MAC Addressing commands* on page 308
- *Using Network Manager* on page 37
- *Network Manager toolbar commands* on page 38
- *Opening and closing the Network Manager pane* on page 38
- *Reference to BACnet objects* on page 315

Configuring BACnet devices and objects

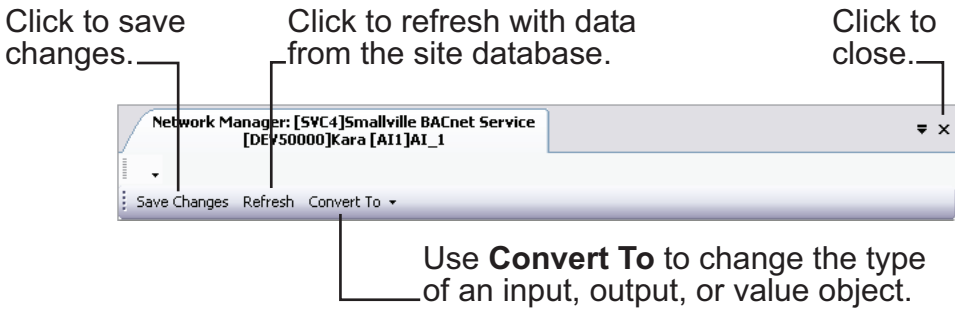
To configure a BACnet device or an object, the device must be in the Network Manager list. If a device is not in the Network manager list, see *Adding BACnet devices to the Network Manager list* on page 298. For devices in the Network Manager list you may use either of the following methods to configure a device or object.

- By opening, changing and saving the configuration page for the device or object.
- By copying a configuration from a backup file.
- By copying the configuration from a similar device or object that is in the Network Manager list.

To open a configuration tab for a BACnet device or object, do the following:

1. In the Network Manager list, click ⊞ or ⊟ to expand or collapse the list to locate a specific BACnet device or object.
   - The device icon will always be in the list under a BACnet service icon and network icon .
   - A BACnet object is in a folder under a device. Each type of object has a unique icon.
2. Once the device or object is located, double-click the object or device icon to open a configuration page in the workspace.
3. Make changes in the configuration page.
   - When complete, click **Save Changes** at the top of the configuration page.
   - Click **Refresh** to discard changes and reload data from the system into the configuration page.
   - To close the configuration page, right-click the tab and choose **Close** or click the close button ✕ in the upper right corner of the page.

Illustration 29–2 Tab in configuration page



Copying configurations

The configuration of a device or object may be copied from either of two sources:

- Directly from another connected device or object in the Network Manager list.
- From a backup file in the Resource Manager list. For details on copying a configuration from a backup file, see *Configuring BACnet devices and objects* on page 300.

To copy a configuration from an item in the Network Manager list, do the following:

1. In the Network Manager list, click ⊞ or ⊟, to expand or collapse the network list to locate a specific device, object or point.
2. Hold ALT and drag the configuration to the object, folder of objects or device.

When dragging a configuration from the Network Manager list, the following actions are permitted:

- Drag an object configuration to a single ***object*** of similar type. For example, drag only input object configurations onto input objects.
- Drag an object configuration to a device icon that contains objects of similar type. A dialog opens with which you can select the specific objects to restore.
- Drag an object to a network icon that includes devices with similar objects.
- Drag a group of objects—such as a folder of input objects—onto a similar group. The objects are matched by their object instance number to the objects in the Network Manager list.

Related topics

- *Adding BACnet devices to the Network Manager list* on page 298
- *Configuring BACnet devices and objects* on page 300
- *Regenerate networks and devices* on page 304
- *Restart a BACnet device* on page 306
- *Setting BACnet system time* on page 307
- *Automatic MAC Addressing commands* on page 308
-

- *Using Network Manager* on page 37
- *Network Manager toolbar commands* on page 38
- *Opening and closing the Network Manager pane* on page 38
- *Reference to BACnet objects* on page 315
- *Input object wizards* on page 441

Creating and deleting BACnet objects

Some BACnet devices support adding and deleting objects. The rules controlling the addition or deletion of objects will vary with the type of device being modified.

- Creating or deleting an input or output object may require restarting the controller.
- Not all devices support all types of objects. Refer to the specifications supplied with the device.
- Only one type of object can be associated with a physical input or output.
- The input or output object instance number will always correspond to the physical input number.
- Some controllers, BAC-5800 and BAC-7000 series controllers, change input and output types from the configuration page. See the topic *Converting input and output objects* on page 303.

Tip: Use these procedures to add input and output objects to the CAN-A168EIO or CAN-5901 IO Expansion modules.

Creating objects

1. In the Network Manager, click ⊞ or ⊟ to expand or collapse the list to locate a BACnet device icon .
2. Click ⊞ to expand the device and locate the icon for a group of objects.
3. Right-click the folder and choose **Create Objects** from the shortcut menu. The **Create Object(s)** dialog opens.
4. Enter the objects to create.
5. When ready, click **OK**.
6. If prompted, restart the controller.

Deleting objects

1. In the Network Manager, click ⊞ or ⊟ to expand or collapse the list to locate a BACnet device icon .
2. Click ⊞ to expand the device and locate the icon for a group of objects.
3. Once the group folder is located, do one of the following:
   - Right-click the group icon and enter the objects to delete in the **Delete Object(s)** dialog.
   - Right-click an individual object and choose **Delete Object(s)** from the shortcut menu.
4. When ready, click **OK**.
5. If prompted, restart the controller.

Related topics

- *Adding BACnet devices to the Network Manager list* on page 298
- *Regenerate networks and devices* on page 304
- *Restart a BACnet device* on page 306
- *Setting BACnet system time* on page 307
- *Automatic MAC Addressing commands* on page 308
- *Input object wizards* on page 441
- *Using Network Manager* on page 37
- *Reference to BACnet objects* on page 315

Converting input and output objects

Depending upon the model of the device, physical device inputs and outputs can be configured as different types of BACnet objects by using one of the following methods.

- By right-clicking the object in the Network Manager list.
- By choosing a new object type in the object configuration page.
- By adding or removing objects as describe in the topic *Creating and deleting BACnet objects* on page 302.

To convert input and output types, use one of the following procedures.

Convert an object from the Network Manager list

1. In the Network Manager list, click ⊞ or ⊟ to expand or collapse the list of devices and objects to locate the input or output object icon.
2. Right-click the icon and choose a new object type from the **Convert to** short-cut menu.
3. When prompted, reinitialize the controller.

Convert object in the object configuration page

1. In the Network Manager list, click ⊞ or ⊟ to expand or collapse the list of devices and objects to locate the input or output object icon.
2. Double-click the object icon to open a configuration page in the workspace.
3. At the top of the page, choose **Convert To** and then choose the type of object.
4. When prompted, reinitialize the controller.

Related topics

- *Adding BACnet devices to the Network Manager list* on page 298
- *Regenerate networks and devices* on page 304
- *Restart a BACnet device* on page 306
- *Setting BACnet system time* on page 307
- *Input object wizards* on page 441

- *Automatic MAC Addressing commands* on page 308
- *Input object wizards* on page 441
- *Using Network Manager* on page 37
- *Reference to BACnet objects* on page 315

Regenerate networks and devices

Use the regenerate commands to refresh the database with device and network data.

- After choosing a regenerate command networks and devices are not available until regeneration is complete.
- Choosing a regenerate command may—depending upon the size of the network—substantially increase traffic on the building automation system network. Use a regenerate command at the lowest level possible on the Network Manager list.

To regenerate a single device

1. Expand the Network Manager list to locate the device icon .
2. Right-click the device and choose **Regenerate Device** from the shortcut menu.
3. The device icons will change color as KMC Connect updates the information for the devices. See the table *Network Manager BACnet device icons* on page 298 for a description of icons.

To regenerate a single network

1. Expand the Network Manager list to locate the network icon .
2. Right-click the network and choose **Regenerate Network** from the shortcut menu.
3. The device icons in the network will change color as KMC Connect updates the information for the devices. See the table *Network Manager BACnet device icons* on page 298 for a description of icons.

To regenerate all networks

1. In the Network Manager list right-click a BACnet service icon and then choose **Regenerate All Networks** from the shortcut menu.
2. The device icons will change color asKMC Connect refreshes the information for the devices. See the table *Network Manager BACnet device icons* on page 298 for a

description of icons.

Related topics

- *Adding BACnet devices to the Network Manager list* on page 298
- *Configuring BACnet devices and objects* on page 300
- *Configuring a BACnet service* on page 311
- *Restart a BACnet device* on page 306
- *Setting BACnet system time* on page 307
- *Automatic MAC Addressing commands* on page 308
- *Using Network Manager* on page 37
- *Network Manager toolbar commands* on page 38
- *Opening and closing the Network Manager pane* on page 38
- *Reference to BACnet objects* on page 315

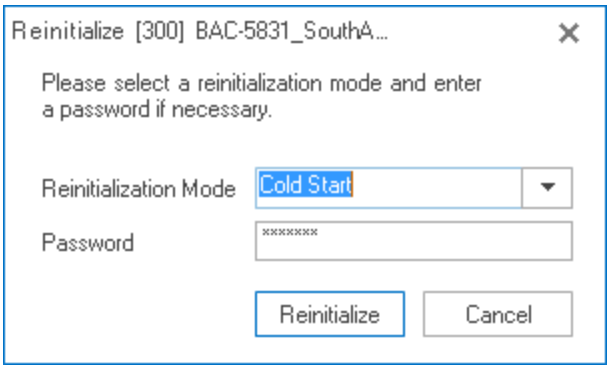
Restart a BACnet device

Use the Reinitialize Device command to perform either a BACnet warm start or cold start in a BACnet device. Changes made to the device are not effective until the device undergoes a warm start, an update notification, a cold start or the device power is cycled.

To restart a BACnet device from KMC Connect do the following:

1. In the Network Manager list, click ⊞ or ⊟ to expand or collapse the list to locate a specific device.
2. Once the device is located, right-click the device icon and choose **Reinitialize Device** from the shortcut menu.
3. Choose either **Warm Start** or **Cold Start**.
4. If required, enter the password and then click **Reinitialize**.

Illustration 29–3 Reinitialize BACnet device dialog



Properties of the Reinitialize dialog

Warm Start Restarts the processor in the controller. In KMC BACnet controllers, all Control Basic programs suspend operation and present values are held at their condition prior to the warm start. Devices from other manufactures restart differently.

Cold Start Restarts the processor in the controller and sets it to its power-up state. In KMC BACnet controllers, all outputs and values are set to default levels until Control Basic programs return the outputs to operational levels. Devices from other manufactures restart differently.

Password If required, enter the password supplied from the manufacturer of the device. For KMC BACnet controllers the password is ***snowman***.

Setting BACnet system time

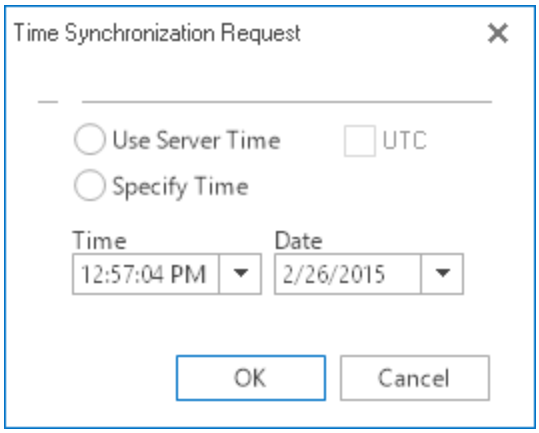
Use the BACnet Time Synchronization dialog to immediately set the time in a single device, all of the devices on a single BACnet network or all devices on all of the networks connected to a BACnet service.

1. Expand or collapse the Network Manager list to reveal the service, network or device.
2. Right-click the service, network or device and choose **Time Synchronization** from the shortcut menu.
3. Do either of the following:
   - To set the time in the item to the time maintained in the computer hosting the site database, click **Use Server Time** and then click **OK**.
   - To set a time and date different than the time and date in the computer hosting the site database, click **Specify Time**, enter time and date values and then click **OK**.
   - To send the time in UTC, click the UTC check box.

Note: UTC (Universal Coordinated Time) is calculated from the time and time zone settings in the computer on which the server is running.

See *Configuring a BACnet service* on page 311 to set up the BACnet service to automatically update system time.

Illustration 29–4 Time Synchronization dialog



Related topics

- *Regenerate networks and devices* on page 304
- *Adding BACnet devices to the Network Manager list* on page 298
- *Configuring BACnet devices and objects* on page 300
- *Regenerate networks and devices* on page 304
- *Restart a BACnet device* on page 306
- *Automatic MAC Addressing commands* on page 308
- *Using Network Manager* on page 37

- *Network Manager toolbar commands* on page 38
- *Opening and closing the Network Manager pane* on page 38
- *Reference to BACnet objects* on page 315

Automatic MAC Addressing commands

Use the Auto MAC Addressing commands to control and manage MS/TP automatic MAC addressing in some BACnet controllers from KMC Controls. Automatic addressing is available on controllers with firmware version 2.0.0.0 and later. Using MS/TP automatic addressing is fully described in the manual *Installation Instructions for MS/TP Automatic MAC Addressing* available from the KMC Controls web site.

To use the Automatic MAC Addressing commands do the following:

1. In the Network Manager list, click ⊞ or ⊟ to expand or collapse the list to locate a specific device with automatic MAC addressing.
2. Right-click the device.
3. Once the device is located, right-click the device icon and choose **Automatic MAC Addressing** and then on of the following commands from the shortcut menu.

Request Status Displays the status of automatic addressing in the controller in the output window.

Restart Anchor (Anchor controller only) Restarts the auto addressing process without sending a warm start or cold start command to the anchor controller.

Lock MAC Addresses (Anchor controller only) Locks the temporary MAC addresses in nomad controllers and makes the MAC addresses permanent.

Scanning BACnet configuration files

The Scan Devices for Uninitialized Configuration Files command checks BACnet controllers from KMC Controls for valid BACnet configuration files. If uninitialized files are detected, the operator can choose to initialize the files. A controller with uninitialized files will not retain configuration changes after controller power is cycled.

To scan for uninitialized files, do the following:

1. Open Network Manager.
2. Locate and select a BACnet service or network to scan.
3. Right-click the service or network icon and then choose **Scan Devices for Uninitialized Configuration Files** from the shortcut menu.
4. If devices with uninitialized files are found, do one of the following:
   - Click **Yes** to initialize files. This will take several minutes per controller.
   - Click **No** to cancel. This will mark each controller with uninitialized files as unavailable.

Backing up and restoring BACnet devices

There are two methods that Design Studio supports for backing up a BACnet device or controller.

- *Importing and exporting .bnd files* on page 310 saves a file that can be edited in Resource Manager.
- *BACnet Backup and restore* on page 309 is a standard BACnet method for archiving device configurations from other manufacturers.

BACnet Backup and restore

BACnet Backup is a standard BACnet procedure for saving the configuration from devices that support this feature. It is a method to archive a device configuration that includes proprietary objects or properties without using a manufacturers proprietary software. In KMC Connect, the BACnet backup is saved as a `.bbk` file. The `.bbk` file can be then be stored on a computer drive and used to restore the saved configuration at a later date.

The `.bbk` file is unique to Design Studio and cannot be used by programs from other manufacturers. It cannot be edited by Design Studio.

Preparation Time

The Preparation Time is a delay built into Design Studio that gives a device time to prepare to send the backup files. It is not used in devices that support the Backup Preparation Time property. If a device reaches a time limit when starting a backup procedure, increasing this value may result in a successful backup.

Password

For KMC BACnet controllers, the password is ***snowman***. Other manufacturers use different passwords.

To save a configuration to a `.bbk` file, do the following:

1. Open Network Manager.
2. Locate and select a BACnet device, network, or service to backup in the Network Manager list.
3. Right-click the icon and choose **Export** from the shortcut menu.
4. Select a location to store the backup file. The default location is the BACbetBackup folder in the Resource Manager.
5. Click **Okay** to start the export.

To restore a device from a `.bbk` file, do the following:

1. Locate the device in the Network Manager list.
2. Right-click the device icon and choose **Import** from the shortcut menu.
3. Locate the `.bbk` file and then click **Open**.
4. If required, enter a password.

5 Click **OK** start the import. Once the correct password is entered, the import begins.

Importing and exporting .bnd files

Use the Network Manager Import and Export commands to backup or restore a BACnet controller configuration from the Network Manager list.

Exporting .bnd files

To export a backup `.bnd` file, do the following:

1 Open Network Manager.

2 Locate and select a BACnet device, network, or service to backup in the Network Manager list.

3 Do one of the following:

- Right-click a device icon and choose **Export** from the shortcut menu.
- Right-click a network or service icon and choose **Backup Devices** from the shortcut menu.

4 Select a location to store the backup file.

5 Click **Okay** to start the export.

Importing .bnd files

To import a backup `.bnd` file, do the following:

1 Locate the device in the Network Manager list.

2 Right-click the device icon and choose **Import** from the shortcut menu.

3 Locate the `.bnd` file and then click **Open**.

4 When the Restore dialog opens, choose the objects and properties to restore.

5 Click **OK** start the import.

Related topics

- *Importing and exporting BACstage .bac files* on page 310
- *Using Network Manager* on page 37
- *Creating backup files* on page 51

Importing and exporting BACstage .bac files

Use the Resource Manager to import and export controller configuration files that are not native KMC Connect file formats.

To import a BACstage `.bac` file

1. Use Windows Explorer to move a `.bac` file to the Resource Manager folder.
2. In Design Studio locate the `.bac` file in the Resource Manager list.
3. Use the objects in the `.bac` file as you would use the objects in a `.bnd` file file. For example, you may drag an input object from the `.bnd` file to a BACnet input object in the Network Manager list.

To export a BACstage `.bac` file

1. Locate the `.bnd` file in the Resource Manager list.
2. Right-click the file and then choose **Save As** from the shortcut menu.
3. Select the file type and then click **OK**.

Related topics

- *About Resource Manager* on page 45
- *Opening and closing the Resource Manager pane* on page 47
- *Editing items in the Resource Manager list* on page 47
- *Restoring a configuration* on page 48
- *Creating backup files* on page 51
- *Managing the Resource Manager list* on page 51
- *Adding files to the Resource Manager folder* on page 52
- *Backing up and restoring BACnet devices* on page 309

Configuring a BACnet service

The BACnet service is the link between a BACnet building automation system and other KMC Connect services. Use the BACnet Service configuration tab to view or change parameters of the BACnet service .

To open the configuration tab, do the following:

1. Locate the BACnet service icon  in the Network Manager list.
2. Right-click the BACnet service icon and choose **Configure Service** from the shortcut menu.
3. Make the changes to the service.
4. When changes are complete, click **Save Changes to Service** at the top of the configuration page.

General Properties

Items under General Properties are usually set up when KMC Connect is installed.

Service Name A descriptive label of the service. This property must be unique among all BACnet devices on the internetwork. The set of characters used in Service Name is restricted to printable characters. Service Name is a BACnet property that is accessible to other BACnet devices and workstations.

Service ID Service ID is assigned by KMC Connect and cannot be changed.

Service Host Address Not used by KMC Connect.

Service Host TCP Port Not used by KMC Connect.

Settings and Statistics

Server BACnet Device Instance A number that uniquely identifies the BAnet service on the internetwork. The device instance number is assigned by the BACnet system designer. Valid instance number's range from 0 to 4,194,303. It is by reference to the device instance number that data is exchanged between BACnet devices.

When a site is first set up, Server BACnet Device Instance is assigned a random value.

Maximum Requests Per Network Sets the maximum number of unanswered requests sent to a BACnet MS/TP network. Once the maximum number of requests is reached, the service stores requests in a buffer. As a request is answered, the service sends to the network the next request in the buffer.

The KMC Connect default entry is 20.

Maximum Response Staleness To reduce redundant network traffic, the BACnet service caches data that is likely to be requested by other KMC Connect services. If a request for the same data from a building automation is received by the service and the data has been in the cache less than the period set by ***Maximum Response Staleness,*** the data in the cache is used. If the data has been in the cache longer than the period set by ***Maximum Response Staleness,*** new data is retrieved from the building system.

Status Request Heartbeat Interval Sets an interval for maintaining communication with each controller. If the BACnet service does not receive some type communication from each controller in the Network Manager list within the heartbeat interval, it will attempt to contact the controller.

The default value is two minutes.

Time Synchronization Interval KMC Connect can be set to function as a BACnet time master device. The value in Time Synchronization Interval sets the interval at which the BACnet service will send a time message. All controllers on the internetwork can then synchronize their internal clocks to the KMC Connect time message. KMC Connect uses computer time for the time message. The interval can be set to hourly, daily, weekly, or monthly. To disable the time message, choose Never. To send a single time update, see *Setting BACnet system time* on page 307.

Monitor Statistics Select to enable statistic gathering for diagnostics.

Incoming Requests Not applicable to KMC Connect.

Pending Requests Not applicable to KMC Connect.

Completed Requests Not applicable to KMC Connect.

Licensing

License Status KMC Connect displays the license information contained in the license key.

License Address and Port The address and UDP port of the computer with the license key.

COV Subscriptions

Use the COV subscription list to view or delete COV subscriptions from an object.

To delete COV subscriptions, do the following:

- Scroll through the list to locate the object and COV subscription.
- Right-click the subscription and then choose either **Unsubscribe COV** or **Unsubscribe All COVs**.
- When changes are complete, click **Save Changes to Service** at the top of the configuration page.

See also the related topic *Unsubscribing from COV subscriptions* on page 123.

Related topics

- *Opening and closing the Network Manager pane* on page 38
- *Adding BACnet devices to the Network Manager list* on page 298

Configuring the IP connection

Conquest series controllers that include an Ethernet port may require configuration for proper network operation.

- Ethernet is the default configuration for Conquest controllers. No additional configuration is required.
- To connect the controller as an IP device, change the address and configuration as described in this topic. IP network configuration properties should be part of the system plans and must be coordinated with the IT department.

Tip: These properties can also be changed with the KMC Connect Lite mobile app.

To change the addressing and configuration, do the following:

1. Locate the device object  in the Network Manager list and open it .
2. Expand the IP Configuration Properties section.
3. Make changes to the settings and click **Save** at the top of the tab.
4. Click **Refresh**.
5. Verify that the settings are valid.
   - If IP Configuration Good is True then proceed.
   - If IP Configuration Good is False, check IP Configuration Status for possible causes.

6 From the list in IP Configuration Command choose **Commit**.

7 Click **Save** at the top.

8 Restart the device. See the topic *Restart a BACnet device* on page 306.

Comm Type Choose from either IP or Ethernet.

- Ethernet requires no further configuration.
- IP requires additional configuration for the address, subnet mask, default gateway, and UDP Port.

IP Address The network address of the controller. The address must be unique on the IT network and is supplied or coordinated with the IT department.

Default Gateway The address of the network gateway router. The address of the gateway is supplied by the IT department.

Subnet Mask Subnet Mask Mask determines which part of the IP address is used for a network identifier and which part is used for a device identifier. The mask must match the mask for the network gateway router and other devices on the subnet.

UDP Port Must match the UDP port of the BACnet network to which it is connected. Port numbers are designated by BACnet routers. The default port number is 47808.

BACnet IP Mode Controller supports normal IP and Foreign device connections to a BBMD (BACnet Broadcast Management Device.

- Normal—Only the IP Address, Default Gateway, and Subnet mask properties are required.
- Foreign device—The BBMD Address and BBMD Port must also be configured.

BBMD Address Enter the address of the remote BBMD. If network address translation (NAT) is used between the computer and the BBMD, contact the network system administrator for the correct public IP address.

BBMD Port Enter the port number of the remote BBMD. If port address translation (PAT) is used between the computer and the BBMD, contact the network system administrator for the correct public UDP port.

IP Configuration Good If the address information is valid, this will display *TRUE*.

IP Configuration Status Displays *OKAY* if the address information is valid; lists probable cause if the address information is not valid.

IP Configuration Command Use to commit or discard pending changes. Choose Commit or Discard and then choose **Save**.

- Idle—Waiting on next command.
- Commit—Save the pending changes in the device.
- Discard—Discard pending changes.

Section 30: Reference to BACnet objects

The following topics are a reference to the features and characteristics of BACnet objects and properties.

KMC Connect supports standard BACnet objects and properties within BACnet devices.

- To locate a BACnet device on the internetwork, see *Using Network Manager* on page 37.
- To configure the objects and properties in a BACnet device see *Configuring BACnet devices and objects* on page 300.
- For an introduction to BACnet see *BACnet overview* on page 291.

This section describes the following BACnet objects:

- *Access Door object* on page 316
- *Averaging objects* on page 320
- *Calendar object* on page 321
- *Command object* on page 322
- *Device objects* on page 323
- *Event enrollment object* on page 329
- *Event Log objects* on page 333
- *File objects* on page 336
- *Group objects* on page 336
- *Input object–accumulator* on page 336
- *Input object–analog* on page 340
- *Input object–binary* on page 345
- *Input objects–multistate* on page 348
- *Life Safety Point object* on page 351
- *Life Safety Zone object* on page 354
- *Loop objects* on page 361
- *Notification object* on page 365
- *Output objects–analog* on page 366
- *Output objects–binary* on page 370
- *Output object–multistate* on page 373
- *Schedule object* on page 379
- *Trend Log object* on page 384
- *Trend Log Multiple objects* on page 388
- *Value objects–analog* on page 392
- *Value objects–binary* on page 396
- *Value objects–multistate* on page 399

Access Door object

The Access Door object represents the physical characteristics of an access–controlled door and its associated physical hardware and devices. Typically this includes door contacts, door locks, and card readers or other-request-for-access devices.

See the related topic *BACnet objects and properties* on page 291.

General Properties

Object Instance The Access Door object number. The objects are numbered sequentially within a BACnet device.

Object Name A text label of the object. The exact length will depend upon the controller that contains the object. The Name property must be unique within the BACnet device that maintains it. The set of characters entered for the Name property must be printable characters.

Present Value The present value of an Access Door objects represents the current, active command of the object. The command may be one of the following standard BACnet commands for this object type.

- LOCK
- UNLOCK
- PULSE_UNLOCK
- EXTENDED_PULSE_UNLOCK

Write Priority When saving the object, this property sets priority for the Present Value property. See the topic *Priority arrays* on page 292.

Profile Name The name the vendor's object profile to which this object conforms. A profile name defines a set of additional properties, behaviors and/or requirements for this object. The definition of the profile is vendor unique and not part of the BACnet standard.

Description A text label of the object. The exact length of the Description property will depend upon the controller that contains the object.

Door Status This property represents the opened or closed status of the door. The standard BACnet states for this property are:

- CLOSED
- OPEN
- UNKNOWN

Lock Status The Lock Status property represents the monitored status of the physical door lock. The standard BACnet states for this property are:

- LOCKED
- UNLOCKED
- FAULT
- UNKNOWN

Secured Status This property indicates if the door is in a SECURED, UNSECURED, or UNKNOWN status. The door is considered secure only if all of the following conditions are met:

- There are no alarm status flags
- The Masked Alarms Values list is empty
- The Door Status property is CLOSED
- The Present Value property is LOCKED
- The Locked Status property is LOCKED or UNKNOWN

Out of Service When the Out Of Service check box is selected, the object is not tracking changes to the physical door.

Relinquish Default Sets the status or value that will take effect when all levels of the priority array are NULL. See *Priority arrays* on page 292.

Door Members This property is a list of input or output devices, authentication devices, schedules, programs, or other objects that are associated with the physical door. The objects in the list and they way they interact with the door depends on the controller that contains the object.

To add an object to the list do either of the following:

- Locate the object in the Network Manager list and drag it to the list.
- Choose the object number from the Object ID list

To remove an object from list click the remove button  .

Door Pulse Time Indicates the maximum time—in tenths of seconds—that the door will remain unlocked when the Present Value property is set to PULSE_UNLOCK.

Door Extended Pulse Time Indicates the maximum time—in tenths of seconds—that the door will remain unlocked when the Present Value property is set to EXTENDED_PULSE_UNLOCK.

Door Unlock Delay Time Indicates the maximum time—in tenths of seconds—that the door will delay unlocking when the Present Value property is set to PULSE_UNLOCK or EXTENDED_PULSE_UNLOCK.

Maintenance Required An indication of required maintenance for the door device that is represented by the Access Door object. The exact method for determining that maintenance is required depends upon the controller that contains the object.

Event/Alarm Properties

Related topics

- *Configuring BACnet devices for intrinsic alarms* on page 111
- *About intrinsic BACnet alarms and events* on page 108*About intrinsic BACnet alarms and events* on page 108
- *Notification object* on page 365

Notification Class Specifies the notification class object to be used when handling and generating event notifications for this object. See *Notification object* on page 365 for details about the notification class object.

Notification Type This property specifies whether the notifications generated by the object are events or alarms. Alarm and event notifications are handled differently by the device—usually a workstation—that receives the notification.

Time Delay The Time Delay property defines a minimum period, in seconds, for a set of conditions to exist before a *TO OFF NORMAL* or *TO NORMAL* event occurs. Use Time Delay with the Alarm Value property to define conditions that indicate Present Value has changed states.

Alarm Values When the state of the Door Alarm State property matches any of the items selected in the Alarm Values property list, a notification event is generated.

- Normal
- Alarm
- Door Open Tool Long
- Forced Open
- Tamper
- Door Fault

Fault Values Door Alarm State property equals one of the selected items in the Alarm Values property list and ToFault is selected in Event Enable a fault is generated.

- Normal
- Alarm
- Door Open Tool Long
- Forced Open
- Tamper
- Door Fault

Event Enable Use Event Enable to enable notifications for To Off Normal, To Normal and To Fault conditions.

Door Open Too Long Time This property delays changing the Door Alarm State property by the time entered in the Door Open Too Long property. The time is tenths of seconds.

Door Alarm State This is the current alarm property for the physical door.

Masked Alarm Values A list of alarm states that are not used for the Door Alarm state.

Status Properties

Status Flags The four BACnet status flags are an indication of the general condition of the object.

Table 30-1 Access Door object status flags

Flag	Description
IN ALARM	False (0) if the event state property is Normal, otherwise True (1)
FAULT	True (1) if Reliability is present and the value for Reliability is not NO FAULT DETECTED, otherwise False (0)
OVERRIDDEN	True (1) if the point has been overridden by some mechanism local to the BACnet device. When this flag is True, Present Value and Reliability are no longer tracking changes to the physical input. Otherwise, the flag is False (0).
OUT OF SERVICE	True (1) if Out Of Service is selected. Otherwise False(0).

Reliability This property is an indication that the Present Value property or the operation of the physical input may be not be reliable as defined by the device or an operator. Reliability can have any of the following values:

- NO_FAULT_DETECTED
- MULTISTATE_FAULT
- CONFIGURATION _ERROR
- COMMUNICATION_FAILURE
- UNRELIABLE_OTHER

Event State Use Event State to determine that this object has an active event state associated with it.

- If the object supports intrinsic reporting, then Event State indicates the state of the object.
- If the object does not support intrinsic reporting, then the value of this property is Normal.

Event Time Stamps This property holds the times of the last event notifications of *TO OFF NORMAL, TO FAULT,* and *TO NORMAL* events.

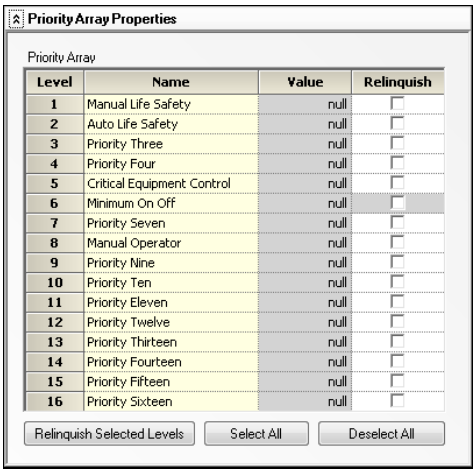
Priority Array Properties

Displays the priority level for the object.

To relinquish a priority value, select the check box and then click **Relinquish**.

See the related topic *Priority arrays* on page 292.

Illustration 30–1 Priority Array Properties



Alarm Routing Properties

Select the KMC Connect alarm class routing from the list.

Averaging objects

An Averaging object represents the results of regularly sampling the value of a property, storing the samples, and then calculating the minimum, maximum and average values of the samples. The object stores the minimum, maximum, and average values of the samples as properties of the object. The sample can be from any boolean, integer, unsigned integer, enumerated, or real property of any object of the BACnet device in which the object is found. The sampled object property may also be found in another BACnet device on the internetwork.

The Averaging object uses a buffer to hold a number of samples. At preset intervals, a new sample is taken which then replaces the oldest sample in the memory. When the new sample is added to the buffer, the minimum, maximum, and average values are recalculated.

To configure the properties in a BACnet device see *Configuring BACnet devices and objects* on page 300.

Object Instance The Averaging object number. The objects are numbered sequentially within a BACnet device.

Object Name A text label of the object. The exact length will depend upon the controller that contains the object. The Name property must be unique within the BACnet device that maintains it. The set of characters entered for the Name property must be printable characters.

Profile Name The name the vendor's object profile to which this object conforms. A profile name defines a set of additional properties, behaviors and/or requirements for this object. The definition of the profile is vendor unique and not part of the BACnet standard.

Description A text label of the object. The exact length of the Description property will depend upon the controller that contains the object.

Minimum Value This property holds the lowest value of all the samples in the buffer.

Minimum Value Time Stamp This property indicates the date and time at which the valued stored by the Minimum Value property was sampled.

Maximum Value This property holds the highest value of all the samples in the buffer.

Maximum Value Time Stamp This property indicates the date and time at which the valued stored by the Maximum Value property was sampled.

Device, Object and Property Designates the device, object, and property as the source of data for the Averaging object.

- **Device**–an object within a BACnet device on the internetwork. Choose **Local** to select the device within which the Averaging object is located.
- **Device**–an object within a BACnet device on the internetwork. Choose **Local** to select the device within which the Averaging object is located.
- **Property**–Choose the property within the designated object. Typically the property is **Present Value**.

Window Interval The time interval—in seconds—over which the minimum, maximum, and average values are calculated. The time between samples can be calculated by dividing the values in Window Samples by Window Interval.

Window Samples Sets the number of samples to be taken over the period of time entered in the Window Interval property. The minimum value is 15.

Average Value This property holds the average value of all the samples in the buffer.

Variance Value This property holds the variance of all the values in the buffer. The variance is calculated after the most recent sample. After a reset the Variance Value property will equal NAN (Not a Number) until a sample is added to the buffer.

Attempted Samples Indicates the number of samples the Averaging object has attempted to collect over the period set by the Window Interval property. This property can be used with the value in Window Samples to determine if the buffer is full. After a device restart this property will remain at zero until a valid sample is added to the buffer..

Valid Samples Indicates the number of samples that have been successfully collected for the current window.

Calendar object

A calendar object is a standard BACnet object that contains a list of calendar dates. Use the calendar object to enter and manage a list of special dates. These special dates may be holidays, special events or other days that require special attention on a calendar.

- See *Schedule object* on page 379 for more information about setting up schedules.
- To configure the properties in a BACnet device see *Configuring BACnet devices and objects* on page 300.

Object Instance The calendar object number. Calendar objects are numbered sequentially within the KMC BACnet device.

Object Name A text label of the object. The exact length will depend upon the controller that contains the object. The Name property must be unique within the BACnet device that maintains it. The set of characters entered for the Name property must be printable characters.

Present Value This property indicates the current value of the calender object. If the current system date and time is in the calendar entries list, **Present Value** is *True*. If the current date and time do not have a match in the calendar entries list, **Present Value** is *False*.

Profile Name The name the vendor's object profile to which this object conforms. A profile name defines a set of additional properties, behaviors and/or requirements for this object. The definition of the profile is vendor unique and not part of the BACnet standard.

Description A text label of the object. The exact length of the Description property will depend upon the controller that contains the object.

Calendar Entries Each entry in the calendar entries date list can include any of the of the following:

- **Date**—A single date. If the date matches the current system time and date *PRESENT VALUE* of the calendar object is *TRUE*.
- **Date Range**—If the current system date falls with the range of dates specified by *RANGE* then *PRESENT VALUE* of the calender object is *TRUE*. If *END DATE* is empty, than all dates beginning with *START DATE* are considered valid dates in the range of dates. If *START DATE* is empty than all dates from the current system date up to *END DATE* are considered to be valid dates in the range of dates.
- **Week & Day**—If the current system time and date match the day of the week and month, then *PRESENT VALUE* is *True*. If no month is selected, than `Present Value` is *TRUE* on that day of the week for every month of the year.

To add a calendar entry to the list.

Use either of the following methods to add a calendar to the calendar list.

- Click **Add** and then choose **Date**, **Date Range** or **Week and Day** from the shortcut menu.
- Right-click the list and then choose **Date**, **Date Range** or **Week and Day** from the shortcut menu.

To change an entry in the list.

Use any of the following methods to change a calendar entry in the list.

- Right-click an entry and then make a selection from the shortcut menu.
- Select the entry and then click **Edit**.
- Select the entry and then click **Remove**.

Command object

The Command Object defines an object that represents one or more multi-action command procedures. The command procedures are stored in the object as action lists. Writing the list number to the Present Value property commands the object to run the command sequence in the action list.

To configure the properties in a BACnet device see *Configuring BACnet devices and objects* on page 300.

Object Instance The number of the object. Command objects are numbered sequentially within a BACnet device.

Object Name A text label of the object. The exact length will depend upon the controller that contains the object. The Name property must be unique within the BACnet device that maintains it. The set of characters entered for the Name property must be printable characters.

Present Value The Present Value property commands which of the action list the object should run. For example writing the value 5 to Present Value will command the object to run the fifth action list.

Profile Name The name the vendor's object profile to which this object conforms. A profile name defines a set of additional properties, behaviors and/or requirements for this object. The definition of the profile is vendor unique and not part of the BACnet standard.

Description A text label of the object. The exact length of the Description property will depend upon the controller that contains the object.

In Process This property changes to True (1) when a value is written to Present Value. This indicates the object is running the sequence in one of the action lists. The In Process property is then set to False (0) when the sequence is complete.

All Writes Successful Indicates the successful completion of a sequence in an action list.

Action The Action property is a number of lists of command sequences. The object runs the sequence in the list by writing the list number to the Present Value property.

Action Text A text list of descriptions for each of the possible values for the Present Value property.

Device objects

The properties in the device object represent general characteristics of a BACnet controller. In each device there is only one device object. Use the Device Objects configuration page to set the device object properties in a BACnet controller.

To configure the objects and properties in a BACnet device object see the topic *Configuring BACnet devices and objects* on page 300.

General Properties

Device Instance A number that uniquely identifies the device on the internetwork. The device instance number is assigned by the BACnet system designer. Valid instance number's range from 0 to 4,194,302 and must be unique on the BACnet internetwork. It is by reference to the device instance number that data is exchanged between BACnet devices.

Pending Device Instance (For Conquest controllers only) Enter the new Device Instance, save the change, and then reinitialize the controller. Valid instance number's range from 0 to 4,194,302.

Device Network The number of the local network to which the device is connected. The network number is zero (0) if the device is connected to the same network as KMC Connect. The network number is displayed for information only and cannot be changed from KMC Connect.

Device MAC Address The MAC address is a number assigned to the device that indicates the node number on the MS/TP network to which it is connected.

- For devices that connect directly to a LAN, the MAC address is entered by the manufacturer and cannot be changed.
- The address must be unique on the local network but, may be duplicated on other MS/TP networks.
- MAC addresses for MS/TP networks start at 0 and are assigned sequentially.
- Valid MS/TP addresses are 0-127 for master devices.

Pending MAC Address (For Conquest controllers only) Enter the new MAC Address, save the change, and then reinitialize the controller. Valid address number's range from 0 to 127.

Device Name A 63-character label of the device. This property must be unique among all devices on the internetwork. The default entry for Device Name in KMC BACnet controllers is the model number of the controller followed by the serial number.

Location An optional 32-character description of the object's physical location.

Profile Name The name of the vendor's object profile to which this object conforms. A profile name defines a set of additional properties, behaviors and/or requirements for this object. The definition of the profile is vendor unique and not part of the BACnet standard.

Description A text label of the object. The exact length of the Description property will depend upon the controller that contains the object.

Vendor Identifier This is a unique vendor identification code, assigned by ASHRAE, which is used to distinguish proprietary extensions to the protocol.

Vendor Name This property identifies the manufacturer of the BACnet Device.

Model Name The model name property is assigned by the vendor to represent the model of the BACnet device.

Firmware Version Displays the firmware version number stored in the device. Check the KMC Controls website for the current version of firmware. When calling for technical support, have the firmware release number available.

Application Software Version This property identifies the version of application software installed in the device. The content of this property string is a local matter. Typical content of the property could be a date-and-time stamp, a programmer's name or a host file version number.

System Status This property reflects the current physical and logical status of the BACnet device. The System Status property can have any of the following values:

- OPERATIONAL
- OPERATIONAL READ ONLY
- DOWNLOAD REQUIRED
- DOWNLOAD IN PROGRESS
- NON OPERATIONAL

Local Time The Local Time property indicates the time of day as known to the device. If the BACnet device does not have any knowledge of time or date, then the Local Time property is omitted.

Local Date The Local Date property indicates the date as known to the device. If the BACnet device does not have any knowledge of time or date, then the Local Date property is omitted

UTC Offset The UTC Offset property indicates the time offset—in minutes—between local standard time and Universal Time Coordinated. The value of the property ranges from -780 to +780 seconds. The time zones to the west of the zero degree meridian are positive values; those to the east are negative values. The value of the UTC Offset property is subtracted from the UTC received in a UTC Time Synchronization service request to calculate the correct local standard time.

Daylight Saving Status The Daylight Saving Status property indicates True when daylight saving time is in effect and False when it is not in effect at the device's location.

Protocol Version Protocol Version represents the version of the BACnet protocol supported by the BACnet device.

Protocol Revision This property indicates the minor revision level of the BACnet standard.

KMC Hardware Info Additional information about the controller—such as the electronic serial number—that is not in the description or other properties.

Restart Required

IP Configuration Properties

See the topic *Configuring the IP connection* on page 313.

Communication Properties

MAX APDU Size This property is the maximum number of octets that may be contained in a single, indivisible application layer protocol data unit.

APDU Timeout Indicates the period—in milliseconds—between retransmissions of an APDU requiring an acknowledgment for which no acknowledgment has been received.

APDU Retries Indicates the maximum number of retries that an APDU shall be retransmitted.

Segmentation Supported BACnet segmentation indicates whether the device supports segmentation of messages and, if so, whether it supports segmented transmission, segmented reception, or both.

Baud Rate Set Baud Rate to match the speed of the MS/TP network to which the device is connected. All devices on the same network must be set to the same speed.

Conquest MS/TP Baud Rate (Conquest controllers only. See *Conquest compliant controllers* on page 469.) Set Baud Rate to match the speed of the MS/TP network to which the Conquest device is connected. All devices on the same network must be set to the same speed.

APDU Segment Timeout The Segment Timeout property indicates the amount of time—in milliseconds—between retransmission of an APDU segment. The default value for this property is 2000 milliseconds. To maintain reliable communication, set the values of the

Segment Timeout properties of all device objects of all intercommunicating devices to the same value.

Max Master Indicates the highest MAC address the device will attempt to locate while polling for master devices on the local network.

- Setting Max Master to allow an additional five controllers beyond the number of controllers connected to the local network will not significantly decrease response time.
- Setting Max Master significantly higher than the highest numbered device could result in increased polling and slower response times.
- In KMC Connect, Max Master cannot be set lower than the MAC address of the controller.

Caution

Setting Max Master lower than the highest addressed controller will result in controllers that are not polled and data from those controllers that is not shared.

Max Info Frames Sets the maximum number of packets that are sent before passing the token.

Max Segments This property indicates the maximum number of APDU segments accepted by the device.

GEST (Gated Event State Transitions) When selected, prevents the in-alarm bit in the status flags property from indicating an alarm condition when ***Event Enable*** within an object is set to ***Disabled***. This prevents some BACnet operator workstations from detecting an alarm condition when the ***Event Enable*** property is set to ***Disabled***.

When the ***Gated Event State Transactions*** check box is clear, the in-alarm bit indicates an alarm when the present value of an object meets alarm conditions regardless of the ***Event Enable*** property value.

The ***Event Enable*** property is set for alarm or event conditions within input, output, value, loop and trend objects.

Device Database and Restart Properties

Database Revision A number under control of the device's firmware that displays the revision of the device's internal database. The revision number is incremented when an object is created, an object is deleted, the name of an object changes, an object identifier number changes or a restore is performed.

Last Restore Time This is an optional property that holds the time at which the device's firmware image was last restored. This property is supported if the device supports the BACnet backup and restore procedures.

Backup Failure Timeout This is an optional property that indicates the time—in seconds—that the device being backed up or restored must wait before unilaterally ending the backup or restore procedure.

Configuration Files This optional property is a BACnet array that identifies the files within the device that define the device's image that can be backed up. The contents of this property is required to be valid only during the backup procedure. This property must be supported if the device supports the BACnet backup and restore procedure.

Last Restore Reason This optional property indicates the reason for the last device restart. The possible reasons for the restart are listed in the following table.

Table 30-2 Reasons for restart

Reason	Description
UNKNOWN	The device cannot determine the cause of the last restart.
COLDSTART	The device was restarted because of a request to reinitialize the device from an operator workstation or some other coldstart request.
WARMSTART	The device was restarted because of a request to reinitialize the device from an operator workstation or some other warmstart request.
DETECTED POWER LOST	The device detected that incoming power was lost.
DETECTED POWERED OFF	The device detected that its power switch was turned off.
HARDWARE WATCHDOG	The hardware watchdog timer reset the device.
SOFTWARE WATCHDOG	The software watchdog timer reset the device.
SUSPENDED	The device operation was suspended. How or why operation was suspended will vary by the manufacturer of the device.

Time of Device Restart This property holds the time at which the device was last restarted.

Restart Notification Recipients This property is a list of devices or addresses that will receive a notification that the device has restarted.

Time Synchronization and Daylight Saving Time Properties

Daylight Saving Time End Enter the day and time that is the end of Daylight Saving Time. This is property is unique to some BACnet controllers from KMC Controls, Inc.

Daylight Saving Time Start Enter the day and time that is the start of Daylight Saving Time. This is property is unique to some BACnet controllers from KMC Controls, Inc.

Enable Daylight Saving Time Enables the controller to change its time to Day Light Saving time based either on the dates entered in Daylight Saving Time End and Start or to automatically calculate Daylight Saving Time.

Automatically Calculate Daylight Saving Time Enables a rules-based calculation of Daylight Saving Time in selected controllers. The Daylight Saving Time rules are entered directly in the user interface of the controllers and are not available in KMC Connect.

Time Synchronization Interval This property specifies the periodic interval at which time and UTC synchronization request messages are sent by the KMC Connect PDS.

Align Intervals If selected and a Time Synchronization Interval is specified, the time synchronization messages are sent at the start of the hour or day.

Interval Offset If Align Intervals is selected, the time synchronization messages are offset, in minutes, from the beginning of the hour or day.

Time Synchronization Recipients This property is a list of recipient devices or addresses that will receive time synchronization messages from the device. If the list is empty, time synchronization is not sent.

UTC Time Synchronization Recipients This property is a list of recipient devices that will receive a UTC synchronization message from the device. If the list is empty, UTC synchronization is not sent.

Active COV Subscription Properties

Active Change of Value Subscriptions This property provides a network-visible indication of active COV subscriptions. The list consists of a Recipient, a Monitored Property Reference, an Issue Confirmed Notifications flag, a Time Remaining value and an optional COVIncrement.

Service and Object Type Properties

Supported Protocol Service Types This property is a list of the standard BACnet services that are supported by the device.

Supported Protocol Object Types This property is a list of the standard BACnet object types that are supported by the device.

Address Binding Properties

Device Address Bindings The Device Address Bindings property is a list of BACnet Address Binding each of which consists of a BACnet Object Identifier of a BACnet Device object and a BACnet device address in the form of a BACnetAddress. Entries in the list identify the actual device addresses that will be used when the remote device must be accessed with a BACnet service request. A value of zero shall be used for the network-number portion of BACnetAddress entries for other devices residing on the same network as this device. The list may be empty if no device identifier-device address bindings are currently known to the device.

Slave Proxy Enable This property indicates ***True*** if the device will perform Slave-Proxy functions for each of the MS/TP ports represented by each array element. It indicates ***False*** if it will not perform the Slave-Proxy functions.

Auto Slave Discovery This property indicates ***True*** if the device will perform automatic slave detection functions for each of the MS/TP ports represented by each array element.

Manual Slave Address Binding This property is a list of the manually configured set of slave devices for which this device is acting as a Slave Proxy.

Slave Address Binding This property is a list of slave devices for which this device is acting as a Slave-Proxy.

VT Class and Session Properties

Supported VT Classes This property is a network-visible list of terminal sessions (VT Sessions) that are supported within the device. The property is a list that consists of a local VT Session identifier, a Remote VT Session identifier, and remote VT Session address.

Active VT Sessions This property is a network-visible list of the active VT-Sessions within the device.

Structured Object List Property

This property is a hierarchical list of Life Safety Zone and Structured View objects within the device.

- To add a new object, click + to add an object and then choose the object from the drop down list.
- To delete an object click -.

Application Properties

Applies to Conquest controllers only. For list of applicable models see the topic *Conquest compliant controllers* on page 469.

Base Application Index Changes the base application index to change the units used in a Conquest application. A restart is required after making a change.

Base Application Name The name of the current enabled application.

Deployed Application Name The name of the deployed application. This is empty if an application has not been deployed.

Deployed Application ID The KMC identification number assigned to the deployed application. This is empty if an application has not been deployed.

Event enrollment object

An event enrollment object is a standard BACnet object that monitors a property in another BACnet object for alarm or event conditions. When the condition is detected, a notification is sent to a notification class object for further handling. The supported BACnet algorithms are listed in the table Event Types, Event States, and Event Parameters.

General Properties

Object Instance

The number of the object. Event enrollment objects are numbered sequentially within the KMC Controls BACnet device.

Object Name A text label of the object. The exact length will depend upon the controller that contains the object. The Name property must be unique within the BACnet device that maintains it. The set of characters entered for the Name property must be printable characters.

Profile Name The name of the vendor's object profile to which this object conforms. A profile name defines a set of additional properties, behaviors and/or requirements for this object. The definition of the profile is vendor unique and not part of the BACnet standard.

Description A text label of the object. The exact length of the Description property will depend upon the controller that contains the object.

Device, Object and Property These parameters designate the referenced property. Enter the device instance number, the object number and the property to monitor.

- Device–an object within a BACnet device on the internetwork. Choose **Local** to select the device within which the trend log is located.
- Object–Choose the object from the Object list or drag the object from the Network Manager list. For KMC BACnet controllers the object must be within the BACnet device as the event enrollment object.
- Property–Choose the property within the designated object. Typically the property is **Present Value**.

Event Type The Event Type property specifies which of the standard algorithms should be applied when monitoring the referenced object as entered under Device, Object and Property. The selection of the Event Type changes the display of the parameter values needed for each algorithm. Each of the parameters are described in the following topics.

Table 30-3 Event Types, Event States, and Event Parameters

Event Type	Event State	Event Parameters
CHANGE OF BITSTRING	NORMAL OFFNORMAL	Time Delay Bitmask List Of Bitstring Values
CHANGE OF STATE	NORMAL OFFNORMAL	Time Delay List Of Values
CHANGE OF VALUE	NORMAL OFFNORMAL	Time Delay Bitmask Referenced Property Increment
COMMAND FAILURE	NORMAL OFFNORMAL	Time Delay Feedback Property Reference
FLOATING LIMIT	NORMAL HIGH LIMIT LOW LIMIT	Time Delay Setpoint Reference Low Diff Limit High Diff Limit Deadband
OUT OF RANGE	NORMAL HIGH LIMIT LOW LIMIT	Time Delay Low Limit High Limit Deadband

Event Types, Event States, and Event Parameters (continued)

Event Type	Event State	Event Parameters
BUFFER READY	NORMAL	Notification Threshold
CHANGE OF LIFE SAFETY	NORMAL OFFNORMAL LIFE SAFET ALARM	Time Delay List Of Alarm Values List Of Life Safety Alarm Values Mode Property Reference
EXTENDED	Any BACnet event state	Extended Event Parameters
UNSIGNED RANGE	NORMAL HIGH LIMIT LOW LIMIT	Time Delay Low limit High Limit
CHANGE OF STATUS FLAG	NORMAL OFFNORMAL	Time Delay Selected Flags

The selection of Event Type changes the display of the parameter values needed for each algorithm. Each of the parameters are described in the following topics.

Bitmask Applies when the Event Type property is set to CHANGE OF BITSTRING. The selected bits represent a bitmask that indicates which bits in the referenced property are to be monitored by the algorithm.

- A selected check box next to a bit indicates that the bit in the referenced property is to be monitored by the algorithm.
- A cleared check box next to a bit indicates that the bit in the referenced property is not significant for the purpose of detecting Change Of Bitstring.

List of Bitstrings This list defines the set of states for which the referenced property is Off Normal. Only the bits selected in the Bitmask property are significant. If the value of the referenced property changes to one of the values in the List of Bitstring property, then the Event State property of the Event Enrollment object changes to To Off Normal and appropriate notifications are sent to the Notification Class object.

List Of Values Applies when Event Type is set to CHANGE OF STATE. If the value of the referenced property changes to one of the values in the List Of Values, then the value of Event State changes to To Off Normal and notifications are sent to the Notification Class object.

Referenced Property Increment This parameter represents the increment by which the referenced property must change to initiate an event.

Time Delay This parameter represents the time—in seconds—that the conditions monitored by the event algorithm must persist before an event notification is issued.

Feedback Property Reference This parameter applies when Event Type is set to COMMAND FAILURE. It identifies the object and property that provides the feedback to

ensure that the commanded property has changed value. This property may reference only object properties that have enumerated values or are of type BOOLEAN.

Setpoint Reference This parameter applies when Event Type is set to FLOATING LIMIT. It indicates the setpoint reference for the reference property interval.

High Limit This parameter applies when Event Type is set to OUT OF RANGE. It defines the upper limit for a normal operating range of the monitored property in the referenced object.

Low Limit This parameter applies when Event Type is set to OUT OF RANGE. It defines the lower limit for a normal operating range of the monitored property in the referenced object.

Deadband This parameter applies when Event Type is set to FLOATING LIMIT and OUT OF RANGE. It specifies a range between the high limit and low limit properties in which the monitored property in the referenced object must remain before the object generates a notification.

High Diff Limit This parameter applies when Event Type is set to FLOATING LIMIT. When added to Setpoint Reference it defines an upper limit for a normal operating range of the monitored property in the referenced object.

Low Diff Limit This parameter applies when Event Type is set to OUT OF RANGE. When added to Setpoint Reference it defines an lower limit for a normal operating range of the monitored property in the referenced object.

Notification Threshold This parameter applies when Event Type is set to BUFFER READY. It specifies the value of Records Since Notification at which notification occurs.

List Of Life Safety Alarm Values This parameter applies when Event Type is set to CHANGE OF LIFE SAFETY. If the value of the referenced property changes to one of the values in the list of Life Safety Alarm Values, then the value of Event State changes to To Off Normal and appropriate notifications are sent to the Notification Class object.

Alarm Values This parameter applies when Event Type is set to CHANGE OF LIFE SAFETY. It is a list of states that apply to the CHANGE OF LIFE SAFETY algorithm. If the value of the referenced property changes to one of the values in the Alarm Values, then the value of Event State changes to To Off Normal and appropriate notifications are sent to the Notification Class object.

Mode Property Reference This parameter applies when Event Type is set to CHANGE OF LIFE SAFETY. It identifies the object and property that provides the operating mode of the referenced object providing life safety functionality (normally the Mode property). This parameter may reference only object properties for BACnet Life Safety.

Extended Event Parameters The interpretation of Extended Event Parameters is determined by the vendor of the device.

Event/Alarm Properties

Related topics

- *Notification object* on page 365
- *Configuring devices for BACnet alarms* on page 105

Notification Class Specifies the notification class object to be used when handling and generating event notifications for this object. See *Notification object* on page 365 for details about the notification class object.

Notification Type This property specifies whether the notifications generated by the object are events or alarms. Alarm and event notifications are handled differently by the device—usually a workstation—that receives the notification.

Event Enable Use Event Enable to enable notifications for To Off Normal, and To Fault.

Status Properties

Event Type Displays the value of Event Type as selected under General Properties.

Event State Use Event State to determine that this object has an active event state associated with it.

- If the object supports intrinsic reporting, then Event State indicates the state of the object.
- If the object does not support intrinsic reporting, then the value of this property is Normal.

Acked Transitions This property controls three flags that separately indicate the receipt of acknowledgments for To Off Normal, To Fault, and To Normal events.

Event Time Stamps This property holds the times of the last event notifications of *TO OFF NORMAL, TO FAULT,* and *TO NORMAL* events.

Alarm Routing Properties

Select the KMC Connect alarm class routing from the list.

Event Log objects

The Event Log object records alarm and event notifications and then saves them with time stamps and other pertinent data. The data is stored as records in an internal buffer for subsequent retrieval.

To configure the properties in a BACnet device see *Configuring BACnet devices and objects* on page 300.

General Properties

Object Instance The number of the object. Event Log objects are numbered sequentially within a BACnet device.

Object Name A text label of the object. The exact length will depend upon the controller that contains the object. The Name property must be unique within the BACnet device that maintains it. The set of characters entered for the Name property must be printable characters.

Profile Name The name the vendor's object profile to which this object conforms. A profile name defines a set of additional properties, behaviors and/or requirements for this object. The definition of the profile is vendor unique and not part of the BACnet standard.

Description A text label of the object. The exact length of the Description property will depend upon the controller that contains the object.

Start and Stop Date and Time Set the period for logging data with the Start Time and Stop Time properties.

Buffer Size Holds the value of the maximum number of records that the Event Log buffer may hold.

Notification Threshold When the number of records in the buffer reaches the value of the Notification Threshold property, a notification is sent to the Notification Class object specified by this Event Log object.

Log Enable If this check box is clear, the Event Log object does not log data. If the Log Enable check box is selected and the current time and date are within the range of time and dates specified by Start Time/Date and Stop Time/Date the object will log notifications.

Stop When Full When selected, the object stops adding new records to the log buffer when the buffer is full. If the check box is clear, the oldest data record is replaced with a new record when the buffer is full.

Event/Alarm Properties

Notification Class Specifies the notification class object to be used when handling and generating event notifications for this object. See ***Notification object*** on page 365 for details about the notification class object.

Notification Type This property specifies whether the notifications generated by the object are events or alarms. Alarm and event notifications are handled differently by the device—usually a workstation—that receives the notification.

Event Enable Use the Event Enable property to enable notifications for To Off Normal, To Normal, or To Fault conditions.

Status Properties

Record Count This property displays the number of records currently resident in the log buffer. Writing a value of zero to this property will delete all records in the log buffer and the Records Since Notification property will change to zero. Upon completion, this event is added to the log as the initial entry.

Total Record Count This property represents the total number of records collected by the Event Log object since creation. When the value of property reaches its maximum value of 4,294,967,295, the next value is 1.

Records Since Last Notification This property represents the number of records collected since the previous notification or since the beginning of logging if no previous notification has occurred. This property is required if intrinsic reporting is supported by this object.

Last Notify Record This property represents the sequence number associated with the most recently collected record whose collection triggered a notification. If no notification

has occurred since logging began the value of this property is zero. This property is required if intrinsic reporting is supported by this object.

Status Flags The four BACnet status flags are an indication of the general condition of the object.

Table 30-4 Event log object status flags

Flag	Description
IN ALARM	False (0) if the event state property is Normal, other wise True (1)
FAULT	True (1) if Reliability is present and the value for Reliability is not NO FAULT DETECTED, otherwise False (0)
OVERRIDDEN	True (1) if the point has been overridden by some mechanism local to the BACnet device. When this flag is True, Present Value and Reliability are no longer tracking changes to the physical output. Otherwise, the flag is False (0).
OUT OF SERVICE	True (1) if Out Of Service is selected. Otherwise False(0).

Event State Use Event State to determine that this object has an active event state associated with it.

- If the object supports intrinsic reporting, then Event State indicates the state of the object.
- If the object does not support intrinsic reporting, then the value of this property is Normal.

Acked Transitions This property controls three flags that separately indicate the receipt of acknowledgments for To Off Normal, To Fault, and To Normal events.

Event Time Stamps This property holds the times of the last event notifications of *TO OFF NORMAL, TO FAULT,* and *TO NORMAL* events.

Reliability This property is an indication that the Present Value property or the operation of the physical input may be not be reliable as defined by the device or an operator. Reliability can have any of the following values:

- No Fault Detected
- Configuration Error
- Communication Failure
- Unreliable Other

Alarm Routing Properties

Select the KMC Connect alarm class routing from the list.

Log Buffer Records

A list of all of the time stamped records of notifications or status change in the event log. Selecting any individual record will display its contents in the lower half of the pane.

File objects

Working with file objects is an advanced topic and not covered here. Do not make changes to file objects unless specifically directed by the Technical Support Department at KMC Controls, Inc.

Group objects

The Group object defines a collection of properties from other objects and one or more of their properties. Use the object to simplify the simultaneous collection of information from several objects. The group of objects can consist of any combination of objects that reside within the same device as the Group object.

To configure the properties in a BACnet device see *Configuring BACnet devices and objects* on page 300.

Object Instance The number of the object. Group objects are numbered sequentially within a BACnet device.

Object Name A text label of the object. The exact length will depend upon the controller that contains the object. The Name property must be unique within the BACnet device that maintains it. The set of characters entered for the Name property must be printable characters.

Profile Name The name the vendor's object profile to which this object conforms. A profile name defines a set of additional properties, behaviors and/or requirements for this object. The definition of the profile is vendor unique and not part of the BACnet standard.

Profile Name The name the vendor's object profile to which this object conforms. A profile name defines a set of additional properties, behaviors and/or requirements for this object. The definition of the profile is vendor unique and not part of the BACnet standard.

Description A text label of the object. The exact length of the Description property will depend upon the controller that contains the object.

Group Members This is a list of all objects and properties that are members of the group. All objects in the list must be within the device that contains the Group object.

Present Value A list of all the values of each property specified by the Group Members property. The Present Value property is read-only in a Group object.

Input object–accumulator

An accumulator input is a standard BACnet object whose properties represent the characteristics of a pulse signal at the input of a controller. Use the input object tool to configure each of the controllers inputs with one of the three input object types.

To configure a property, see the topic *Configuring BACnet devices and objects* on page 300.

Related topics

- *BACnet objects and properties* on page 291
- *Input object–analog* on page 340
- *Input object–binary* on page 345
- *Input objects–multistate* on page 348

General Properties

Object Instance The number of the object. Input objects are numbered sequentially within the KMC Controls BACnet device; the object instance number directly correspond to the controller's input terminal.

Object Name A text label of the object. The exact length will depend upon the controller that contains the object. The Name property must be unique within the BACnet device that maintains it. The set of characters entered for the Name property must be printable characters.

Present Value This property indicates the current value of the number of input pulses acquired by the object since the value was set with the Value Set property. If the object includes a Prescale property, Present Value is expressed in the prescaled number of pulses.

Profile Name The name the vendor's object profile to which this object conforms. A profile name defines a set of additional properties, behaviors and/or requirements for this object. The definition of the profile is vendor unique and not part of the BACnet standard.

Description A text label of the object. The exact length of the Description property will depend upon the controller that contains the object.

Device Type This property is a description of the physical device connected to the input. Choose an available type from the drop-down list.

Logging Object This property indicates an object in the same device as the accumulator object which, when it acquires data from Logging Record in the accumulator object, causes the accumulator object to acquire, present and store the data from the underlying system.

Logging Record This read-only property is a list of values that must be acquired and returned for proper interpretation of the data.

Out Of Service Out Of Service indicates that the physical input is internally disconnected from the input object. KMC Connect sets this property to True (1) when selected and False (0) when unselected. When Out Of Service is selected, and sent to the controller, Present Value does not respond to changes at the physical input of the device.

Max Present Value This property indicates the maximum value of Present Value.

Units Select a unit of measure to associate with the input signal. See *Supported engineering units* on page 417 for a list of the available units.

Pulse Rate Pulse Rate indicates the number of input pulses received during the most recent period specified by the value in Limit Monitoring Interval.

Limit Monitoring Interval This property specifies the monitoring period—in seconds—for determining the value of Pulse Rate.

Value Change Time This property holds the date and time of the most recent occurrence of a write operation to Value Before Change or Value Set properties. If a write has not yet occurred, this property consists of wildcard values for all date and time fields.

Value Before Change Value Before Change indicates the value of the Present Value property just prior to the most recent write to the Value Set or Value Before Change properties. If a write has not yet occurred, the value of this property is zero (0). If this property is writable, Value Set is read-only.

Value Set This property indicates the value of Present Value after the most recent write to **Value Set** or Value Before Change. If a write has not yet occurred, the value of this property is zero (0). If this property is writable, Value Before Change is read-only.

Scale Multiply Present Value by Scale to provide a value in the units indicated by Units.

Prescale This property presents the coefficients that are used for converting the pulse signals generated by the measuring instrument into the value displayed by Present Value.

Event/Alarm Properties

Notification Class Specifies the notification class object to be used when handling and generating event notifications for this object. See *Notification object* on page 365 for details about the notification class object.

Notification Type This property specifies whether the notifications generated by the object are events or alarms. Alarm and event notifications are handled differently by the device—usually a workstation—that receives the notification.

High Limit This property is used with intrinsic reporting to define a limit that Pulse Rate must exceed before an event is generated. Use with Limit Enable and Time Delay to define conditions that indicate Pulse Rate is out of a normal operating range.

Low Limit This property is used with intrinsic reporting to define a limit that Pulse Rate must fall below before an event is generated. Use with Limit Enable and Time Delay to define events that indicate Pulse Rate is out of a normal operating range.

Time Delay Time Delay defines a minimum period for a set of conditions to exist before a To Off Normal or To Normal notification event occurs. Use Time Delay with High Limit and Low Limit to define conditions that indicate Pulse Rate is out of an expected, predefined operating range. Time Delay is expressed in seconds.

Limit Enable This property separately enables and disables reporting of high limit and low limit Off Normal events and their return to normal.

Event Enable Use Event Enable to enable notifications for To Off Normal, To Normal and To Fault.

Status Properties

Status Flags The four BACnet status flags are an indication of the general condition of the object.

Table 30-5 Input object status flags

Flag	Description
IN ALARM	False (0) if the event state property is Normal, other wise True (1)
FAULT	True (1) if Reliability is present and the value for Reliability is not NO FAULT DETECTED, otherwise False (0)
OVERRIDDEN	True (1) if the point has been overridden by some mechanism local to the BACnet device. When this flag is True, Present Value and Reliability are no longer tracking changes to the physical input. Otherwise, the flag is False (0).
OUT OF SERVICE	True (1) if Out Of Service is selected. Otherwise False(0).

Reliability This property is an indication that the Present Value property or the operation of the physical input may be not be reliable as defined by the device or an operator. Reliability can have any of the following values:

- NO_FAULT_DETECTED
- UNDER_RANGE
- UNRELIABLE_OTHER
- NO_SENSOR
- OPEN_LOOP
- OVER_RANGE
- SHORTED_LOOP

Event State Use Event State to determine that this object has an active event state associated with it.

- If the object supports intrinsic reporting, then Event State indicates the state of the object.
- If the object does not support intrinsic reporting, then the value of this property is Normal.

If Reliability is present and does not have a value of No_fault_detected, then the value of Event_State is Fault. Changes in the Event State to the value tem">Fault are considered to be fault events.

Acked Transitions This property controls three flags that separately indicate the receipt of acknowledgments for To Off Normal, To Fault, and To Normal events. For analog input objects, the transition of High Limit and Low Limit are considered to be Off Normal events. These flags are cleared upon the occurrence of the corresponding event and set under any of the following conditions:

- Upon receipt of the corresponding acknowledgment

- Upon the occurrence of the event if the corresponding flag is not set in the Event Enable property. Under this condition, event notifications are not generated for this condition and therefore no acknowledgments are expected.
- Upon the occurrence of the event if the corresponding flag is set in the Event Enable property and the corresponding flag in the Ack Required property of the notification class object implicitly referenced by the notification class property of this object is not set (meaning no acknowledgment is expected.)

Event Time Stamps Holds the times of the last event notifications of To Off Normal, To Fault and To Normal events.

Alarm Routing Properties

Select the KMC Connect alarm class routing from the list.

Input object–analog

An analog input is a standard BACnet object. The properties in the object represent the characteristics of an analog signal at the input of a controller. Use the input object tool to configure each of the controllers inputs with one of the three input object types.

To configure a property, see the following topics:

- *Analog input object wizard* on page 441
- *Configuring BACnet devices and objects* on page 300
- *Calibration* on page 342

Related topics

- *BACnet objects and properties* on page 291
- *Input object–accumulator* on page 336
- *Input object–binary* on page 345
- *Input objects–multistate* on page 348

General Properties

Object Instance The input object number. Input objects are numbered sequentially within the KMC Controls BACnet device and directly correspond to the controller's input terminal.

Object Name A text label of the object. The exact length will depend upon the controller that contains the object. The Name property must be unique within the BACnet device that maintains it. The set of characters entered for the Name property must be printable characters.

Present Value This numerical property indicates the current value of the input being measured. To manually change the present value property, first select **Out-Of-Service** and then change **Present Value**.

Profile Name The name the vendor's object profile to which this object conforms. A profile name defines a set of additional properties, behaviors and/or requirements for this object. The definition of the profile is vendor unique and not part of the BACnet standard.

Description A text label of the object. The exact length of the Description property will depend upon the controller that contains the object.

Device Type This property is a description of the physical device connected to the input. Choose an available type from the drop-down list.

Termination For Conquest compliant controllers, this selects and indicates the input termination. For other controllers, this is an indication of how the termination should be set.

Out Of Service Out Of service indicates that the physical input is internally disconnected from the input object. KMC Connect sets this property to True (1) when selected and False (0) when unselected. When Out Of Service is selected, and sent to the controller, Present Value does not respond to changes at the physical input of the device.

Units Select a unit of measure to associate with the input signal. See *Supported engineering units* on page 417 for a list of the available units.

Minimum Present Value This property indicates the smallest input value—expressed in engineering units—that can reliably be obtained for the input object Present Value.

Maximum Present Value This property indicates the largest input value—expressed in engineering units—that can reliably be obtained for the input object Present Value.

COV Increment This property specifies the minimum change of Present Value that will send a COV notification to subscriber notification clients.

Resolution This property indicates the smallest recognizable change—expressed in engineering units— in the present value of this object.

Update Interval This property sets the maximum period—in hundredths of a second—between updates to **Present Value**.

Filter Weight Sets the number of samples that are averaged together to calculate the displayed value. A sample is taken on each scan.

Table 30-6 Filter weight values

Filter Weight	Scans to average
0	1
1	2
2	4
3	8
4	16
5	32
6	64
7	128

Calibration

1. To add a calibration factor to a BACnet analog input object, do the following:
2. Open the configuration tab for the object.
3. Right-click over **Present Value**.
4. Choose **Calibrate Present Value** from the shortcut menu.
5. Enter a calibration factor to adjust the input for sensor inaccuracies.
   - For a low reading enter a positive correction value.
   - For a high reading enter a negative correction value.
   - The valid range for a calibration factor is from -30 to 30.
   - The default value is 0 (no calibration).

Custom Device Calibration

For objects that have Device Type set to CUSTOM, additional calibration factors can be added. The Custom Device Calibration function converts the input voltage—such as the voltage from a temperature transmitter—to a calculated Present Value. The calculation is based on an input voltage that falls between 0 and 5 volts.

To set the values for Custom Device Calibration do the following:

1. Open the configuration tab for the object.
2. Right-click over **Present Value**.
3. Choose **Custom Device Calibration** from the shortcut menu.
4. On a BAC-A1616BC or CAN-A168EIO, if the 0-12 VOLT jumper is in place, select the **Jumpered (12 volt)** check box.
5. Enter values in either From Range and To Range or Offset and Multiplier.

Range Enter values in Desired Range text boxes when two points are known that correspond to 0 and 5 volts at the input.

- In **Desired Range To** enter the value to display in Present Value when the input voltage equals 5 volts.
- In **Desired Range From** enter the value to display in Present Value when the input voltage equals 0 volts.

Offset and multiplier Use offset and multiplier when a calibration factor and offset for the input device are known. For example if a transducer output is calibrated for degrees Centigrade and Present Value needs to display in degrees Fahrenheit.

Event/Alarm Properties

Related topics

Notification Class Specifies the notification class object to be used when handling and generating event notifications for this object. See *Notification object* on page 365 for details about the notification class object.

Notification Type This property specifies whether the notifications generated by the object are events or alarms. Alarm and event notifications are handled differently by the device—usually a workstation—that receives the notification.

High Limit This property is used with intrinsic reporting to define an upper limit for a normal operating range of Present Value. Use with Limit Enable, Deadband and Time Delay to define conditions that indicate Present Value is out of a normal operating range.

Low Limit This property is used with intrinsic reporting to define a lower limit for a normal operating range of Present Value. Use with Limit Enable, Deadband and Delay to define events that indicate Present Value is out of a normal operating range.

Deadband This property specifies a range between the high limit and low limit properties in which Present Value must remain before the device generates a To Normal notification event.

Time Delay Time Delay defines a minimum period for a set of conditions to exist before a To Off Normal or To Normal notification event occurs. Use Time Delay with High Limit, Low Limit and Deadband to define conditions that indicate Present Value is out of an expected, predefined operating range. Time Delay is expressed in seconds.

Limit Enable This property separately enables and disables reporting of high limit and low limit Off Normal events and their return to normal.

Event Enable Use the Event Enable property to enable notifications for To Off Normal, To Normal, or To Fault conditions.

Status Properties

Status Flags

The four BACnet status flags are an indication of the general condition of the analog input.

Table 30-7 Input object status flags

Flag	Description
IN ALARM	False (0) if the event state property is Normal, other wise True (1)
FAULT	True (1) if Reliability is present and the value for Reliability is not NO FAULT DETECTED, otherwise False (0)
OVERRIDDEN	True (1) if the point has been overridden by some mechanism local to the BACnet device. When this flag is True, Present Value and Reliability are no longer tracking changes to the physical input. Otherwise, the flag is False (0).
OUT OF SERVICE	True (1) if Out Of Service is selected. Otherwise False(0).

Reliability This property is an indication that the Present Value property or the operation of the physical input may be not be reliable as defined by the device or an operator. Reliability can have any of the following values:

- NO_FAULT_DETECTED
- UNDER_RANGE
- UNRELIABLE_OTHER
- NO_SENSOR
- OPEN_LOOP
- OVER_RANGE
- SHORTED_LOOP

Event State Use Event State to determine that this object has an active event state associated with it.

- If the object supports intrinsic reporting, then Event State indicates the state of the object.
- If the object does not support intrinsic reporting, then the value of this property is Normal.

If Reliability is present and does not have a value of No_fault_detected, then the value of Event_State is Fault. Changes in the Event State to the value Fault are considered to be fault events.

Acked Transitions This property controls three flags that separately indicate the receipt of acknowledgments for To Off Normal, To Fault, and To Normal events. For analog input objects, the transition of High Limit and Low Limit are considered to be Off Normal events. These flags are cleared upon the occurrence of the corresponding event and set under any of the following conditions:

- Upon receipt of the corresponding acknowledgment
- Upon the occurrence of the event if the corresponding flag is not set in the Event Enable property. Under this condition, event notifications are not generated for this condition and therefore no acknowledgments are expected.
- Upon the occurrence of the event if the corresponding flag is set in the Event Enable property and the corresponding flag in the Ack Required property of the notification class object implicitly referenced by the notification class property of this object is not set (meaning no acknowledgment is expected.)

Event Time Stamps This property holds the times of the last event notifications of *TO OFF NORMAL, TO FAULT,* and *TO NORMAL* events.

Alarm Routing Properties

Select the KMC Connect alarm class routing from the list.

Input object–binary

An binary input is a standard BACnet object. The properties in the object represent the characteristics of a binary signal at the input of a controller. Use the input object tool to configure each of the controllers inputs with one of the three input object types.

To configure a property, see the topics *Input object wizards* on page 441 and *Configuring BACnet devices and objects* on page 300.

Related topics

- *BACnet objects and properties* on page 291
- *Input object–accumulator* on page 336
- *Input object–analog* on page 340
- *Input objects–multistate* on page 348

General Properties

Object Instance The input object number. Input objects are numbered sequentially within the KMC Controls BACnet device and directly correspond to the controller's input terminal.

Object Name A text label of the object. The exact length will depend upon the controller that contains the object. The Name property must be unique within the BACnet device that maintains it. The set of characters entered for the Name property must be printable characters.

Present Value This numerical property indicates the logical state of the input. A binary input will be in one of two states such as Start/Stop, On/Off or True/False.

To manually change Present Value, first select Out-Of-Service and then change Present Value.

Profile Name The name the vendor's object profile to which this object conforms. A profile name defines a set of additional properties, behaviors and/or requirements for this object. The definition of the profile is vendor unique and not part of the BACnet standard.

Description A text label of the object. The exact length of the Description property will depend upon the controller that contains the object.

Termination For Conquest compliant controllers, this selects and indicates the input termination. For other controllers, this is an indication of how the termination should be set.

Device Type This property is a description of the physical device connected to the input.

Out Of Service Out Of service indicates that the physical input is internally disconnected from the input object. KMC Connect sets this property to True when selected and False when not selected. When Out Of Service is selected, and sent to the controller, Present Value does not respond to changes at the physical input of the device.

PolarityThe polarity property sets the relationship between the physical state of the input and the logical state represented by Present Value. KMC Connect displays Present value as either Active or Inactive Text.

Table 30-8 Input object polarity relationships

Passive input with pull-up	Polarity	Present Value	Text example
	Normal	Active	Off, Stop
	Normal	Inactive	On, Start
	Reverse	Inactive	On, Start
	Reverse	Active	Off, Stop

Active Text Text to display when Present Value is in the active state.

Inactive Text Text to display when Present Value is in the inactive state.

Event/Alarm Properties

Notification Class Specifies the notification class object to be used when handling and generating event notifications for this object. See *Notification object* on page 365 for details about the notification class object.

Notification Type This property specifies whether the notifications generated by the object are events or alarms. Alarm and event notifications are handled differently by the device—usually a workstation—that receives the notification.

Alarm Value When the object's present value changes to the value in Alarm Value, a notification event is generated.

Time Delay Time Delay defines a minimum period for a set of conditions to exist before a To Off Normal or To Normal event occurs. Use Time Delay with Alarm Value to define conditions that indicate Present Value has changed states.

Event Enable Use Event Enable to enable notifications for To Off Normal, TO NORMAL and To Fault.

Status Properties

Status Flags The four BACnet status flags are an indication of the general condition of the object.

Table 30-9 Input object status flags

Flag	Description
IN ALARM	False (0) if the event state property is Normal, otherwise True (1)
FAULT	True (1) if Reliability is present and the value for Reliability is not NO FAULT DETECTED, otherwise False (0)

Table 30-9 Input object status flags (continued)

Flag	Description
OVERRIDDEN	True (1) if the point has been overridden by some mechanism local to the BACnet device. When this flag is True, Present Value and Reliability are no longer tracking changes to the physical input. Otherwise, the flag is False (0).
OUT OF SERVICE	True (1) if Out Of Service is selected. Otherwise False(0).

Reliability This property is an indication that the Present Value property or the operation of the physical input may be not be reliable as defined by the device or an operator. Reliability can have any of the following values:

- NO_FAULT_DETECTED
- SHORTED_LOOP
- UNRELIABLE_OTHER
- NO_SENSOR
- OPEN_LOOP

Time of Active Time Reset This property holds the date and time when Elapsed Active Time was most recently set to zero.

Elapsed Active Time This property holds the accumulated number of seconds that Present Value has been in the active state since Elapsed Active Time was set to zero.

Event State Use the Event State property to determine that this input object has an active event state associated with it.

- If the input supports intrinsic reporting, then Event State indicates the state of the object.
- If the input does not support intrinsic reporting, then the value of this property is Normal.
- If Reliability is present and does not have a value of No Fault Detected, then the value of Event State is Fault. Changes in Event State to the value Fault are considered to be fault events.

Acked Transitions This property controls three flags that separately indicate the receipt of acknowledgments for To Off Normal, To Fault, and To Normal events. These flags are cleared upon the occurrence of the corresponding event and set under any of the following conditions:

- Upon receipt of the corresponding acknowledgment
- Upon the occurrence of the event if the corresponding flag is not set in the Event Enable property. Under this condition, event notifications are not generated for this condition and therefore no acknowledgments are expected.

- Upon the occurrence of the event if the corresponding flag is set in the Event Enable property and the corresponding flag in the Ack Required property of the Notification Class object implicitly referenced by the Notification Class property of this object is not set (meaning no acknowledgment is expected.

Event Time Stamps This property holds the times of the last event notifications of *TO OFF NORMAL, TO FAULT,* and *TO NORMAL* events.

Change of State Time This property holds the date and time that Present Value most recently changed state.

Change of State Count This property holds the number of times that Present Value has changed state since Change of State Count was most recently set to zero (0).

Time of State Count Reset This property holds the date and time that Change of State Count was most recently set to zero.

Alarm Routing Properties

Select the KMC Connect alarm class routing from the list.

Input objects–multistate

The Multistate Input object is a standard BACnet object whose Present Value property represents the result of a calculation performed within the device. Present Value may be the result of a logical combination of multiple binary inputs or the threshold of one or more analog inputs or the result of a mathematical computation. The result of the calculation is a single number that represents the state of the object.

To configure a property, see the topic *Configuring BACnet devices and objects* on page 300.

Related topics

- *BACnet objects and properties* on page 291
- *Input object–accumulator* on page 336
- *Input object–analog* on page 340
- *Input object–binary* on page 345

General Properties

Object Instance The number of the object. Input objects are numbered sequentially within a BACnet device.

Object Name A text label of the object. The exact length will depend upon the controller that contains the object. The Name property must be unique within the BACnet device that maintains it. The set of characters entered for the Name property must be printable characters.

Present Value

This numerical property indicates the current state of the Mulitstate Input object. The number of states for this property is defined by the value of the Number Of States property.

Profile Name The name the vendor's object profile to which this object conforms. A profile name defines a set of additional properties, behaviors and/or requirements for this object. The definition of the profile is vendor unique and not part of the BACnet standard.

Description A text label of the object. The exact length of the Description property will depend upon the controller that contains the object.

Device Type A text description that is typically used to describe the type of device attached to the Multistate Input object.

Termination For Conquest compliant controllers, this selects and indicates the input termination. For other controllers, this is an indication of how the termination should be set.

Out Of Service Out Of Service indicates that the physical input is internally disconnected from the input object. KMC Connect sets this property to True (1) when selected and False (0) when unselected. When Out Of Service is selected, and sent to the controller, Present Value does not respond to changes at the physical input of the device.

Number of States The Number of States property defines the number of states that Present Value may have. It is always greater than zero. The value of Number Of States will automatically change as entries are added or removed from the State Text list.

State Text Each entry in the State Text list corresponds to a value of Present Value. For example, if Present Value equals 1 then the value of State Text is the first entry in the list.

- To start a new entry in State Text, right-click and choose **Add String** from the shortcut menu.
- To change the text of an entry in State Text, right-click an entry and choose **Edit String** from the shortcut menu.
- To delete an entry from State Text, right-click an entry and choose **Remove String** from the shortcut menu.
- To change the order of the list in State Text, right-click an entry and choose **Move Up** or **Move Down** from the shortcut menu.

Event/Alarm Properties

Notification Class Specifies the notification class object to be used when handling and generating event notifications for this object. See *Notification object* on page 365 for details about the notification class object.

Notification Type This property specifies whether the notifications generated by the object are events or alarms. Alarm and event notifications are handled differently by the device—usually a workstation—that receives the notification.

Time Delay Time Delay defines a minimum period for a set of conditions to exist before a To Off Normal or To Normal notification occurs. Use Time Delay with the Alarm Values property to define conditions that indicate Present Value has changed state. Time Delay is expressed in seconds.

Event Enable Use the Event Enable property to enable notifications for To Off Normal, To Normal, or To Fault conditions.

Alarm Values This property is used with intrinsic reporting to define a change of the Present Value property that will generate an alarm event.

Fault Values This property list any states that the Present Value property must equal for a To Fault notification to be generated.

Status Properties

Status Flags The four BACnet status flags are an indication of the general condition of the object.

Table 30-10 Multistate input object status flags

Flag	Description
IN ALARM	False (0) if the event state property is Normal, other wise True (1)
FAULT	True (1) if Reliability is present and the value for Reliability is not NO FAULT DETECTED, otherwise False (0)
OVERRIDDEN	True (1) if the point has been overridden by some mechanism local to the BACnet device. When this flag is True, Present Value and Reliability are no longer tracking changes to the physical input. Otherwise, the flag is False (0).
OUT OF SERVICE	True (1) if Out Of Service is selected. Otherwise False(0).

Reliability This property is an indication that the Present Value property or the operation of the physical input may be not be reliable as defined by the device or an operator. Reliability can have any of the following values:

- No Fault Detected
- No Sensor
- Over Range
- Under Range
- Open Loop
- Shorted Loop
- Multistate Fault
- Communication Failure
- Unreliable Other

Event State Use Event State to determine that this object has an active event state associated with it.

- If the object supports intrinsic reporting, then Event State indicates the state of the object.
- If the object does not support intrinsic reporting, then the value of this property is Normal.

Acked Transitions This property controls three flags that separately indicate the receipt of acknowledgments for To Off Normal, To Fault, and To Normal events.

Event Time Stamps This property holds the times of the last event notifications of *TO OFF NORMAL, TO FAULT,* and *TO NORMAL* events.

Alarm Routing Properties

Select the KMC Connect alarm class routing from the list.

Life Safety Point object

A Life Safety Point object is a standard BACnet object that represents the characteristics of an indicating or initiating device. The device is typically used for fire, life safety, or security applications. The condition of the Life Safety Point object is determined by the mode and state of the object.

- The mode of the object mode is usually under the control of the device operator.
- The state of the object indicates the controller condition depending on the internal logic of the device.

Life Safety Point object applications can include the following safety related devices:

- Automatic fire detectors
- Sirens
- Pull stations

See also *Life Safety Zone object* on page 354.

General Properties

Object Instance The Life Safety Point object number. The objects are numbered sequentially within a BACnet device.

Object Name A text label of the object. The exact length will depend upon the controller that contains the object. The Name property must be unique within the BACnet device that maintains it. The set of characters entered for the Name property must be printable characters.

Present Value This property represents the state of the object. The state will be one of the 24 BACnet states for Life Safety Point objects. The internal logic of controller sets the state of present value. Present value may remain latched in a non-normal state until reset.

Profile Name The name the vendor's object profile to which this object conforms. A profile name defines a set of additional properties, behaviors and/or requirements for this object. The definition of the profile is vendor unique and not part of the BACnet standard.

Description A text label of the object. The exact length of the Description property will depend upon the controller that contains the object.

Device Type A text description of the physical device that the Life Safety Point object represents.

Silenced Indicates that the most recent transition that produced an audible or visual alarm has been silenced by an operator.

Operation Expected Indicates the next operation expected for the object to handle.

Accepted Modes A read-only list of the modes that can be written to the mode property by a BACnet service.

Mode

This property sets the object to one of 15 BACnet defined operating states. Changing the mode property may also trigger an alarm. Mode can be changed only to a value in the Accepted Modes list.

Out Of Service The Out Of service property indicates that the physical input is internally disconnected from the Life Safety Point object. KMC Connect sets this property to ***True (1)*** when selected and ***False*** (0) when unselected. When Out Of Service is selected, and sent to the controller, Present Value does not respond to changes at the physical input of the device.

Units Select a unit of measure to associate with the direct reading property.

Tracking Value This property represents the unlatched state of the object. Unlike the Present Value property, the Tracking Value property will not latch. The state will be one of the 24 BACnet states for Life Safety Point objects. The internal logic of controller sets the state of the Tracking Value property.

Setting Sets a threshold that determines the logical state of the Present Value property. The Setting property is a range from 0 (least sensitive) to 100 (most sensitive). The actual meaning to the setting value is determined and is unique to the controller that contains the Life Safety Point object.

Maintenance Required An indication of required maintenance for the device that is represented by the Life Safety Point object.

Direct Reading The analog value of the measured or calculated reading from the device that is represented by the Life Safety Point object.

Member Of Life Safety Zones A list of the Life Safety Zone objects that include the Life Safety Point object as a member.

To add an object to the Member Of list do either of the following:

- Locate the Life Safety Zone object in the Network Manager list and drag it to the Member list.
- Choose the object number from the Object ID list.

To remove an object from the Member Of list click the remove button  .

Event/Alarm Properties

Notification Class Specifies the notification class object to be used when handling and generating event notifications for this object. See ***Notification object*** on page 365 for details about the notification class object.

Notify Type This property specifies whether the notifications generated by the object are cvents or alarms. Alarm and event notifications are handled differently by the device—usually a workstation—that receives the notification.

Life Safety Alarm Values

To add a value, do the following:

1. Right-click in the Life Safety Alarm Values list and and choose **Add Value** from the shortcut menu.
2. Choose an alarm value from the the **New Value** list.

To delete a value, do the following:

1. Right-click a value in the list.
2. Choose **Remove Value** from the shortcut menu.

Time Delay Time Delay defines a minimum period, in seconds, for a set of conditions to exist before a To Off Normal or To Normal event occurs. Use Time Delay with Alarm Value to define conditions that indicate Present Value has changed states.

Event Enable

Use **Event Enable** to enable notifications for **To Off Normal**, **To Normal** and **To Fault**.

Alarm Values When the Present Value property equals one of the items in the Alarm Values property a notification event is generated. Items are added or deleted from the list by right-clicking and choosing either Add or Remove from the shortcut menu.

Fault Values When the Present Value property equals one of the items in the Fault Values property a fault event is generated. Items are added or deleted from the list by right-clicking and choosing either Add or Remove from the shortcut menu.

Status Properties

Status Flags The four BACnet status flags are an indication of the general condition of the object.

Table 30-11 Life Safety Point object status flags

Flag	Description
IN ALARM	False (0) if the event state property is Normal, otherwise True (1)
FAULT	True (1) if Reliability is present and the value for Reliability is not NO FAULT DETECTED, otherwise False (0)
OVERRIDDEN	True (1) if the point has been overridden by some mechanism local to the BACnet device. When this flag is True, Present Value and Reliability are no longer tracking changes to the physical input. Otherwise, the flag is False (0).
OUT OF SERVICE	True (1) if Out Of Service is selected. Otherwise False(0).

Reliability This property is an indication that Present Value or the operation of the physical input may be not be reliable as defined by the device or an operator. Reliability can have any of the following values:

- NO_FAULT_DETECTED
- SHORTED_LOOP
- UNRELIABLE_OTHER
- NO_SENSOR
- OPEN_LOOP

Event State Use **Event State** to determine that this input object has an active event state associated with it.

- If the input supports intrinsic reporting, then **Event State** indicates the state of the object.
- If the input does not support intrinsic reporting, then the value of this property is **Normal**.
- If **Reliability** is present and does not have a value of **No Fault Detected**, then the value of **Event State** is **Fault**. Changes in **Event State** to the value **Fault** are considered to be fault events.

Acked Transitions This property controls three flags that separately indicate the receipt of acknowledgments for **To Off Normal**, **To Fault**, and **To Normal** events. These flags are cleared upon the occurrence of the corresponding event and set under any of the following conditions:

This property controls three flags that separately indicate the receipt of acknowledgments for **To Off Normal**, **To Fault**, and **To Normal** events. These flags are cleared upon the occurrence of the corresponding event and set under any of the following conditions:

- Upon receipt of the corresponding acknowledgment
- Upon the occurrence of the event if the corresponding flag is not set in the **Event Enable** property. Under this condition, event notifications are not generated for this condition and therefore no acknowledgments are expected.
- Upon the occurrence of the event if the corresponding flag is set in the **Event Enable** property and the corresponding flag in the **Ack Required** property of the Notification Class object implicitly referenced by the Notification Class property of this object is not set (meaning no acknowledgment is expected.

Event Time Stamps Holds the times of the last event notifications of To Off Normal, To Fault and To Normal events.

Life Safety Zone object

A Life Safety Zone object is a standard BACnet object that represents the characteristics of an arbitrary group of Life Safety Point and Life Safety Zone objects that are typically used for fire, life safety, or security applications. The condition of the Life Safety Zone object is determined by the mode and state of the object.

- The mode of the object mode is usually under the control of the device operator.
- The state of the object indicates the controller condition depending on the internal logic of the device.

See also *Life Safety Point object* on page 351.

General Properties

Object Instance The Life Safety Zone object number. The objects are numbered sequentially within a BACnet device.

Object Name A text label of the object. The exact length will depend upon the controller that contains the object. The Name property must be unique within the BACnet device that maintains it. The set of characters entered for the Name property must be printable characters.

Present Value This property represents the state of the object. The state will be one of the 24 BACnet states for Life Safety objects. The internal logic of controller sets the state of present value. Present value may remain latched in a non-normal state until reset.

Profile Name The name the vendor's object profile to which this object conforms. A profile name defines a set of additional properties, behaviors and/or requirements for this object. The definition of the profile is vendor unique and not part of the BACnet standard.

Description A text label of the object. The exact length of the Description property will depend upon the controller that contains the object.

Silenced Indicates that the most recent transition that produced an audible or visual alarm has been silenced by an operator.

Operation Expected Indicates the next operation expected for the object to handle.

Accepted Modes A read-only list of the modes that can be written to the mode property by a BACnet service.

Mode This property sets the object to one of 15 BACnet defined operating states. Changing the mode property may also trigger an alarm. Mode can be changed only to a value in the Accepted Modes list.

Out Of Service The Out Of Service property indicates that the physical input is internally disconnected from the Life Safety Point object. KMC Connect sets this property to ***True (1)*** when selected and ***False*** (0) when unselected. When Out Of Service is selected, and sent to the controller, Present Value does not respond to changes at the physical input of the device.

Tracking Value This property represents the unlatched state of the object. Unlike the Present Value property, the Tracking Value property will not latch. The state will be one of the 24 BACnet states for Life Safety Point objects. The internal logic of controller sets the state of the Tracking Value property.

Maintenance Required An indication of required maintenance for the device that is represented by the Life Safety Point object.

Zone Members A list of the Life Safety Point and other Life Safety Zone objects that are a member of this Life Safety Zone object.

To add an object to the Member Of list do either of the following:

- Locate the Life Safety Zone object in the Network Manager list and drag it to the Zone Member list.
- Choose the object number from the Object ID list

To remove an object from the Member Of list click the remove button  .

Member of Life Safety Zone A list of the Life Safety Zone objects that include this Life Safety Zone object as a member.

To add an object to the Member Of list do either of the following:

- Locate the Life Safety Zone object in the Network Manager list and drag it to the Member Of list.
- Choose the object number from the Object ID list

To remove an object from Member Of list click the remove button  .

Event/Alarm Properties

Notification Class Specifies the notification class object to be used when handling and generating event notifications for this object. See *Notification object* on page 365 for details about the notification class object.

Notify Type This property specifies whether the notifications generated by the object are *Events* or *Alarms*. Alarm and event notifications are handled differently by the device—usually a workstation—that receives the notification.

Life Safety Alarm Values

To add a value, do the following:

1. Right-click in the Life Safety Alarm Values list and choose **Add Value** from the shortcut menu.
2. Choose an alarm value from the **New Value** list.

To delete a value,

1. Right-click a value in the list.
2. Choose **Remove Value** from the shortcut menu.

Time Delay Time Delay defines a minimum period, in seconds, for a set of conditions to exist before a **To Off Normal** or **To Normal** event occurs. Use **Time Delay** with **Alarm Value** to define conditions that indicate Present Value has changed states.

Event Enable Use **Event Enable** to enable notifications for **To Off Normal**, **To Normal** and **To Fault**.

Alarm Values When the Present Value property equals one of the items in the Alarm Values property a notification event is generated. Items are added or deleted from the list by right-clicking and choosing either Add or Remove from the shortcut menu.

Fault Values When the Present Value property equals one of the items in the Fault Values property a fault event is generated. Items are added or deleted from the list by right- clicking and choosing either Add or Remove from the shortcut menu.

Status Properties

Status Flags The four BACnet status flags are an indication of the general condition of the object.

Table 30-12 Life Safety Point object status flags

Flag	Description
IN ALARM	False (0) if the event state property is Normal, otherwise True (1)
FAULT	True (1) if Reliability is present and the value for Reliability is not NO FAULT DETECTED, otherwise False (0)
OVERRIDDEN	True (1) if the point has been overridden by some mechanism local to the BACnet device. When this flag is True, Present Value and Reliability are no longer tracking changes to the physical input. Otherwise, the flag is False (0).
OUT OF SERVICE	True (1) if Out Of Service is selected. Otherwise False(0).

Reliability This property is an indication that Present Value or the operation of the physical input may be not be reliable as defined by the device or an operator. Reliability can have any of the following values:

- NO_FAULT_DETECTED
- SHORTED_LOOP
- UNRELIABLE_OTHER
- NO_SENSOR
- OPEN_LOOP

Event State Use **Event State** to determine that this input object has an active event state associated with it.

- If the input supports intrinsic reporting, then **Event State** indicates the state of the object.
- If the input does not support intrinsic reporting, then the value of this property is **Normal**.
- If **Reliability** is present and does not have a value of **No Fault Detected**, then the value of **Event State** is **Fault**. Changes in **Event State** to the value **Fault** are considered to be fault events.

Acked Transitions This property controls three flags that separately indicate the receipt of acknowledgments for **To Off Normal**, **To Fault**, and **To Normal** events. These flags are

cleared upon the occurrence of the corresponding event and set under any of the following conditions:

- Upon receipt of the corresponding acknowledgment
- Upon the occurrence of the event if the corresponding flag is not set in the **Event Enable** property. Under this condition, event notifications are not generated for this condition and therefore no acknowledgments are expected.
- Upon the occurrence of the event if the corresponding flag is set in the **Event Enable** property and the corresponding flag in the **Ack Required** property of the Notification Class object implicitly referenced by the Notification Class property of this object is not set (meaning no acknowledgment is expected.

Event Time Stamps Holds the times of the last event notifications of To Off Normal, To Fault and To Normal events.

Load Control objects

A load control object represents a mechanism for controlling load requirements through load shedding. One or more objects may be used in the device to allow independent control over different sub-loads. The Load Control object may also be used in a hierarchical fashion to control other Load Control objects in other BACnet devices.

Note: Much of the actions in a Load Control object are defined by the device that contains the Load Control object and are not described here.

General Properties

Object Instance The Load Control object number. The objects are numbered sequentially within a BACnet device.

Object Name A text label of the object. The exact length will depend upon the controller that contains the object. The Name property must be unique within the BACnet device that maintains it. The set of characters entered for the Name property must be printable characters.

Present Value The present value of a load control object indicates the current load shedding state of the object. The states for the property are:

- **Shed Inactive**—The Load Control object is waiting for a shed request.
- **Shed Request Pending**— The object is determining, based on newly written shed request parameters, whether a shed request needs to be executed immediately or at some time in the future.
- **Shed Non Compliant**—The object is attempting to meet a shed request and will do so until the shed request is achieved, the object is reconfigured, or the request has completed unsuccessfully.
- **Shed Compliant**—The object is continuing to meet a shed request until the shed request is either reconfigured, finished, or the conditions change and the object is no longer able to maintain the requested shed level.

Profile Name The name the vendor's object profile to which this object conforms. A profile name defines a set of additional properties, behaviors and/or requirements for this object. The definition of the profile is vendor unique and not part of the BACnet standard.

Description A text label of the object. The exact length of the Description property will depend upon the controller that contains the object.

Shed Levels This property represents the shed levels when the Requested Shed Level property is set to Level. When commanded with the Level choice, the Load Control object starts the shedding action described by the corresponding element in the Shed Level Descriptions list.

Requested Shed Level This property indicates the desired load shedding. The following table describes the default values and power targets for the different choices of Requested Shed Level.

Table 30-13 Requested Shed Level Default Values

Choice	Default Requested Level	Power load target in kW
PERCENT	100	(Current baseline) * Requested Shed Level/ 100
LEVEL	0	Locally pre-specified shed target for the given level
AMOUNT	0.0	(Current baseline) - Requested Shed Level

Enable If the Enabled check box is selected, the Load Control object can respond to load shed requests. If the Enable check box is clear, the object is set to Shed Inactive and will not respond to requests to shed a load.

Start Time Enter the time of the duty window in which the load controlled by the object must be compliant with the shed request. If no shed request is pending or active, the Start Time property will contain all wildcard values. Setting Start Time earlier than the time maintained in the controller will start an immediate shed request.

State Description A text description that provides additional information for human operators about the shed state of the Load Control object. The exact length of the State Description property will depend upon the controller that contains the object. The set of characters entered for the property must be printable characters.

Shed Duration (minutes) The Shed Duration property indicates the duration of the load shedding action. The action will begin at the time specified in the Start Time property. The units for the Shed Duration property is in minutes. If no shed request is pending or active, or if the device has run a load control command to completion, the value for this property is zero.

Duty Window (Minutes) The Duty Window property indicates the time window—in minutes—for performance measurement or compliance purposes. The average power consumption across a duty window must be less than or equal to the requested reduced consumption.

Full Duty Baseline Indicates the baseline power consumption value for the load controlled by this Load Control object. Requests are made with reference to this baseline. For example as a percent of baseline or amount of baseline.

Shed Level Descriptions This property is a list of text descriptions for each of the shed levels that the Load Control object can take on. This allows a local configuration tool to provide to a user an understanding of what each shed level in this object's load shedding algorithm will do. The level at which each shed action will occur can then be configured by writing to the Shed Levels property.

Expected Shed Level This property indicates the amount of power that the object expects to be able to shed in response to a load shed request.

Actual Shed Level This property indicates the actual amount of power being shed in response to a load shed request. The units for the Actual Shed Level property are the same as the units for the Requested Shed Level.

Event/Alarm Properties

Notification Class Specifies the notification class object to be used when handling and generating event notifications for this object. See *Notification object* on page 365 for details about the notification class object.

Notification Type This property specifies whether the notifications generated by the object are events or alarms. Alarm and event notifications are handled differently by the device—usually a workstation—that receives the notification.

Time Delay This property specifies the minimum period—in seconds—hat the Present Value property must remain equal to Shed Non Compliant when the current time is later than Start Time and before a To Off Normal event is generated, or not equal to Shed Non Compliant before a To Normal event is generated. This property is required if intrinsic reporting is supported by this object.

Event Enable Use the Event Enable property to enable notifications for To Off Normal, To Normal, or To Fault conditions.

Status Properties

Status Flags The four BACnet status flags are an indication of the general condition of the object.

Table 30-14 Load Control object status flags

Flag	Description
IN ALARM	False (0) if the event state property is Normal, other wise True (1)
FAULT	True (1) if Reliability is present and the value for Reliability is not NO FAULT DETECTED, otherwise False (0)
OVERRIDDEN	True (1) if the point has been overridden by some mechanism local to the BACnet device. When this flag is True, Present Value and Reliability are no longer tracking changes to the physical input. Otherwise, the flag is False (0).
OUT OF SERVICE	This flag is always False(0) in a Load Control Object.

Reliability This property is an indication that the Load Control object is reliably reporting its compliance with load shed requests. Reliability can have either of the following values:

- No Fault Detected
- Unreliable other

Event State Use Event State to determine that this object has an active event state associated with it.

- If the object supports intrinsic reporting, then Event State indicates the state of the object.
- If the object does not support intrinsic reporting, then the value of this property is Normal.

Acked Transitions This property controls three flags that separately indicate the receipt of acknowledgments for To Off Normal, To Fault, and To Normal events.

Event Time Stamps This property holds the times of the last event notifications of *TO OFF NORMAL, TO FAULT,* and *TO NORMAL* events.

Alarm Routing Properties

Select the KMC Connect alarm class routing from the list.

Loop objects

A loop object provides access to a control loop. Use the loop objects dialog to manage the PID controller loops in the connected controller. A PID controller is a mathematical function which calculates the analog output required to maintain a process at or near a setpoint. The output of the loop object directly controls a property—usually present value—of an object.

To program the properties of loop object with Control Basic, see *Programming with mnemonics* on page 174.

General Properties

Object Instance The loop object number. Loop objects are numbered sequentially within the KMC BACnet device.

Object Name A text label of the object. The exact length will depend upon the controller that contains the object. The Name property must be unique within the BACnet device that maintains it. The set of characters entered for the Name property must be printable characters.

Present Value A numerical property that indicates the current value—in engineering units—of the output of the loop object.

Profile Name The name the vendor's object profile to which this object conforms. A profile name defines a set of additional properties, behaviors and/or requirements for this object. The definition of the profile is vendor unique and not part of the BACnet standard.

Description A text label of the object. The exact length of the Description property will depend upon the controller that contains the object.

Manipulated Variable Reference (Output)

Use the manipulated variable to direct the output of the loop to an object.

- Object selects the object to receive the output of the loop. Select the object from the Object list or drag the object from the Network Manager list.
- Property selects the property—usually Present Value–within the referenced object.

Update Interval Indicates the interval that the loop algorithm is updating Present Value of the loop object.

Priority For Writing Sets the priority for writing to the object in Manipulated Variable Reference. See the topic *Priority arrays* on page 292.

Maximum Output This property sets the maximum allowable value of the loop's present value property. It is normally used to prevent the loop algorithm from controlling beyond the range of the controlled device.

Minimum Output This property sets the minimum allowable value of the loop's present value property. It is normally used to prevent the loop algorithm from controlling beyond the range of the controlled device.

Controlled Variable Reference (Input) The controlled variable is usually the present value of an object or value object that is measuring a process variable such as a temperature. In a typical application, an analog input or analog value object represents a space temperature that is to be to maintained at a setpoint.

- **Object** selects the object that represents the controlled process variable. Select the object from the Object list or drag the object from the Network Manager list.
- **Property** selects the property—usually Present Value–within the referenced object.

Controlled Variable This property holds the value of the object in Controlled Variable Reference.

- **Value** is the value of the designated object Controlled Variable Units
- **Units** holds the engineering units of the selected property in Controlled Variable Reference.

Out Of Service When Out Of Service check box is selected, the output of the loop object is not updating the object selected by the Manipulated property. Select the Out Of Service check box to manually change Present Value.

Output Units Use Output Units to select the unit of measure for the output signal. KMC Connect supports several units of measure for loop objects. The available units are listed in the section *Supported engineering units* on page 417.

COV Increment This property specifies the minimum change of Present Value that will send a COV notification to subscriber notification clients.

Action The action of the loop. Action can be either direct acting or reverse acting.

- Direct acting loop objects increase the value of Manipulated Variable as the value of Controlled rises above the value of Setpoint.
- Reverse acting loop objects decrease the value of Manipulated Variable as the value of Controlled rises above the value of Setpoint.

Setpoint Reference (Setpoint Object) Use Setpoint Reference to designate where the setpoint for the loop is stored.

- If *None* is selected from the Object list, the loop setpoint is the fixed value that entered in the Setpoint property .
- If an object is selected from the Object list, the loop's setpoint is held in a property of the selected object.
- The setpoint property is typically the Present Value of the selected object but other properties can be selected from the Property list.

Proportional Constant and Units Proportional Constant is the value of the proportional gain parameter used by the loop algorithm. It represents the amount of sensed change—expressed in the value selected in Proportional Units—that will cause the output to move from 0 to 100%.

Integral Constant and Units Integral Constant is the value of the integral parameter—expressed in hours or minutes as selected in Integral Units— used by the loop algorithm. Integral adds a correction factor to the control loop based on how long the condition has been outside the setpoint. It specifies the number of times the magnitude of the error is added or subtracted to the output signal, over time, to eliminate the offset.

Derivative Constants and Units Derivative—specified in minutes—slows the rate of change of the error. Use Derivative to reduce overshoot. If the error is changing at 1.0 per second (60/min) and the rate was 0.25 minutes then the derivative component would equal 60 /Min X 0.25 Min =15%. This 15% would be added in over the 1 minute in a direction to reduce the rate of changing regardless of whether the input is above or below the setpoint.

Caution

Use the Derivative property only in systems without time lags. The input must start responding immediately to an output change. If there is a time delay the control loop will be unstable and will perform better without rate correction.

Bias Sets the output value at setpoint. The bias is the value the controller will reach at equilibrium when derivative is not used.

Event/Alarm Properties

Related topics

Notification Class Specifies the notification class object to be used when handling and generating event notifications for this object. See *Notification object* on page 365 for details about the notification class object.

Notification Type This property specifies whether the notifications generated by the object are events or alarms. Alarm and event notifications are handled differently by the device—usually a workstation—that receives the notification.

Error Limit This property sets the absolute magnitude that the difference between the values of Setpoint and Controlled Variable must exceed before a To Off Normal notification event is generated.

Deadband This property specifies a range between the high limit and low limit properties in which Present Value must remain before the device generates a To Normal notification event.

Time Delay Time Delay defines a minimum period for a set of conditions to exist before a To Off Normal or To Normal notification event occurs. Use Time Delay with Error Limit to indicate that the difference between Setpoint and Controlled is out of an expected, predefined operating range. Time Delay is expressed in seconds.

Event Enable Use the Event Enable property to enable notifications for To Off Normal, To Normal, or To Fault conditions.

Status Properties

Status Flags The four BACnet status flags are an indication of the general condition of the object.

Table 30-15 Loop object status flags

Flag	Description
IN ALARM	False (0) if the event state property is Normal, otherwise True (1)
FAULT	True (1) if Reliability is present and the value for Reliability is not NO FAULT DETECTED, otherwise False (0)
OVERRIDDEN	True (1) if the point has been overridden by some mechanism local to the BACnet device. When this flag is True, Present Value and Reliability are no longer tracking changes to the physical input. Otherwise, the flag is False (0).
OUT OF SERVICE	True (1) if Out Of Service is selected. Otherwise False(0).

Reliability This property is an indication that the Present Value property or the operation of the physical input may be not be reliable as defined by the device or an operator. Reliability can have any of the following values:

- NO_FAULT_DETECTED
- NO_SENSOR
- OPEN_LOOP
- SHORTED_LOOP
- UNRELIABLE_OTHER

Event State Use Event State to determine that this object has an active event state associated with it.

- If the object supports intrinsic reporting, then Event State indicates the state of the object.

- If the object does not support intrinsic reporting, then the value of this property is Normal.

If Reliability is present and does not have a value of No_fault_detected, then the value of the Event State property is Fault. Changes in the Event State to the value Fault are considered to be fault events.

Acked Transitions This property controls three flags that separately indicate the receipt of acknowledgments for To Off Normal, To Fault, and To Normal events.

Event Time Stamps This property holds the times of the last event notifications of *TO OFF NORMAL, TO FAULT,* and *TO NORMAL* events.

Alarm Routing Properties

Select the KMC Connect alarm class routing from the list.

Notification object

Use the notification class object to manage the distribution and processing of alarms and events originating within a device. The notification object:

- Maintains a list of destination devices which are usually BACnet operator workstations
- Sets the prioritization of To Off Normal and To Normal events by the destination device
- Designates if the event notification requires an acknowledgment
- Designates the process a recipient device should perform upon the receipt of an event.

Object Instance The notification class object number. Notification class objects are numbered sequentially within the BACnet device.

Object Name A text label of the object. The exact length will depend upon the controller that contains the object. The Name property must be unique within the BACnet device that maintains it. The set of characters entered for the Name property must be printable characters.

Profile Name The name the vendor's object profile to which this object conforms. A profile name defines a set of additional properties, behaviors and/or requirements for this object. The definition of the profile is vendor unique and not part of the BACnet standard.

Description A text label of the object. The exact length of the Description property will depend upon the controller that contains the object.

Recipient List The notification object recipient list may include up to four devices—each with a specific day and time—may be designated as recipients of the event.

- **Valid Days**—The days of the week on which this destination may be used between From Time and To Time.
- **From Time/To Time**—The window of time (inclusive) during which the destination is viable on the days of the week checked in Valid Days.
- **Transitions**—A set of flags that indicate To Off Normal, To Fault or To Normal for which the recipient is suitable.

- **Send Confirmed Notifications**—Check when confirmed notifications are to be sent. Leave unchecked if confirmed notifications are not required.
- **Process Identifier**—The handle of a process within the recipient device that is to receive the event notification.

Add and Remove To add a device to the recipient list, use the Add and Remove buttons. Devices can be added either by their device instance number or their network and MAC address.

Notification Class This property indicates the numeric value of this notification class and is equal to the instance number of the notification class object. Event-initiating objects use this number to indirectly refer to this notification class object.

Ack. Required Select the condition or conditions for which acknowledgment is required.

Notification Priorities Choose the priority for To Off Normal, To Fault, or To Normal event notifications. The highest priority is 0; the lowest is 255.

Table 30-16 Alarm and event priority

Alarm and event priority	Network priority
00-63	Life safety message
64-127	Critical equipment message
128-191	Urgent message
192-255	Normal message

Output objects–analog

An analog output is a standard BACnet object. The properties in the object set the characteristics of an analog signal at the output of a controller. Use the output object tool to configure each of the controller's outputs for either of the two output object types.

Related topics

- *BACnet objects and properties* on page 291
- *Priority arrays* on page 292
- *Output objects–binary* on page 370

General Properties

Output Instance

The output object number. Output objects are numbered sequentially within the KMC BACnet device and directly correspond to the controller's output terminal.

Object Name

A 16-character label of the object. Name must be unique within the BACnet device that maintains it. The set of characters entered for Name must be printable characters.

Present Value

This is a numerical property that indicates the current value—in engineering units—of the output terminal of the device.

Write Priority When saving the object, this property sets priority for the Present Value property. See the topic *Priority arrays* on page 292.

Profile Name The name the vendor's object profile to which this object conforms. A profile name defines a set of additional properties, behaviors and/or requirements for this object. The definition of the profile is vendor unique and not part of the BACnet standard.

Description A text label of the object. The exact length of the Description property will depend upon the controller that contains the object.

Device Type

This property is a description of the physical device connected to the output. Choose from an available type in the drop-down list.

Out Of Service

Out Of Service indicates that the physical output is internally disconnected from the output object. This property will be True when selected or False when not selected. When Out Of Service is selected, Present Value does not change the value at the output terminal of the controller.

Relinquish Default Sets the status or value that will take effect when all levels of the priority array are NULL. See *Priority arrays* on page 292.

Units

Select a unit of measure to associate with the output signal. The available units are listed in the section *Supported engineering units* on page 417. KMC Connect supports several units of measure for both analog and binary outputs. For binary outputs, the first unit in the pair of units is the Normal Inactive state of the output. See *Output objects–binary* on page 370 for the relationship between Units and Polarity property.

Minimum Present Value

This property indicates the smallest value—expressed in engineering units—to which the output can reliably be set.

Maximum Present Value

This property indicates the largest value—expressed in engineering units—to which the output can reliably be set.

COV Increment

This property specifies the minimum change of Present Value that will send a COV notification to subscriber notification clients.

Resolution

This property indicates the smallest recognizable change—expressed in engineering units—in the present value of this object.

Event/Alarm Properties

Notification Class Specifies the notification class object to be used when handling and generating event notifications for this object. See ***Notification object*** on page 365 for details about the notification class object.

Notification Type This property specifies whether the notifications generated by the object are events or alarms. Alarm and event notifications are handled differently by the device—usually a workstation—that receives the notification.

High Limit

This property is used with intrinsic reporting to define an upper limit for a normal operating range of Present Value. Use with Limit Enable, Deadband and Time Delay to define conditions that indicate Present Value is out of a normal operating range.

Low Limit

This property is used with intrinsic reporting to define a lower limit for a normal operating range of Present Value. Use with Limit Enable, Deadband and Delay to define events that indicate Present Value is out of a normal operating range.

Deadband

This property specifies a range between the high limit and low limit properties in which Present Value must remain before the device generates a To Normal notification event.

Time Delay

The Time Delay property defines a minimum period for a set of conditions to exist before a To Off Normal or To Normal notification event occurs. Use Time Delay with High Limit, Low Limit and Deadband to define conditions that indicate Present Value is out of an expected, predefined operating range. Time Delay is expressed in seconds.

Limit Enable

This property separately enables and disables reporting of high limit and low limit **Off Normal** events and their return to normal.

Event Enable Use the Event Enable property to enable notifications for To Off Normal, To Normal, or To Fault conditions.

Status Properties

Status Flags The four BACnet status flags are an indication of the general condition of the object.

Table 30-17 Output object status flags

Flag	Description
IN ALARM	False (0) if the event state property is Normal, other wise True (1)
FAULT	True (1) if Reliability is present and the value for Reliability is not NO FAULT DETECTED, otherwise False (0)

Output object status flags (continued)

Flag	Description
OVERRIDDEN	True (1) if the point has been overridden by some mechanism local to the BACnet device. When this flag is True, Present Value and Reliability are no longer tracking changes to the physical output. Otherwise, the flag is False (0).
OUT OF SERVICE	True (1) if Out Of Service is selected. Otherwise False(0).

Reliability This property is an indication that the Present Value property or the operation of the physical input may be not be reliable as defined by the device or an operator. Reliability can have any of the following values:

- NO_FAULT_DETECTED
- UNDER_RANGE
- UNRELIABLE_OTHER
- NO_SENSOR
- OPEN_LOOP
- OVER_RANGE
- SHORTED_LOOP

Event State Use Event State to determine that this object has an active event state associated with it.

- If the object supports intrinsic reporting, then Event State indicates the state of the object.
- If the object does not support intrinsic reporting, then the value of this property is Normal.

If Reliability is present and does not have a value of No_fault_detected, then the value of Event_State is Fault. Changes in the Event State to the value Fault are considered to be fault events.

Acked Transitions This property controls three flags that separately indicate the receipt of acknowledgments for To Off Normal, To Fault, and To Normal events.

Event Time Stamps This property holds the times of the last event notifications of *TO OFF NORMAL, TO FAULT,* and *TO NORMAL* events.

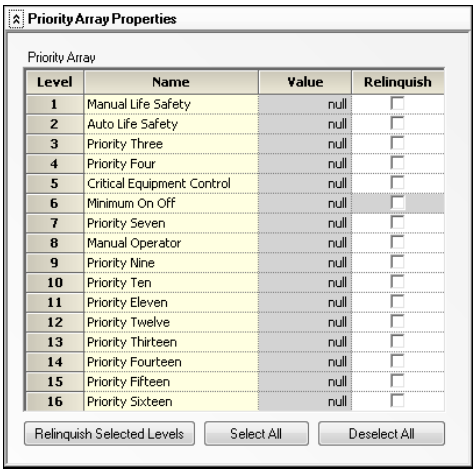
Priority Array Properties

Displays the priority level for the object.

To relinquish a priority value, select the check box and then click **Relinquish**.

See the related topic *Priority arrays* on page 292.

Illustration 30–2 Priority Array Properties



Alarm Routing Properties

Select the KMC Connect alarm class routing from the list.

Output objects–binary

A binary output is a standard BACnet object. The properties in the object set the characteristics of an binary signal at the output of a controller. Use the output object tool to configure each of the controller's outputs for either of the two output object types.

Related topics

- *BACnet objects and properties* on page 291
- *Output objects–analog* on page 366.
- *Priority arrays* on page 292

General Properties

Object Instance The output object number. Output objects are numbered sequentially within the KMC BACnet device and directly correspond to the controller's output terminal.

Object Name A text label of the object. The exact length will depend upon the controller that contains the object. The Name property must be unique within the BACnet device that maintains it. The set of characters entered for the Name property must be printable characters.

Present Value This property reflects the logical state of an output that is either Inactive or Active. The relationship between Present Value and the physical state of the output is determined by the polarity property. The possible states are summarized in the table *Output object polarity relationships* on page 371.

To manually change the present value property, enter the new value and then press the enter key or click another property. A dialog opens in which the write priority level is selected. See *Priority arrays* on page 292.

Write Priority When saving the object, this property sets priority for the Present Value property. See the topic *Priority arrays* on page 292.

Profile Name The name the vendor's object profile to which this object conforms. A profile name defines a set of additional properties, behaviors and/or requirements for this object. The definition of the profile is vendor unique and not part of the BACnet standard.

Description A text label of the object. The exact length of the Description property will depend upon the controller that contains the object.

Device Type This property is a description of the physical device connected to the output. Choose from an available type in the drop-down list.

Out Of Service The Out Of Service property indicates that the physical output is internally disconnected from the output object. This property will be True when selected or False when not selected. When Out Of Service is selected, Present Value does not change the value at the output terminal of the controller.

Relinquish Default Sets the status or value that will take effect when all levels of the priority array are NULL. See *Priority arrays* on page 292.

Polarity The polarity property sets the relationship between the physical state of the output and the logical state represented by Present Value. KMC Connect displays Present value as either Active Text or Inactive Text.

Table 30-18 Output object polarity relationships

Polarity	Voltage at output	Text displayed	Example
Normal	0	Inactive Text	Off, Stop
Normal	10	Active Text	On, Start
Reverse	0	Active Text	On, Start
Reverse	10	Inactive Text	Off, Stop

Active Text Text to display when the output is in the active state.

Inactive Text Text to display when the output is in the inactive state.

Minimum On Time Sets the minimum time—expressed in seconds—that Present Value must remain in the active state after changing to the active state.

Minimum Off Time Sets the minimum time—expressed in seconds—that Present Value must remain in the inactive state after changing to the inactive state.

Event/Alarm Properties

Notification Class Specifies the notification class object to be used when handling and generating event notifications for this object. See *Notification object* on page 365 for details about the notification class object.

Notification Type This property specifies whether the notifications generated by the object are events or alarms. Alarm and event notifications are handled differently by the device—usually a workstation—that receives the notification.

Feedback Value This property is used with intrinsic reporting to indicate the value from which Present Value must differ to generate a notification.

Time Delay Time Delay defines a minimum period for a set of conditions to exist before a To Off Normal or To Normal event occurs. Use Time Delay with Feedback Value to define conditions that indicate Present Value has changed states. Time Delay is expressed in seconds.

Event Enable Use the Event Enable property to enable notifications for To Off Normal, To Normal, or To Fault conditions.

Relinquish Default Sets the status or value that will take effect when all levels of the priority array are NULL. See *Priority arrays* on page 292.

Status Properties

Status Flags The four BACnet status flags are an indication of the general condition of the object.

Table 30-19 Binary output object status flags

Flag	Description
IN ALARM	False (0) if the event state property is Normal, other wise True (1)
FAULT	True (1) if Reliability is present and the value for Reliability is not NO FAULT DETECTED, otherwise False (0)
OVERRIDDEN	True (1) if the point has been overridden by some mechanism local to the BACnet device. When this flag is True, Present Value and Reliability are no longer tracking changes to the physical output. Otherwise, the flag is False (0).
OUT OF SERVICE	True (1) if Out Of Service is selected. Otherwise False(0).

Reliability This property is an indication that the Present Value property or the operation of the physical input may be not be reliable as defined by the device or an operator. Reliability can have any of the following values:

- NO_FAULT_DETECTED
- SHORTED_LOOP
- NO_SENSOR
- UNRELIABLE_OTHER
- OPEN_LOOP

Time of Active Time Reset This property holds the date and time that Elapsed Active Time was most recent set to zero.

Elapsed Active Time This proper holds the accumulated number of seconds that Present Value has been in the active state since Elapsed Active Time was set to zero.

Event State Use Event State to determine that this object has an active event state associated with it.

- If the object supports intrinsic reporting, then Event State indicates the state of the object.

- If the object does not support intrinsic reporting, then the value of this property is Normal.

If Reliability is present and does not have a value of No Fault Detected, then the value of Event State is Fault. Changes in the Event State to the value Fault are considered to be fault events.

Acked Transitions This property controls three flags that separately indicate the receipt of acknowledgments for To Off Normal, To Fault, and To Normal events.

Event Time Stamps This property holds the times of the last event notifications of *TO OFF NORMAL, TO FAULT,* and *TO NORMAL* events.

Change of State Time This property holds the date and time that Present Value most recently changed state.

Change of State Count This property holds the number of times that Present Value has changed state since Change of State Count was most recently set to zero (0).

Time of State Count Reset This property holds the date and time that Change of Sate Count was most recent set to zero (0).

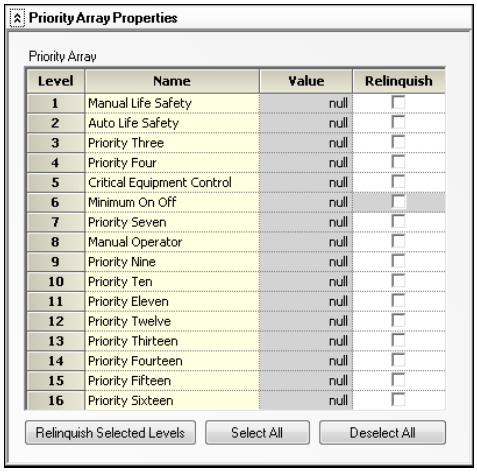
Priority Array Properties

Displays the priority level for the object.

To relinquish a priority value, select the check box and then click **Relinquish**.

See the related topic *Priority arrays* on page 292.

Illustration 30–3 Priority Array Properties



Alarm Routing Properties

Select the KMC Connect alarm class routing from the list.

Output object–multistate

The Multistate Output object is a standard object whose properties represent the state of one or more physical outputs or processes within the BACnet Device in which the object resides. For example, a particular state may represent the active or inactive condition of

several physical outputs or the value of an analog output. The result of the calculation is a single number that represents the state of the object.

Related topics

- *BACnet objects and properties* on page 291
- *Priority arrays* on page 292
- *Output objects–analog* on page 366
- *Output objects–binary* on page 370

General Properties

Object Instance The number of the object. Output objects are numbered sequentially within a BACnet device.

Object Name A text label of the object. The exact length will depend upon the controller that contains the object. The Name property must be unique within the BACnet device that maintains it. The set of characters entered for the Name property must be printable characters.

Present Value This numerical property indicates the current state of the Mulitstate Output object. The number of states for this property is defined by the value of the Number Of States property.

Write Priority When saving the object, this property sets priority for the Present Value property. See the topic *Priority arrays* on page 292.

Profile Name The name the vendor's object profile to which this object conforms. A profile name defines a set of additional properties, behaviors and/or requirements for this object. The definition of the profile is vendor unique and not part of the BACnet standard.

Description A text label of the object. The exact length of the Description property will depend upon the controller that contains the object.

Device Type A text description that is typically used to describe the type of device attached to the Multistate Output object.

Out Of Service Out Of Service indicates that the physical output is internally disconnected from the output object. This property will be True(1) when selected or False(0) when not selected. When Out Of Service is selected, Present Value does not change the value at the output terminal of the controller.

Relinquish Default Sets the status or value that will take effect when all levels of the priority array are NULL. See *Priority arrays* on page 292.

Number of States The Number of States property defines the number of states that Present Value may have. It is always greater than zero. The value of Number of States will automatically change as entries are added or removed from the State Text list.

State Text Each entry in the State Text list corresponds to a value of Present Value. For example, if Present Value equals 1 then the value of State Text is the first entry in the list.

- To start a new entry in State Text, right-click and choose **Add String** from the shortcut menu.

- To change the text of an entry in State Text, right-click an entry and choose **Edit String** from the shortcut menu.
- To delete an entry from State Text, right-click an entry and choose **Remove String** from the shortcut menu.
- To change the order of the list in State Text, right-click an entry and choose **Move Up** or **Move Down** from the shortcut menu.

Event/Alarm Properties

Notification Class Specifies the notification class object to be used when handling and generating event notifications for this object. See *Notification object* on page 365 for details about the notification class object.

Notification Type This property specifies whether the notifications generated by the object are events or alarms. Alarm and event notifications are handled differently by the device—usually a workstation—that receives the notification.

Feedback Value This property is used with intrinsic reporting to indicate the value from which Present Value must differ before a To Off Normal notification is generated.

Time Delay Time Delay defines a minimum period for a set of conditions to exist before a To Off Normal or To Normal event occurs. Use Time Delay with Alarm Value to define conditions that indicate Present Value has changed state. Time Delay is expressed in seconds.

Event Enable Use the Event Enable property to enable notifications for To Off Normal, To Normal, or To Fault conditions.

Status Properties

Status Flags The four BACnet status flags are an indication of the general condition of the object.

Table 30-20 Multistate output object status flags

Flag	Description
IN ALARM	False (0) if the event state property is Normal, other wise True (1)
FAULT	True (1) if Reliability is present and the value for Reliability is not NO FAULT DETECTED, otherwise False (0)
OVERRIDDEN	True (1) if the point has been overridden by some mechanism local to the BACnet device. When this flag is True, Present Value and Reliability are no longer tracking changes to the physical input. Otherwise, the flag is False (0).
OUT OF SERVICE	True (1) if Out Of Service is selected. Otherwise False(0).

Reliability This property is an indication that the Present Value property or the operation of the physical input may be not be reliable as defined by the device or an operator. Reliability can have any of the following values:

- No Fault Detected
- Open Loop
- Shorted Loop
- No Output
- Communication Failure
- Unreliable Other

Event State Use Event State to determine that this object has an active event state associated with it.

- If the object supports intrinsic reporting, then Event State indicates the state of the object.
- If the object does not support intrinsic reporting, then the value of this property is Normal.

Acked Transitions This property controls three flags that separately indicate the receipt of acknowledgments for To Off Normal, To Fault, and To Normal events.

Event Time Stamps This property holds the times of the last event notifications of *TO OFF NORMAL, TO FAULT,* and *TO NORMAL* events.

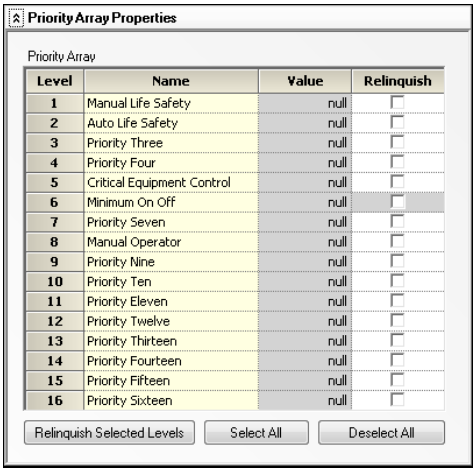
Priority Array Properties

Displays the priority level for the object.

To relinquish a priority value, select the check box and then click **Relinquish**.

See the related topic *Priority arrays* on page 292.

Illustration 30–4 Priority Array Properties



Alarm Routing Properties

Select the KMC Connect alarm class routing from the list.

Program object

The program object provides a network-visible view of selected parameters of an application program in the form of properties. The form and content of the application program is unique to each BACnet vendor. In KMC BACnet controllers, the program within a program object is Control Basic. Control Basic programs are edited with the KMC Connect code editor.

Related topics

- *About Control Basic programs* on page 165
- *Programming with the Code Editor* on page 151

Object Instance The number of the program object within the device.

Object Name A text label of the object. The exact length will depend upon the controller that contains the object. The Name property must be unique within the BACnet device that maintains it. The set of characters entered for the Name property must be printable characters.

Profile Name The name the vendor's object profile to which this object conforms. A profile name defines a set of additional properties, behaviors and/or requirements for this object. The definition of the profile is vendor unique and not part of the BACnet standard.

Description A text label of the object. The exact length of the Description property will depend upon the controller that contains the object.

Instance Of This property is the local name of the program in the program object.

Program Location This property is used by the application program to indicate its location within the program code. The exact method is determined by the vendor.

Reliability The reliability property is an indication of whether the program is valid as evaluated by the device in which the program is running. The reliability property for program objects may have any of the following values:

- NO FAULT DETECTED
- PROCESS ERROR
- UNRELIABLE OTHER

Out Of Service Out Of Service indicates that the program is properly loaded and initialized, although the process may or may not be actually executing. This property will be True when selected or False when not selected. When Out Of Service is selected, State is Idle.

Run On Coldstart When selected, Control Basic in the program object will automatically start after either a warm start or cold start.

Program State This property reflects the current state of the program within the device.

Table 30-21 Control Basic program states

State	Description of state
IDLE	The program is not executing
LOADING	The program is being loaded.

Control Basic program states (continued)

State	Description of state
RUNNING	The program is currently executing.
WAITING	The program is waiting for some external event.
HALTED	The program is halted because of some error condition.
UNLOADING	The program has been requested to terminate.

Program Change Use Change Program to alter the operational state of a program. The device may also change the state of Program Change as a result of program execution.

Table 30-22 Program object program changes

Change	Description of change
READY	The program is ready for a change request. This is the normal state of the object.
LOAD	Requests that the program be loaded, if it is not already loaded
RUN	Request that the program begin executing, if not already running
HALT	Request that the program halt execution.
RESTART	Request that the process restart at its initialization point
UNLOAD	Request that execution halts and the program unloads.

Reason For Halt If the program is stopped for any reason, Reason For Halt displays an explanation of the halt.

Table 30-23 Control Basic reason for halt

State	Description of state
NORMAL	The Control Basic program has not stopped because of any error condition.
LOAD_FAILED	The Control Basic program could not complete loading.
INTERNAL	The Control Basic program halted because of some internal mechanism.
PROGRAM	The Control Basic program was halted by a program change request.
OTHER	The Control Basic program is halted for some other reason.

Description for Halt This property is a character string that may be used to describe the reason why a program stopped running. If implemented in a device, this property provides a more detailed description than found in Reason For Halt property

Status Flags These flags are associated with values of other properties within the program object. A more detailed status can be determined by reading the properties that are linked to these flags.

Table 30-24 Program status flags

Flag	Description
In Alarm	Unselected in a program object
Fault	Not selected when the reliability property is present and does not have a value of *No Fault Detected*. Selected when the reliability property has any other value.
Overridden	Selected when the program has been overridden by some mechanism within the BACnet device. When Overriddens is selected, neither *Program Change*, *Program State* nor any other program-specific property may be changed through BACnet services. Otherwise, the value is not selected.
Out Of Service	Selected if *Out Of Service* is selected, otherwise it is not selected.

Schedule object

BACnet schedule objects directly control the state of one or more BACnet objects based on the times and values entered into either a weekly schedule or an exception schedule.

General Properties

Object Instance The schedule object number within the device. Schedules are numbered sequentially within BACnet devices.

Object Name A text label of the object. The exact length will depend upon the controller that contains the object. The Name property must be unique within the BACnet device that maintains it. The set of characters entered for the Name property must be printable characters.

Present Value This property indicates the value most recently written to an object property of the objects in the Object Reference List. The value may be from either a *Weekly Schedule* on page 381 or one of the *Exception Schedules*.

Profile Name The name the vendor's object profile to which this object conforms. A profile name defines a set of additional properties, behaviors and/or requirements for this object. The definition of the profile is vendor unique and not part of the BACnet standard.

Description A text label of the object. The exact length of the Description property will depend upon the controller that contains the object.

Effective Period Set the active period of the schedule with Start Time and End Time. Create seasonal schedules by defining several schedules with non-overlapping Start and End Time periods to control the same assigned objects.

Out Of Service When Out Of Service is selected and sent to the controller, Present Value does not respond when a weekly schedule changes state.

Priority for Writing Sets the BACnet priority for writing of assigned objects. See *Priority arrays* on page 292.

Schedule Default Each day in a weekly schedule covers a 24–hour period. Schedule Default defines the value and data type of the schedule's present value between 12:00 A.M. (midnight) and the first time slot in the weekly schedule. The present value of the schedule remains at the value of the last time slot until 12:00 A.M.

Object Reference List This is a list of the objects to be controlled by the schedule. A single schedule controls only one type of object.

- Device Instance—The instance number and mnemonic of the device that contains the controlled object. An entry of zero indicates the object is in the same device as the schedule.
- Object ID—The object that will be controlled by the schedule.
- Property ID—The property that is under control of the schedule. Present value is the default.

To add an object to the Object Reference list do either of the following:

- Locate object in the Network Manager list and drag it to the Object Reference List.
- Enter the Object ID in the Object Reference List

To remove an object from Object Reference List click the remove button  .

Status Properties

Status Flags The four BACnet status flags are an indication of the general condition of the object.

Table 30-25 Schedule object status flags

Flag	Description
In Alarm	*FALSE* (0) if the event state property is *NORMAL*, other wise *TRUE* (1)
Fault	True (1) if Reliability is present and the value for Reliability is not *No Fault Detected*, otherwise False (0)
OVERRIDDEN	*TRUE* (1) if the point has been overridden by some mechanism local to the BACnet device. When this flag is *TRUE*, *PRESENT VALUE*, and *RELIABILITY* are no longer tracking changes to the physical schedule. Otherwise, the flag is *FALSE* (0).
Out Of Service	*TRUE* (1) if *OUT OF SERVICE* is selected. Otherwise *FALSE* (0).

Reliability This property is an indication that the Present Value property or the operation of the physical input may be not be reliable as defined by the device or an operator. Reliability can have any of the following values:

- NO FAULT DETECTED
- UNRELIABLE OTHER
- CONFIGURATION ERROR

Weekly Schedule

The weekly schedule consists of several sets of time-value pairs for each day of the week. As the day and time maintained within the controller becomes equal to a time and day in the schedule, the present value of the objects in the Object Reference List is set to the value of the time-value pair. When the current time and date are within the date range of the schedule ***and*** the day and time of the weekly schedule, the value associated with the day of the week and time in Weekly Schedule is assigned to Present Value in the assigned object.

- The schedule for the days in a normal week are defined by the weekly schedule.
- Days on the schedule that require a schedule different from the normal days are defined by exception schedules.
- A single schedule typically controls either analog, binary, or multistate objects but not a mix of objects.
- For KMC BACnet devices, reference objects must be within the same device as the schedule object.

Data Type Choose from one of the four data types to set the default data type for the schedule default data type, weekly schedules and exception schedules.

Caution

Changing the data type deletes all schedule times from the Schedule object. Set the data type—either in the Weekly Schedule area or at the top of the tab before entering times and values into a schedule.

The available data types are available for weekly schedules.

- Boolean
- Real
- Enumerated
- Unsigned

The default data type can also be set by choosing a new **Change Schedule Data Type** list at the top of the Schedule object tab.

- Default defines the state of the schedule's present value and data type for the period between 12:00 midnight and the first time-value pair each day.
- The present value of the schedule will remain at the value of the last time-value pair until 12:00 midnight.
- Time-value pairs with Null as the value are ignored.
- Duplicate times are not permitted in the same day.

To add a time-value pair

To add a time value pair, do either of the following:

- Click the add button  at the bottom of the list of time-value pairs. The new pair is added to the bottom of the list.
- Drag the remove button  next to a time-value to any day in the schedule.

To change time in a time-value pair

1. To change time, select the hours, minute or seconds value.
2. Click the up or down arrows  next to the time value.

To remove a time-value pair

To remove any time-value pair click the remove button  .

Exception Schedules

Use an exception schedule to override the values in the weekly schedule.

To add an exception schedule to the object, click the Add button and then choose the data type and type of type of exception.

The **Exception Schedule** dialog includes the following choices for setting dates.

Data Type Choose from one of the four data types to set the data type for the exception schedule. The available data types for available for exception schedules.

- Boolean
- Real
- Enumerated
- Unsigned.

Date A single date on which the values and times listed in the exception schedule will override the values of the weekly schedule.

Date Range A range of dates on which the values and times listed in the exception schedule will override the values of the weekly schedule.

- If End Date is empty, then all dates beginning with Start Date are considered valid dates in the range of dates.
- If Start Date is empty then all dates from the current system date up to End Date are considered to be valid dates in the range of dates.

Week and Day A day of the week and month on which the values and times listed in the exception schedule will override the values in the weekly schedule.

Calendar Object Use a calendar object for special days such as holidays or other dates that require special attention on a schedule. If the date in the calendar object is true then the exception schedule will override the weekly schedule. To configure a Calendar Object, see the topic *Calendar object* on page 321.

Event priority Sets the order of precedence for conflicting exceptions schedules. For example if Exception Schedule 2 has an event priority of 18 and Exception Schedule 4 has an event priority of 10 then Exception Schedule 2 will override Exception Schedule 4 when there is a conflict in the values for the reference object.

Time and Value Set the time and values for the exception schedule the same way they are entered in the weekly schedule. See *Weekly Schedule* on page 381.

Calendar View

The Calendar view displays all scheduled times in daily, weekly, or monthly graphical format. The Month view starts on the calendar month that includes the date in Start Time.

Structured View objects

The Structured View object defines a standard object that holds references to subordinate objects. Subordinate objects may reside either in the same device as the Structured View object or in other devices on the network. A multilevel hierarchy of objects may be created by including other Structured View objects as subordinate objects.

Object Instance The Structured View object number within the device. The objects are numbered sequentially within the device.

Object Name A text label of the object. The exact length will depend upon the controller that contains the object. The Name property must be unique within the BACnet device that maintains it. The set of characters entered for the Name property must be printable characters.

Profile Name The name the vendor's object profile to which this object conforms. A profile name defines a set of additional properties, behaviors and/or requirements for this object. The definition of the profile is vendor unique and not part of the BACnet standard.

Description A text label of the object. The exact length of the Description property will depend upon the controller that contains the object.

Node Type This property provides a general classification of the object in the hierarchy of objects. The allowable values for this property are:

- UNKNOWN

- SYSTEM
- NETWORK
- DEVICE
- ORGANIZATIONAL
- AREA
- EQUIPMENT
- POINT
- COLLECTION
- PROPERTY
- FUNCTIONAL
- OTHER

Node Subtype This property is a short text description of the item represented by the node. It provides a more specific classification of the object in the hierarchy of objects.

Subordinate List This property is a list of the subordinate objects in Structured View object.

To add a subordinate object

To add an object to the Subordinate List property, do either of the following:

- Click the add button  and then and then enter a Device Instance and Object ID to the line added to the list.
- Drag an object from the Network Manager list to the Subordinate List.

To remove a subordinate object

To remove any object, click the remove button  next to the object's Device Instance.

Subordinate Annotations This optional property is text description for each member of the Subordinate List.

Trend Log object

Use Trend Log objects to monitor and record the value of a property. Each trend log periodically saves the data record along with a time stamp and relevant status information at the time the controller saved the record to a trend log.

Related topics

- *Configuring BACnet trend logs and groups* on page 89
- *Trend Log Multiple objects* on page 388

General Properties

Object Instance The Trend Log object number within the device. The objects are numbered sequentially within the device.

Object Name A text label of the object. The exact length will depend upon the controller that contains the object. The Name property must be unique within the BACnet device that maintains it. The set of characters entered for the Name property must be printable characters.

Profile Name The name the vendor's object profile to which this object conforms. A profile name defines a set of additional properties, behaviors and/or requirements for this object. The definition of the profile is vendor unique and not part of the BACnet standard.

Description A text label of the object. The exact length of the Description property will depend upon the controller that contains the object.

Start and Stop Date and Time Set the period for logging data with Start and Stop Time and Date.

Log Enable If the Log Enable check box is clear, the object does not log data. If the Log Enable check box is selected and the current time and date are within the range of time and dates specified by Start Time/Date and Stop Time/Date, the Trend Log object logs data.

Stop When Full When selected, once the buffer is full the Trend Log object stops adding new records to the log buffer. If the check box is clear, the oldest data record is replaced with a new record.

Trigger When the Logging Type property is set to Triggered and the Trigger property changes from False to True, the object will log a record from the monitored property. This property will remain True until all recording is complete. The object will then change this property to False.

Log Interval This property, specifies the periodic interval—in hundredths of seconds—for which the data from the monitored property is to be logged.

Buffer Size Holds the value of the maximum number of records the trend buffer may hold.

COV Resubscription Interval If the trend log is acquiring data from a remote device by COV subscription and COV subscription is in effect, this property specifies the number of seconds between COV resubscriptions.

If COV subscriptions are in effect, the first COV subscription is issued when the trend log object begins operation or when Log Enable becomes True. If present, the value of this property must be non-zero. If this property is not present, then COV subscription cannot be attempted.

Client COV Increment If the Trend Log object is acquiring COV data, this property specifies the increment to be used in determining that a change of value has occurred.

Notification Threshold When the number of records reaches the value of Notification Threshold, a notification is sent to the notification class object specified for this Trend Log object.

Log Object Property Reference Designate the device, object and property as the source of data for the trend log.

- Device–an object within a BACnet device on the internetwork. Choose Local to select the device within which the trend log is located.

- Object–Choose the object from the Object list or drag the object from the Network Manager list. The object may be an input, output, value or accumulator object. For KMC BACnet controllers the object must be within the BACnet device as the trend log.
- Property–Choose the property within the designated object. Typically the property is Present Value.

Align Intervals If this check box is selected, clock-aligned periodic logging is enabled. When periodic logging is enabled and the value of Log Interval is a factor of (that is, it divides without remainder) a second, minute, hour, or day, then the beginning of the period specified for logging is aligned to the second, minute, hour, or day, respectively.

Interval Offset This value specifies an interval, in hundredths of a second, between the beginning of the period specified for logging until the actual acquisition of the data record.

Logging Type Select whether the Trend Log object will collect records using polling or triggered acquisition.

Event/Alarm Properties

Notification Class Specifies the notification class object to be used when handling and generating event notifications for this object. See ***Notification object*** on page 365 for details about the notification class object.

Notification Type This property specifies whether the notifications generated by the object are events or alarms. Alarm and event notifications are handled differently by the device—usually a workstation—that receives the notification.

Event Enable Use the Event Enable property to enable notifications for To Off Normal, To Normal, or To Fault conditions.

Status Properties

Status Flags The four BACnet status flags are an indication of the general condition of the object.

Table 30-26 Trend Log object status flags

Flag	Description
IN ALARM	False (0) if the event state property is Normal, other wise True (1).
FAULT	True (1) if Reliability is present and the value for Reliability is not NO FAULT DETECTED, otherwise False (0).
OVERRIDDEN	This flag is always False(0) in a Trend Log object.
OUT OF SERVICE	This flag is always False(0) in a Trend Log object.

Reliability This property is an indication that the Present Value property or the operation of the physical input may be not be reliable as defined by the device or an operator. Reliability can have any of the following values:

- No Fault Detected
- Configuration Error
- Communication Failure
- Unreliable Other

Record Count This property displays the number of records currently resident in the log buffer. Writing a value zero to this property will delete all records in the log buffer and reset the Records Since Notification property to zero. Upon completion, this event will be entered in the log as the initial entry.

Total Record Count This property represents the total number of records collected by the Trend Log object since creation. When the value of property reaches its maximum value of 4,294,967,295, the next value is 1.

Records Since Last Notification This property represents the number of records collected since the previous notification, or since the beginning of logging if no previous notification has occurred. This property is required if intrinsic reporting is supported by this object.

Last Notify Record This property represents the sequence number associated with the most recently collected record whose collection triggered a notification. If no notification has occurred since logging began the value of this property is zero. This property is required if intrinsic reporting is supported by this object.

Previous Notify Time This property represents the value that the property Current Notify Time had at the time of the previous notification. At the beginning of a notification operation this property is set to the value of Current Notify Time, when the property Current Notify Time is updated. If no previous notification has occurred this property shall contain all wildcard values.

Current Notify Time This property represents the timestamp associated with the most recently collected record whose collection triggered a notification. If no notification has occurred since logging began this property contains all wildcard values.

Event State Use Event State to determine that this object has an active event state associated with it.

- If the object supports intrinsic reporting, then Event State indicates the state of the object.
- If the object does not support intrinsic reporting, then the value of this property is Normal.

If Reliability is present and does not have a value of No Fault Detected, then the value of Event State is Fault. Changes in the Event State to the value Fault are considered to be fault events.

Acked Transitions This property controls three flags that separately indicate the receipt of acknowledgments for To Off Normal, To Fault, and To Normal events. For analog objects, the transition of High Limit and Low Limit are considered to be Off Normal events. These flags are cleared upon the occurrence of the corresponding event and set under any of the following conditions:

- Upon receipt of the corresponding acknowledgment

- Upon the occurrence of the event if the corresponding flag is not set in Event Enable. Under this condition, event notifications are not generated for this condition and therefore no acknowledgments are expected.
- Upon the occurrence of the event if the corresponding flag is set in Event Enable and the corresponding flag in the Ack Required property of the notification class object implicitly referenced by the notification class property of this object is not set (meaning no acknowledgment is expected.).

Event Time Stamps This property holds the times of the last event notifications of *TO OFF NORMAL, TO FAULT,* and *TO NORMAL* events.

Alarm Routing Properties

Select the KMC Connect alarm class routing from the list.

View Trend

Choose View Trend to display the trend data stored in a controller.

Autoload When selected, new controller trend data is added to the data in the trend viewer.

Update Click to retrieve new data not already on the screen and then add it to the screen data.

Refresh Clears data from the screen, retrieves records from the device and then and plots the new data on the screen.

Custom Refresh Set the trend log to refresh on a specific date or when a specific number of records have been collected.

Trend Log Multiple objects

Use Trend Log Multiple objects to monitor one or more properties in one or more reference objects. The Trend Log Multiple object periodically saves the value of the monitored properties along with a timestamp and relevant status information at the time the object saved the data The monitored properties may either be in the same device as the Trend Log Multiple object or a different device.

Related topics

- *Configuring BACnet trend logs and groups* on page 89
- *Trend Log object* on page 384

General Properties

Object Instance The Trend Log Multiple object number within the device. The objects are numbered sequentially within the device.

Object Name A text label of the object. The exact length will depend upon the controller that contains the object. The Name property must be unique within the BACnet device that maintains it. The set of characters entered for the Name property must be printable characters.

Profile Name The name the vendor's object profile to which this object conforms. A profile name defines a set of additional properties, behaviors and/or requirements for this object. The definition of the profile is vendor unique and not part of the BACnet standard.

Description A text label of the object. The exact length of the Description property will depend upon the controller that contains the object.

Start and Stop Date and Time Set the period for logging data with Start and Stop Time and Date.

Log Enable If the Log Enable check box is clear, the object does not log data. If the Log Enable check box is selected and the current time and date are within the range of time and dates specified by Start Time/Date and Stop Time/Date the Trend Log Multiple object logs data.

Stop When Full When selected, once the buffer is full the Trend Log Multiple object stops adding new records to the log buffer. If the check box is clear, the oldest data record is replaced with a new record.

Trigger When the Logging Type property is set to Triggered and the Trigger property changes from False to True, the object will log a record from each of the monitored properties. The property will remain True until all recording is complete. The object will then change this property to False.

Log Interval This property, specifies the periodic interval—in hundredths of seconds—for which the data from Object to Log is to be logged.

Buffer Size Holds the value of the maximum number of records the trend buffer may hold.

COV Resubscription Interval If the trend log is acquiring data from a remote device by COV subscription and COV subscription is in effect, this property specifies the number of seconds between COV resubscriptions.

If COV subscriptions are in effect, the first COV subscription is issued when the trend log object begins operation or when Log Enable becomes True. If present, the value of this property must be non-zero. If this property is not present, then COV subscription cannot be attempted.

Client COV Increment If the trend log object is acquiring COV data, this property specifies the increment to be used in determining that a change of value has occurred.

Notification Threshold When the number of records reaches the value of Notification Threshold, a notification is sent to the Notification Class object specified for this trend.

Log Object Property Reference Designate the device, object, property, and data type as the source of data for the log object.

- Device Instance–Enter the device instance number for a BACnet device on the internetwork. Enter zero (0) to select the device within which the trend log is located.
- Object ID–Choose the object from the Object list or drag the object from the Network Manager list. For KMC BACnet controllers the object must be within the same BACnet device as the log object.
- Property ID–Choose the property within the designated object. Typically the property is Present Value.

- Data Type—Choose from one of the four data types to set the data type for the monitored property. The available data types are:
  - Boolean
  - Real
  - Enumerated
  - Unsigned.

To add a monitored object and property

To add an object and property to the Log Object Property List, do either of the following:

- Click the add button  and then and then enter a Device Instance, Object ID, Property ID, and Data Type to the line added to the list.
- Drag an object from the Network Manager list to the Log Object Property List.

To remove a monitored object and property

To remove any object, click the remove button  next to the object's Device Instance.

Align Intervals If this check box is selected, clock-aligned periodic logging is enabled. When periodic logging is enabled and the value of Log Interval is a factor of (that is, it divides without remainder) a second, minute, hour, or day, then the beginning of the period specified for logging is aligned to the second, minute, hour, or day, respectively.

Interval Offset This value specifies an interval, in hundredths of a second, between the beginning of the period specified for logging until the actual acquisition of the data record.

Logging Type Select whether the Trend Log Multiple object will collect records using polling or triggered acquisition.

Event/Alarm Properties

Notification Type This property specifies whether the notifications generated by the object are events or alarms. Alarm and event notifications are handled differently by the device—usually a workstation—that receives the notification.

Notification Class Specifies the notification class object to be used when handling and generating event notifications for this object. See ***Notification object*** on page 365 for details about the notification class object.

Event Enable Use the Event Enable property to enable notifications for To Off Normal, To Normal, or To Fault conditions.

Status Properties

Status Flags The four BACnet status flags are an indication of the general condition of the object.

Table 30-27 Trend Log Multiple object status flags

Flag	Description
IN ALARM	False (0) if the event state property is Normal, other wise True (1)
FAULT	True (1) if Reliability is present and the value for Reliability is not NO FAULT DETECTED, otherwise False (0)
OVERRIDDEN	This flag is always False(0) in a Trend Log Multiple object.
OUT OF SERVICE	This flag is always False(0) in a Trend Log Multiple object.

Reliability This property is an indication that the Present Value property or the operation of the physical input may be not be reliable as defined by the device or an operator. Reliability can have any of the following values:

- No Fault Detected
- Configuration Error
- Communication Failure
- Unreliable Other

Record Count This property displays the number of records currently resident in the log buffer. Writing a value zero to this property will delete all records in the log buffer and reset the Records Since Notification property to zero. Upon completion, this event will be entered in the log as the initial entry.

Total Record Count This property represents the total number of records collected by the Trend Log Multiple object since creation. When the value of property reaches its maximum value of 4,294,967,295, the next value is 1.

Records Since Last Notification This property represents the number of records collected since the previous notification, or since the beginning of logging if no previous notification has occurred. This property is required if intrinsic reporting is supported by this object.

Last Notify Record This property represents the sequence number associated with the most recently collected record whose collection triggered a notification. If no notification has occurred since logging began the value of this property is zero. This property is required if intrinsic reporting is supported by this object.

Previous Notify Time This property represents the value that the property Current Notify Time had at the time of the previous notification. At the beginning of a notification operation this property is set to the value of Current Notify Time, when the property Current Notify Time is updated. If no previous notification has occurred this property shall contain all wildcard values.

Current Notify Time This property represents the timestamp associated with the most recently collected record whose collection triggered a notification. If no notification has occurred since logging began this property contains all wildcard values.

Event State Use Event State to determine that this object has an active event state associated with it.

- If the object supports intrinsic reporting, then Event State indicates the state of the object.
- If the object does not support intrinsic reporting, then the value of this property is Normal.

If Reliability is present and does not have a value of No Fault Detected, then the value of Event State is Fault. Changes in the Event State to the value Fault are considered to be fault events.

Acked Transitions This property controls three flags that separately indicate the receipt of acknowledgments for To Off Normal, To Fault, and To Normal events.

Event Time Stamps This property holds the times of the last event notifications of *TO OFF NORMAL, TO FAULT,* and *TO NORMAL* events.

Alarm Routing Properties

Select the KMC Connect alarm class routing from the list.

View Trend

Choose **View Trend** to display the trend data stored in a controller.

Chart View and Table View Select one of these tabs to view the data in the device as either a line graph or a table.

Autoload When selected, new controller trend data is added to the data in the trend viewer.

Update Click to retrieve new data not already on the screen and then add it to the screen data.

Refresh Clears data from the screen, retrieves records from the device and then and plots the new data on the screen.

Custom Refresh Set the trend log to refresh on a specific date or when a specific number of records have been collected.

Value objects–analog

An analog value object is a standard BACnet object whose properties represent an analog value residing in memory. Use a value object to define program variables in Control Basic. Variables are place holders for information such as setpoints, time delays and modes.

Related topics

- *BACnet objects and properties* on page 291
- *Value objects–binary* on page 396
- *Value objects–multistate* on page 399

General Properties

Object Instance The value object number. Value objects are numbered sequentially within the KMC BACnet device.

Object Name A text label of the object. The exact length will depend upon the controller that contains the object. The Name property must be unique within the BACnet device that maintains it. The set of characters entered for the Name property must be printable characters.

Present Value This numerical property indicates the current value—in engineering units—of the value object. To manually change the present value property, enter the new value and then press the enter key or click another property. A dialog opens in which the write priority level is selected.

Write Priority When saving the object, this property sets priority for the Present Value property. See the topic *Priority arrays* on page 292.

Profile Name The name the vendor's object profile to which this object conforms. A profile name defines a set of additional properties, behaviors and/or requirements for this object. The definition of the profile is vendor unique and not part of the BACnet standard.

Description A text label of the object. The exact length of the Description property will depend upon the controller that contains the object.

Out Of Service When Out Of Service is checked, the present value cannot be updated by programs running on the controller. This property will be True when selected or False when not selected. Changes can still be made to the object's priority array.

Relinquish Default Sets the status or value that will take effect when all levels of the priority array are NULL. See *Priority arrays* on page 292.

Units Use Units to select the unit of measure for the value object. The available units are listed in the section *Supported engineering units* on page 417.

COV Increment This property specifies the minimum change of Present Value that will send a COV notification to subscriber notification clients.

Event/Alarm Properties

Notification Class Specifies the notification class object to be used when handling and generating event notifications for this object. See *Notification object* on page 365 for details about the notification class object.

Notification Type This property specifies whether the notifications generated by the object are events or alarms. Alarm and event notifications are handled differently by the device—usually a workstation—that receives the notification.

High Limit This property is used with intrinsic reporting to define an upper limit for a normal operating range of Present Value. Use with Limit Enable, Deadband and Delay to define conditions that indicate Present Value is out of a normal operating range.

Low Limit This property is used with intrinsic reporting to define a lower limit for a normal operating range of Present Value. Use with Limit Enable, Deadband and Delay to define events that indicate Present Value is out of a normal operating range.

Deadband This property specifies a range between the high limit and low limit properties in which Present Value must remain before the device generates a To Normal notification event.

Time Delay Time Delay defines a minimum period for a set of conditions to exist before a To Off Normal or To Normal notification event occurs. Use Time Delay with High Limit,

Low Limit and Deadband to define conditions that indicate Present Value is out of an expected, predefined operating range. Time Delay is expressed in seconds.

Limit Enable This property separately enables and disables reporting of high limit and low limit Off Normal events and their return to normal.

Event Enable Use the Event Enable property to enable notifications for To Off Normal, To Normal, or To Fault conditions.

Status Properties

Status Flags The four BACnet status flags are an indication of the general condition of the object.

Table 30-28 Analog value object status flags

Flag	Description
IN ALARM	False (0) if the event state property is Normal, other wise True (1)
FAULT	True (1) if Reliability is present and the value for Reliability is not No Fault Detected, otherwise False (0)
OVERRIDDEN	True (1) if the point has been overridden by some mechanism local to the BACnet device. Otherwise, the flag is False (0). When this flag is True, Present Value cannot be changed through BACnet services.
OUT OF SERVICE	True (1) if Out Of Service is selected. Otherwise False(0).

Reliability This property is an indication that Present Value or the operation of the object may be not be reliable as defined by the device or an operator. Reliability can have any of the following values:

- NO FAULT DETECTED
- UNDER RANGE
- UNRELIABLE OTHER
- OPEN LOOP
- NO SENSOR
- OVER RANGE
- SHORTED LOOP

Event State Use Event State to determine that this object has an active event state associated with it.

- If the object supports intrinsic reporting, then Event State indicates the state of the object.
- If the object does not support intrinsic reporting, then the value of this property is Normal.

If Reliability is present and does not have a value of No Fault Detected, then the value of Event State is Fault. Changes in the Event State property to the value Fault are considered to be fault events.

Acked Transitions This property controls three flags that separately indicate the receipt of acknowledgments for To Off Normal, To Fault, and To Normal events. For analog objects, the transition of High Limit and Low Limit are considered to be Off Normal events. These flags are cleared upon the occurrence of the corresponding event and set under any of the following conditions:

- Upon receipt of the corresponding acknowledgment
- Upon the occurrence of the event if the corresponding flag is not set in the Event Enable property. Under this condition, event notifications are not generated for this condition and therefore no acknowledgments are expected.

Upon the occurrence of the event if the corresponding flag is set in the Event Enable property and the corresponding flag in the Ack Required property of the notification class object implicitly referenced by the notification class property of this object is not set (meaning no acknowledgment is expected.).

Event Time Stamps This property holds the times of the last event notifications of *TO OFF NORMAL, TO FAULT,* and *TO NORMAL* events.

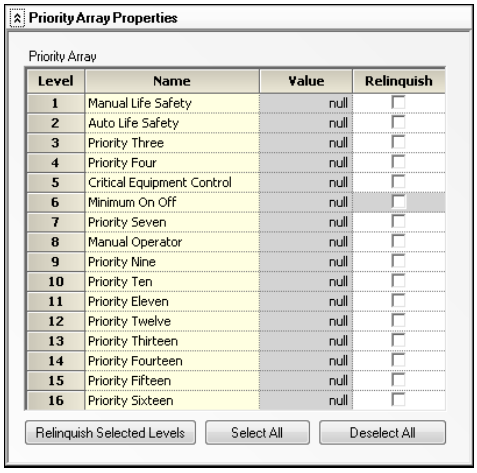
Priority Array Properties

Displays the priority level for the object.

To relinquish a priority value, select the check box and then click **Relinquish**.

See the related topic *Priority arrays* on page 292.

Illustration 30–5 Priority Array Properties



Alarm Routing Properties

Select the KMC Connect alarm class routing from the list.

Value objects–binary

An binary value object is a standard BACnet object whose properties represent an binary value residing in memory. Use a value object to define program variables in Control Basic. Variables are place holders for information such flags and conditions.

Related topics

- *BACnet objects and properties* on page 291
- *Value objects–analog* on page 392
- *Value objects–multistate* on page 399

General Properties

Object Instance The value object number. Value objects are numbered sequentially within the KMC BACnet device.

Object Name A text label of the object. The exact length will depend upon the controller that contains the object. The Name property must be unique within the BACnet device that maintains it. The set of characters entered for the Name property must be printable characters.

Present Value This numerical property indicates the current state—in engineering units—of the value object. To manually change the present value property, enter the new value and then press the enter key or click another property. A dialog opens in which the write priority level is selected.

Write Priority When saving the object, this property sets priority for the Present Value property. See the topic *Priority arrays* on page 292.

Profile Name The name the vendor's object profile to which this object conforms. A profile name defines a set of additional properties, behaviors and/or requirements for this object. The definition of the profile is vendor unique and not part of the BACnet standard.

Description A text label of the object. The exact length of the Description property will depend upon the controller that contains the object.

Out Of Service When Out Of Service is checked, the present value cannot be updated by programs running on the controller. This property will be True when checked or False when unchecked. Changes can still be made to the object's priority array.

Relinquish Default Sets the status or value that will take effect when all levels of the priority array are NULL. See *Priority arrays* on page 292.

Minimum Off Time Sets the minimum time—expressed in seconds—that Present Value must remain in the inactive state after changing to the inactive state.

Minimum On Time Sets the minimum time—expressed in seconds—that Present Value must remain in the active state after changing to the active state.

Active Text Text to display when the value object is in the active state.

Inactive Text Text to display when the value object is in the inactive state.

Event/Alarm Properties

Notification Class Specifies the notification class object to be used when handling and generating event notifications for this object. See *Notification object* on page 365 for details about the notification class object.

Notification Type This property specifies whether the notifications generated by the object are events or alarms. Alarm and event notifications are handled differently by the device—usually a workstation—that receives the notification.

Alarm Value This property is used with intrinsic reporting to define a change of Present Value that will generate an alarm event.

Time Delay The property Time Delay defines a minimum period for a set of conditions to exist before a To Off Normal or To Normal event occurs. Use Time Delay with Alarm Value to define conditions that indicate Present Value has changed state. Time Delay is expressed in seconds.

Event Enable Use the Event Enable property to enable notifications for To Off Normal, To Normal, or To Fault conditions.

Status Properties

Status Flags The four BACnet status flags are an indication of the general condition of the object.

Table 30-29 Binary value object status flags

Flag	Description
IN ALARM	False (0) if the event state property is Normal, other wise True (1)
FAULT	True (1) if Reliability is present and the value for Reliability is not No Fault Detected, otherwise False (0)
OVERRIDDEN	True (1) if the point has been overridden by some mechanism local to the BACnet device. Otherwise, the flag is False (0). When this flag is True, Present Value cannot be changed through BACnet services.
OUT OF SERVICE	True (1) if Out Of Service is selected. Otherwise False(0).

Reliability This property is an indication that Present Value or the operation of the object may be not be reliable as defined by the device or an operator. Reliability can have any of the following values:

- NO FAULT DETECTED
- UNRELIABLE OTHER

Time of Active Time Reset This property holds the date and time that Elapsed Active Time was most recently set to zero.

Elapsed Active Time This property holds the accumulated number of seconds that Present Value has been in the active state since Elapsed Active Time was set to zero.

Event State Use Event State to determine that this value object has an active event state associated with it.

- If the object supports intrinsic reporting, then Event State indicates the state of the object.
- If the object does not support intrinsic reporting, then the value of this property is Normal.
- If Reliability is present and does not have a value of No Fault Detected, then the value of Event State is Fault. Changes in Event State to the value Fault are considered to be fault events.

Acked Transitions This property controls three flags that separately indicate the receipt of acknowledgments for To Off Normal, To Fault, and To Normal events. These flags are cleared upon the occurrence of the corresponding event and set under any of the following conditions:

- Upon receipt of the corresponding acknowledgment
- Upon the occurrence of the event if the corresponding flag is not set in Event Enable. Under this condition, event notifications are not generated for this condition and therefore no acknowledgments are expected.
- Upon the occurrence of the event if the corresponding flag is set in Event Enable and the corresponding flag in Ack Required of the notification class object implicitly referenced by the notification class property of this object is not set (meaning no acknowledgment is expected).

Event Time Stamps This property holds the times of the last event notifications of *TO OFF NORMAL, TO FAULT,* and *TO NORMAL* events.

Change of State Time This property holds the date and time that Present Value most recently changed state.

Change of State Count This property holds the number of times that Present Value has changed state since Change of State Count was most recently set to zero.

Time of State Count Reset This property holds the date and time that Change of State Count was most recent set to zero.

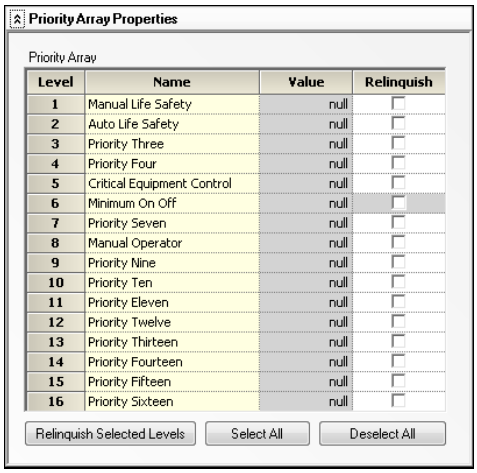
Priority Array Properties

Displays the priority level for the object.

To relinquish a priority value, select the check box and then click **Relinquish**.

See the related topic *Priority arrays* on page 292.

Illustration 30–6 Priority Array Properties



Alarm Routing Properties

Select the KMC Connect alarm class routing from the list.

Value objects–multistate

A multistate value object is a standard BACnet object whose properties represent the result of a calculation performed within the device. For multistate value objects in BACnet devices from KMC Controls the calculation is performed with Control Basic.

Related topics

- *BACnet objects and properties* on page 291
- *Value objects–analog* on page 392
- *Value objects–binary* on page 396

General Properties

Object Instance The multistate value object number. Multistate value objects are numbered sequentially within the KMC BACnet device.

Object Name A text label of the object. The exact length will depend upon the controller that contains the object. The Name property must be unique within the BACnet device that maintains it. The set of characters entered for the Name property must be printable characters.

Present Value This numerical property indicates the current state—in engineering units—of the value object. To manually change the present value property, enter the new value and then press **ENTER** or click another property.

Write Priority When saving the object, this property sets priority for the Present Value property. See the topic *Priority arrays* on page 292.

Profile Name The name the vendor's object profile to which this object conforms. A profile name defines a set of additional properties, behaviors and/or requirements for this object. The definition of the profile is vendor unique and not part of the BACnet standard.

Description A text label of the object. The exact length of the Description property will depend upon the controller that contains the object.

Out Of Service When Out Of Service is selected, the present value cannot be updated by programs running on the controller. This property will be True (1) when selected or False (0) when cleared. Changes can still be made to the object's priority array.

Relinquish Default Sets the status or value that will take effect when all levels of the priority array are NULL. See *Priority arrays* on page 292.

Number of States The Number of States property defines the number of states that Present Value may have. It is always greater than zero. The value of Number of States will automatically change as entries are added or removed from State Text.

State Text Each entry in the State Text list corresponds to a value of Present Value. For example, if Present Value equals 1 then the value of State Text is the first entry in the list.

- To start a new entry in State Text, right-click and choose Add String from the shortcut menu.
- To change the text of an entry in State Text, right-click an entry and choose Edit String from the shortcut menu.
- To delete an entry from State Text, right-click an entry and choose Remove String from the shortcut menu.
- To change the order of the list in State Text, right-click an entry and choose Move Up or Move Down from the shortcut menu.

Event/Alarm Properties

Notification Class Specifies the notification class object to be used when handling and generating event notifications for this object. See *Notification object* on page 365 for details about the notification class object.

Notification Type This property specifies whether the notifications generated by the object are events or alarms. Alarm and event notifications are handled differently by the device—usually a workstation—that receives the notification.

Time Delay Time Delay defines a minimum period for a set of conditions to exist before a To Off Normal or To Normal event occurs. Use Time Delay with Alarm Value to define conditions that indicate Present Value has changed state. Time Delay is expressed in seconds.

Event Enable Use the Event Enable property to enable notifications for To Off Normal, To Normal, or To Fault conditions.

Alarm Value This property is used with intrinsic reporting to define a change of Present Value that will generate an alarm event.

Fault Values This property indicates any states that the Present Value property must equal before a To Fault notification is generated.

Status Properties

Status Flags The four BACnet status flags are an indication of the general condition of the object.

Table 30-30 Multistate Value object status flags

Flag	Description
IN ALARM	False (0) if the event state property is Normal, other wise True (1)
FAULT	True (1) if Reliability is present and the value for Reliability is not No Fault Detected, otherwise False (0)
OVERRIDDEN	True (1) if the point has been overridden by some mechanism local to the BACnet device. Otherwise, the flag is False (0). When this flag is True, Present Value cannot be changed through BACnet services.
OUT OF SERVICE	True (1) if Out Of Service is selected. Otherwise False(0).

Reliability This property is an indication that the Present Value property or the operation of the physical input may be not be reliable as defined by the device or an operator. Reliability can have any of the following values:

- NO_FAULT_DETECTED
- MULTI_STATE_FAULT
- UNRELIABLE_OTHER

Event State Use Event_State to determine that this value object has an active event state associated with it.

- If the object supports intrinsic reporting, then Event State indicates the state of the object.
- If the object does not support intrinsic reporting, then the value of this property is Normal.
- If Reliability is present and does not have a value of No Fault Detected, then the value of Event State is Fault. Changes in Event_State to the value Fault are considered to be fault events.

Acked Transitions This property controls three flags that separately indicate the receipt of acknowledgments for To Off Normal, To Fault, and To Normal events. These flags are cleared upon the occurrence of the corresponding event and set under any of the following conditions:

- Upon receipt of the corresponding acknowledgment
- Upon the occurrence of the event if the corresponding flag is not set in Event Enable. Under this condition, event notifications are not generated for this condition and therefore no acknowledgments are expected.
- Upon the occurrence of the event if the corresponding flag is set in Event Enable and the corresponding flag in Ack Required of the notification class object implicitly referenced by the notification class property of this object is not set (meaning no acknowledgment is expected).

Event Time Stamps This property holds the times of the last event notifications of *TO OFF NORMAL, TO FAULT,* and *TO NORMAL* events.

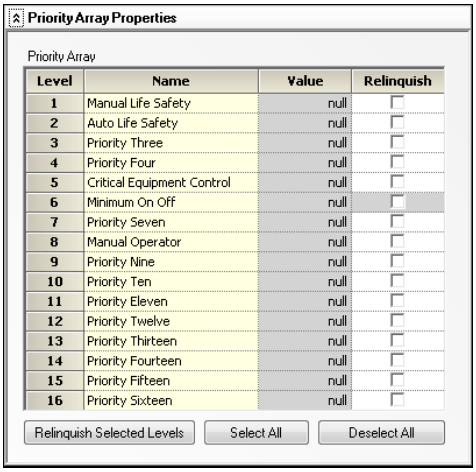
Priority Array Properties

Displays the priority level for the object.

To relinquish a priority value, select the check box and then click **Relinquish**.

See the related topic *Priority arrays* on page 292.

Illustration 30–7 Priority Array Properties



Alarm Routing Properties

Select the KMC Connect alarm class routing from the list.

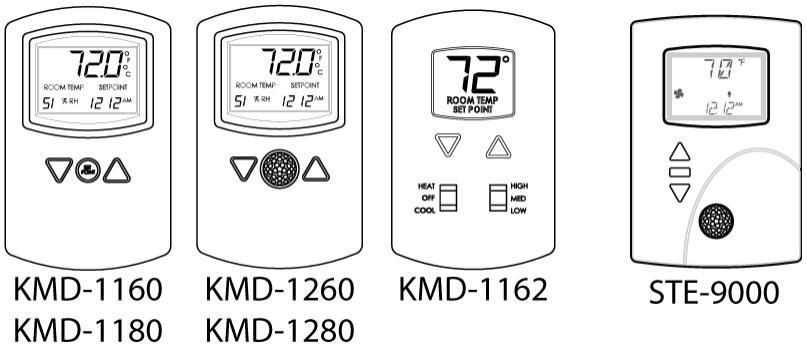
Section 31: Configuring NetSensors

This section covers the configuration of KMD series and STE-9000 series of NetSensors.

The two types of NetSensors have unique setup procedures. Both procedures are covered in the following topics.

- **KMD-1160 and KMD-1180 series** NetSensors are used with the BAC-5800 and BAC-7000 series controllers. Configuration is described in the topic *KMD series NetSensors for BACnet* on page 404.
- **STE-9000 series** NetSensors are used with the BAC-5900 and BAC-9000 series of controllers. Configuration is described in the topic *Configuring the STE-9000 series NetSensor* on page 409.

Illustration 31–1 NetSensor families



KMD series NetSensors for BACnet

The KMD-1160, KMD-1180, KMD-1260, and KMD-1280 series NetSensors are wall mounted display and sensor unit that connects directly to a KMC Controls BACnet controller. The unit consists of an LCD display, a thermistor, up to seven buttons, and optional humidity and motion sensors.

- For methods to program the NetSensor, see the procedures *Programming the NetSensor* on page 406.
- To setup an STE-9000 series NetSensor, see the topic *Configuring the STE-9000 series NetSensor* on page 409.
- To open a NetSensor configuration page, see the topic, *Configuring BACnet devices and objects* on page 300.

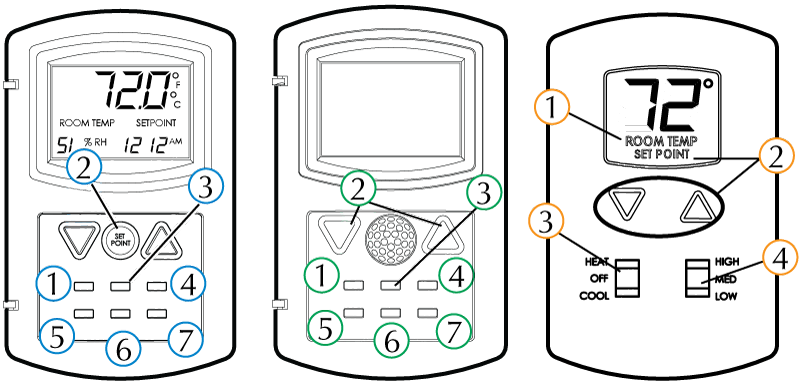
NetSensor buttons

The functions of the NetSensor buttons are listed in the following table.

Table 31-1 NetSensor buttons

Button	Function
Up arrow	Increases displayed analog values and toggles binary values
Down arrow	Decreases displayed analog values and toggles binary values
Button 1	The default display value. Only Button 1 may be assigned to the internal temperature sensor.
Button 2	Labeled as ***Setpoint*** but can be defined as an input, output or value object within the controller.
Buttons 3–6	Can be defined as an input, output or value object within the controller.
Button 7	On applicable models, assigned as the humidity sensor physically included on the NetSensor.
Aux	Press buttons 5 and 7 together and then press the up arrow or down arrow button to change ***Aux*** from ***Off*** to ***On***.

Illustration 31–2 NetSensor buttons



Assignment Points to the value object that stores the button value. The value object can be either an analog or binary object. The object can be selected from the Assignment list or by dragging an object from the Network Manager list to Assignment.

Current Value This numerical property indicates the current value—in engineering units—of the input being measured.

Calibration (Button 1 only) Enter a value to correct the temperatures reading as compared to a calibrated test instrument.

- For a low temperature reading enter a positive correction value
- For a high temperature reading enter a negative correction value.
- The maximum calibration is 3.2 degrees Fahrenheit above or below the displayed value.

Display Format Use *Display Format* to select one of the units from the drop-down list.

- Units in the table *NetSensor analog display formats* are active when the object under *Assignment* is an analog input, output or value object.
- Units in the table *NetSensor binary display formats* are active when a binary input, output or value object is selected under *Assignment*.

Table 31-2 NetSensor analog display formats

Unit	Action and display
°C	Displays temperature in degrees Celsius. Available only on Button 1. If °C is selected, Button 1 is assigned to the internal temperature sensor and must be associated with a value object.
°F	Displays temperature in degrees Fahrenheit. Available only on Button 1. If °F is selected, Button 1 is assigned to the internal temperature sensor and must be associated with a value object.
0	Sets the precision of the display to nearest whole number.
0.0	Sets the precision of the display to one place to the right of the decimal point.
0.00	Sets the precision of display to two places to the right of the decimal point.
Time	Sets the NetSensor to display a time format.
Off/Low/High	The NetSensor cycles through each word as arrow buttons are pressed and released. The analog value object cycles from 0 to 2.
Off/On1/2/3	The NetSensor display cycles through each word as arrow buttons are pressed and released. The analog value object cycles from 0 to 3.
System Time	Use to set time in a stand-alone controller when an operator workstation or other time master device is not available.
Day Of Week	Use to set the day of week in a stand-alone controller when an operator workstation or other time master device is not available.

Table 31-3 NetSensor binary display formats

Label	Action and display
On/Off	The NetSensor toggles between words as arrow buttons are pressed and released. The binary object toggles between 0 and 1.
Low/High	The NetSensor toggles between words as arrow buttons are pressed and released. The binary object toggles between 0 and 1.
Cool/Heat	The NetSensor toggles between words as arrow buttons are pressed and released.The binary object toggles between 0 and 1.

Permission The permission property sets the button to be either a display-only button or a button that an operator can use to change a value in the building automation system.

- *Read Only* indicates an operator may only view the data displayed on the NetSensor.
- *Read/Write* indicates an operator may use the arrow buttons to change the data.

Retain Changes When selected, the present value of the object associated with a button is written to the flash memory in a KMC BACnet controller and retrieved from the controller after a cold start.

Auto Blanking When this check box is selected, the NetSensor display will go dark approximately 15 seconds after the last button is pushed.

Programming the NetSensor

Button 1–internal temperature sensor Button 1 and the space temperature sensor inside of the NetSensor are usually associated with analog value object AV1.

- In *Assignment*, select an analog value object to associate with the temperature button or drag an object from the Network Manager list to *Assignment*.
- Set *Display Format* to *°F* or *°C*.
- Set *Permission* to *Read Only*. Operators can then view the room temperature by touching Button 1 but cannot change the value.

Button 1–controller points To manage or display an object value from Button 1, configure as follows:

- In *Assignment*, select an input, output or value object to associate with the temperature button or drag an object from the Network Manager list to *Assignment*.
- Set *Display Format* to *0, 0.0* or *0.00*.
- Set *Permission* to *Read Only* or *Read/Write*.
  - If set to *Read Only*, operators can view the value associated with the Button but cannot change it.
  - Is set to *Read/Write* operators can change the value of the object by first touching Button 1 and then pressing the up or down arrow buttons.

Setpoint Button 2 is usually associated with analog value object AV2.

- In ***Assignment***, select an analog value object to associate with the setpoint button or drag an object from the Network Manager list to ***Assignment***.
- Set ***Display Format*** to ***0, 0.0*** or ***0.00***.
- Set ***Permission*** to ***Read/Write***. Operators can view and change the setpoint by first touching button 2 and then pressing an up or down arrow button.
- Write a Control Basic statement to start equipment based on the conditions of the setpoint.

```
IF AV2 < AV1 THEN START BO6
```

Humidity (Humidity equipped models only) Button 7—the humidity sensor— is usually associated with analog value object AV7.

- In ***Assignment***, select an analog value object to associate with the humidity button or drag an object from the Network Manager list to ***Assignment***.
- Set ***Display Format*** to ***0.***
- Set ***Permission*** to ***Read/Write***. Operators can view the room humidity by touching button 7 but cannot change the value.

Time Typically button 5 is assigned to display system time and is associated with analog value object AV5.

- Set ***Display Format*** to ***Time*** which will automatically format the display with a colon (:).
- In ***Assignment***, select an analog value object to associate with the time button or drag an object from the Network Manager list to ***Assignment***.
- Add a Control Basic line as follows:

```
AV5 = TIME
```

Verifying a functioning NetSensor To check if a functioning NetSensor is present, use the Control Basic function *NETSENSORSTATUS* on page 209.

```
IF NOT NETSENSORSTATUS THEN STOP BV1
```

Motion sensing (Motion sensing models only) Detects movement in the room and changes the value of the Auxiliary function.

- Under ***Assignment***, select an analog value object to associate with the motion sensor. Typically this is analog value object AV8
- Set ***Display Format*** to ***0.***
- Set ***Permission*** to ***Read/Write***.

- Use Control Basic to test the state of the object assigned to Auxiliary.
  - A value of -1 indicates motion
  - A value of -2 indicates no motion
  - A value of 0 or 1 indicates the auxiliary function is active.

The following Control Basic program detects motion and changes the state of variable BV9.

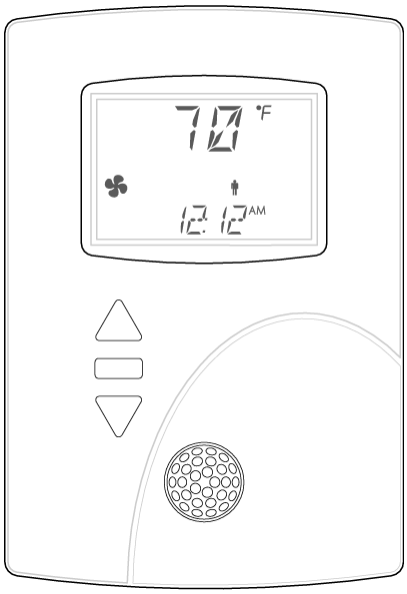
```
IF+ AV8 = -1 THEN START BV9 , STOP A
IF AV8 = -2 < 1 THEN START A
IF TIMEON( A ) > 0:02:00 THEN STOP BV9
```

Configuring the STE-9000 series NetSensor

The STE-9000 series NetSensor is a wall mounted display and sensor unit that connects to a BAC-5900, BAC-9000, or BAC-9300 series controller. The unit consists of an LCD display, a thermistor, control buttons and optional humidity, motion sensor, and CO2 sensors. Up to eight NetSensors can be configured for each controller.

- To setup a KMD series NetSensor, see the topic *KMD series NetSensors for BACnet* on page 404.
- To open a NetSensor configuration page, see the topic, *Configuring BACnet devices and objects* on page 300.

Illustration 31–3 STE-9000 series NetSensor



Caution

BAC-9000 and BAC-9300 series controllers are programmed for specific display functions with an STE-9000 sensor. Changing the sensor configuration may disrupt controller operation. Refer to the controller application guide before changing NetSensor configuration.

General tab

Sensor model

Configured Sensor Selects the default configuration for each model of NetSensor.

Detected Sensor Shows the model of the NetSensor connected to the controller.

Firmware Version Shows the firmware version.

Object Name

Object Name Enter the name of the NetSensor to be shown in the Network Manager list.

Level One Password Enter the numerical Password One for the user accessible setpoints.

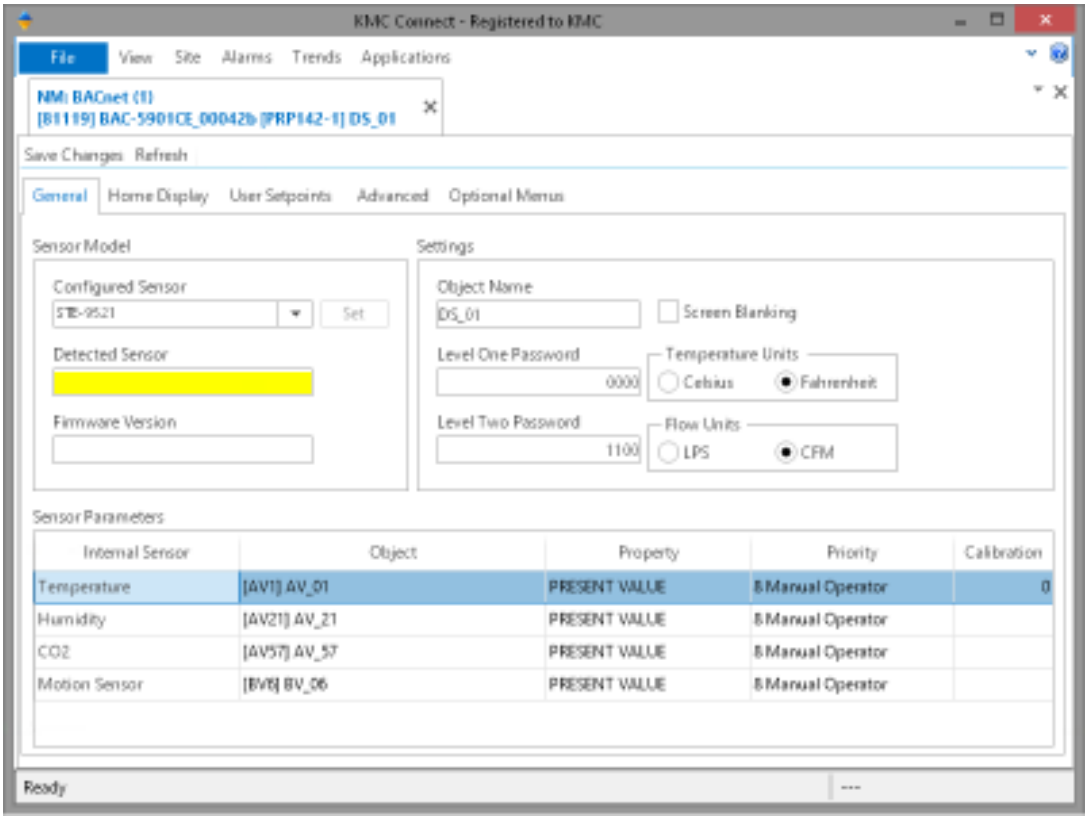
Level Two Password Enter the numerical Password Two for the configuration functions used by a controls technician.

Screen Blanking When selected, the screen automatically darkens after a few seconds.

Temperature Units Select the temperature display units as either degrees Fahrenheit or Celsius.

Flow Units Select the flow units to be in CFM (Cubic Feet per Minute) or LPS (Liters Per Second).

Sensor Parameters The properties in the Sensor Parameters list associate a sensor function in the NetSensor with a value object in the controller.



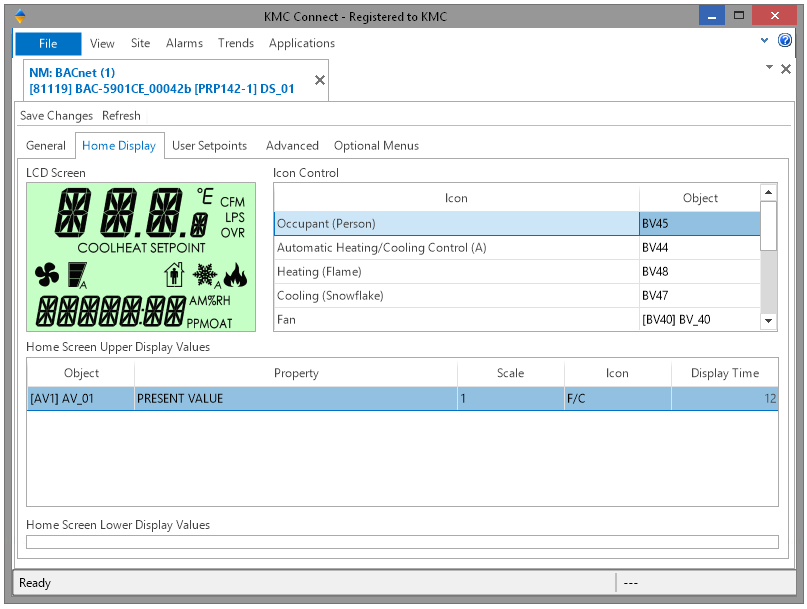
Home Display tab

LCD Screen A representation of the sensor display.

Icon Control The properties in the Icon Control list associate a screen icon with a Binary Value object. The icon is displayed when the object is active.

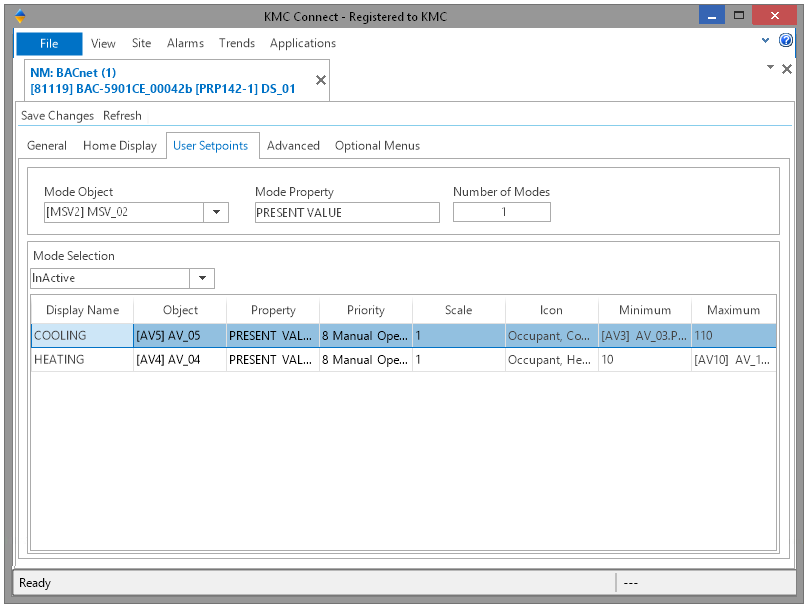
Home Screen Upper Display Values This a list of objects whose Present Value properties are displayed in the upper display of the home screen. The time of display for each object is set by the value in column Display Time.

Home Screen Lower Display Values This a list of objects whose Present Value properties are displayed in the lower display of the home screen. Set the time of display for each object in the column **Display Time**.



User Setpoints tab

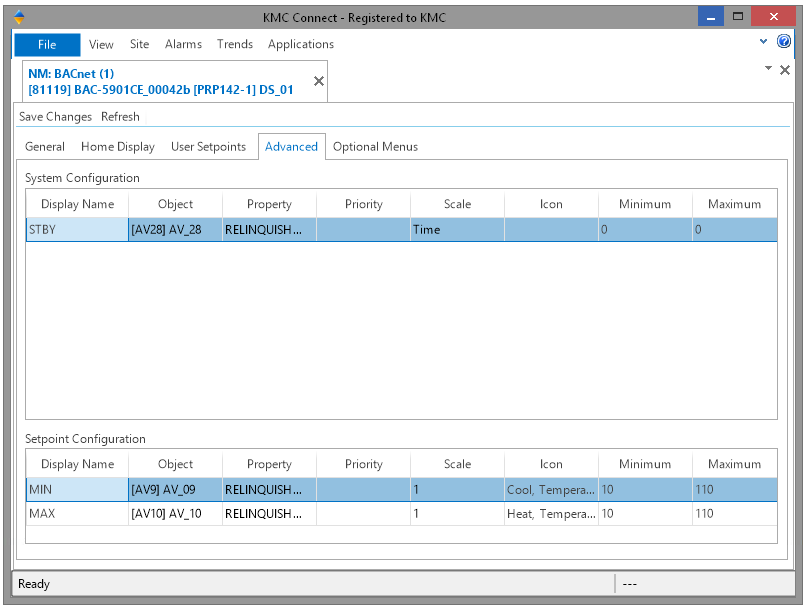
This tab controls the order of items in the NetSensor configuration menus. To make changes, right-click a line in the list and choose Add, Insert, Delete, or Move Up.



Advanced tab

System Configuration Items in this list associate a value object with a system configuration function such as fan or reheat type.

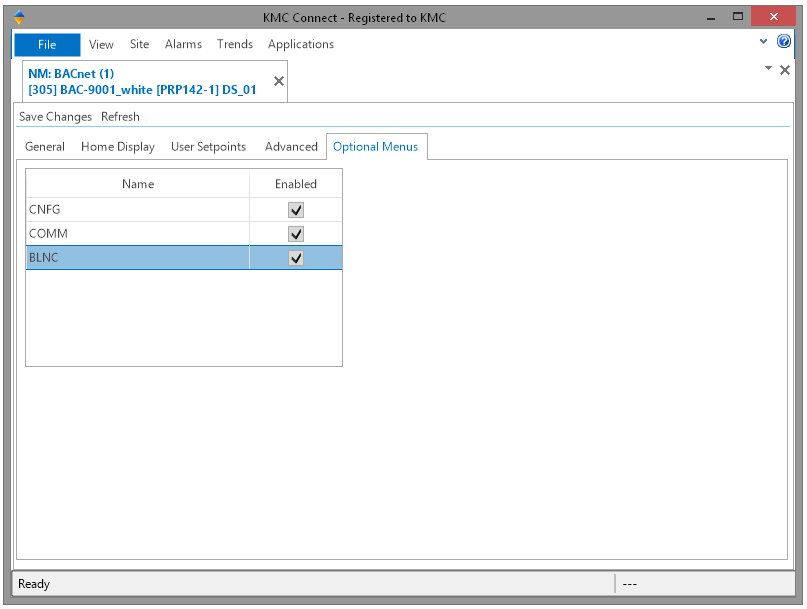
Setpoint Configuration This list designates the value objects and values for setpoint limits.



Optional Menus tab

Enables or disables the CNFG (Configuration), COMM (Communication), or BLNC (Balancing) menus.

Note: The Balancing menu is available only in VAV controllers.



Section 32: BACnet tables

The topics in this section cover for using look-up tables with analog inputs or from within a Control Basic program.

KMC Connect supports look up tables for both analog inputs and Control Basic. The fundamental operation of the tables are similar but the tables are not interchangeable and each has its own rules.

CBasic A CBasic (Control Basic) table performs a look-up function within a Control Basic program. It is based on 32 pairs of data that are interpreted in a Control Basic function. See the topic *CB tables* on page 414 and the Control Basic keyword *TBL* on page 228.

Input or device tables An input table, also referred to as a device table, is used with an analog input object to convert an input—such as a voltage from a temperature transmitter—to an output such as temperature in degrees. See the topic *Input tables* on page 413.

Input tables

An input table converts a non-linear input—such as the voltage across a thermistor—to an output such as temperature in degrees. KMC BACnet controllers include two types of input tables.

- Built in tables for Type II and Type III thermistors and airflow sensors
- Custom tables for special applications from third party devices

Table 32-1 Device table descriptions

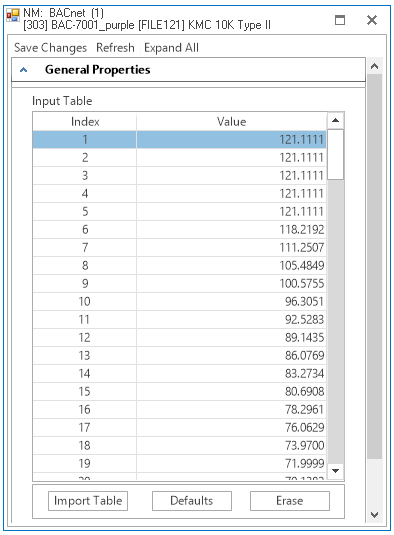
Device in table	Description
KMD 10K Type II	KMC Type II thermistors
KMD 10K Type III	KMC Type III thermistors
Airflow Sensor Table	BAC–7000 series VAV controllers

To open a BACnet Input table, see the topic *Configuring BACnet devices and objects* on page 300.

See the related topic *CB tables* on page 414 to use a look-up table in Control Basic.

To set up a custom input table, calculate a set of conversion factors based on 128 sample points spread over the voltage range of the input. Enter the output value that corresponds to the input voltage in the ***Index*** column.

Illustration 32–1 Input table



Generate Defaults Restores the default values for the KMD thermistors or the airflow sensor.

Erase Clears the table of all data.

Import Tables Import a simple text file of values. The text file must contain exactly 128 items.

CB tables

A CB(Control Basic) table is used by the Control Basic function *TBL* on page 228 to perform a look up function. This may be useful for the following reasons:

- To create special functions within Control Basic.
- When the value of an expression is nonlinear or requires a complicated calculation.

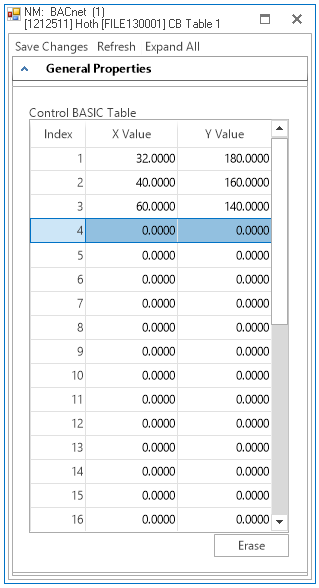
A table consists of 32 data pairs stored in two columns. Entries in the ***X Value*** column corresponds to an input value which is usually an analog input or value object. The ***Y Value*** entries are the result of the look up function and are also stored in an analog object or variable.

- Values in the ***X Value*** column must be entered in ascending order.
- The range of the input value is limited by the range of values in the ***X Value*** column.
- Control Basic performs a linear interpolation between data pairs.

To open a CB table, see the topic *Configuring BACnet devices and objects* on page 300.

See also the keyword *TBL* on page 228 to use the table data in Control Basic.

Illustration 32–2 CB table



Section 33: Supported engineering units

KMC Connect supports the engineering units listed in this section.

Illustration 33–1 Analog units

%	deg-days-C	KBTU	m3/hr	PF
%/sec	deg-days-F	KBTU/hr	m3/min	ppb
%Obscuration/ft	DegF	kg	m3/sec	ppm
%Obscuration/m2	DegF/hr	kg/hr	mA	psi/DegF
%RH	DegF/min	kg/m3	MBTU	radians
/hr	DegK	kg/min	meters	radians/sec
/min	DegK/hr	kg/sec	Meters/s/s	RPM
/sec	DegK/min	Khertz	Mhertz	sdays
amperes	degrees-phase	KJ/kg	millibars	sec/100
amps/m	delta-deg-F	KJoules	minutes	seconds
amps/m2	delta-deg-K	Kjoules/DegK	Mjoules	siemens
amps-sq-m	farads	Kjoules/kg-dry-air	Mjoules/DegK	siemens/m
bars	foot-canbdles	KMH	Mjoules/ft2	teslas
BTU	ft	KOhms	MJoules/kg-dry-air	therms
BTU/hr	ft/min	Kpa	Mjoules/m2	ton-hours
BTU/lb	ft/sec	KV	mm	tons
BTU/lb-dry-air	ft2	KVA	mm/min	tons/hr
candelas	ft3	KVAR	mm/sec	Tons-R
candelas/m2	ft3/min	KWH	mm-mercury	US-Gal
cm2	ft3/sec	KWH	MOhms	US-Gal/min
cm2	grams/min	KWH/ft2	mOhms	V/DegK
cm-mercury	grams/sec	KWH/m2	months	V/m
cm-water	gr-water/kg-dry-air	L/hr	MPH	VA
currency1	hectoPa	L/min	msec	VAR
currency10	henrys	L/sec	MV	Volts
currency2	hertz	lbf/in2	MVA	watt-hours
currency3	horsepower	lb-mass	MVAR	watts
currency4	hours	lb-mass/hr	mVolts	watts/ft2
currency5	Igal/min	lb-mass/min	MW	watts/m/DegK
currency6	Imperial-Gal	lb-mass/sec	mwatts	watts/m2
currency7	in	liters	MWH	watts/m2/DegK
currency8	in2	lixes	newton	webers
currency9	in-mercury	lumens	newton-m	weeks
cycles/hr	inw	m/hr	newtons/m	years
cycles/min	joules	m/min	newton-sec	
Deg-Angular	joules/DegK	m/sec	no-units	
DegC	joules/kg-DegK	m2	ohm-meters	
DegC/hr	joules/kg-dry-air	m2/newton	ohms	
DegC/min	joules/sec	m3	Pa	

Table 33-1 Binary unit pairs

Normal Inactive	Normal Active
OFF	ON
Stop	Start
Normal	Alarm
Closed	Open

Table 33-1 Binary unit pairs (continued)

Normal Inactive	Normal Active
Cool	Heat
Unocc	Occupied
Disable	Enable
Normal	High
Normal	Low
No	Yes
Low	High
Inactive	Active

Section 34: Reference to KMC BACnet controllers

The information in the following table lists characteristics of the BACnet objects in KMC BACnet controllers.

- For detailed specifications for each controller see the installation and operation guide packed with the controller.
- For information about connecting controllers to a building automation system see the installation and operation guide packed with the controller.
- For configuring the properties in an object, see the topic *Reference to BACnet objects* on page 315.

Notes:

- Model BAC–5802 and models ending with ***C*** do not have a hardware based real-time clock.
- All VAV models include one input dedicated to the airflow sensor and one output dedicated to the damper motor.

Table 34-1 KMC BACnet advanced application controllers

	BAC-5801 BAC-5802	**BAC-5831**	**BAC-7301 BAC-7301C**	**BAC-7302 BAC-7302C**	**BAC-7303 BAC-7303C**	**BAC-7401 BAC-7401C**
Function and type	General AAC	General AAC	AHU AAC	RTU AAC	FCU AAC	HPU AAC
Universal inputs Analog, accumulator or binary	8	16	4	4	4	4
Outputs Analog or binary	8	12	3	1	2	4
Outputs, single stage triac			1	1	1	
Outputs, dual-stage triac				2	1	
Outputs, relay						
Binary value objects	40	40	40	40	40	40
Analog value objects	40	40	40	40	40	40
PID loop objects	8	12	4	4	4	4
Weekly schedule objects	8	8	8	8	8	8

Table 34-1 KMC BACnet advanced application controllers (continued)

	BAC-5801 BAC-5802	BAC-5831	BAC-7301 BAC-7301C	BAC-7302 BAC-7302C	BAC-7303 BAC-7303C	BAC-7401 BAC-7401C
Calendar objects	3	3	3	3	3	3
Programs objects	10	10	10	10	8	8
Tables (User Defined)	2 (+3)	2 (+3)	2 (+3)	2 (+3)	2 (+3)	2 (+3)
Trend objects	8	8	8	8	8	8
Notification objects	8	8	8	8	8	8

Table 34-2 KMC BACnet VAV advanced application controllers

	BAC-7001 BAC-7051	BAC-7003 BAC-7053
Function and type	VAV AAC	VAV AAC
Universal inputs Analog, accumulator or binary	3	3
Airflow input	Yes	Yes
Outputs Analog or binary	3	1
Outputs, single stage triac		1
Outputs, dual-stage triac		
Outputs, relay		1
Binary value objects	40	40
Analog value objects	40	40
PID loop objects	4	4
Weekly schedule objects	8	8
Calendar objects	3	3
Programs objects	10	10
Tables (User Defined)	2 (+3)	2 (+3)
Trend objects	8	8
Notification objects	8	8

Table 34-3 KMC BACnet Building Controller

	BAC-A1616BC	**CAN-168EIO**
Function and type	B-BC	I/O Expansion
Universal inputs Analog, accumulator or binary	16	16
Outputs Analog or binary	16	8
Binary value objects	100	40
Analog value objects	100	40
Multistate value objects	10	
PID loop objects	16	12
Weekly schedule objects	10	8
Calendar objects	10	
Programs objects	32	
Tables (User Defined)	16 (+8)	
Trend objects	32	
Notification objects	10	
Event enrolment object	10	

KMC Connect

Part VIII: Applications and wizards

Section 35: Application selection, configuration, and deployment

The Selection, Configuration, and Deployment applications are tools to configure controllers with KMC supplied controller application programs.

Use the Selection, Configuration, and Deployment applications to manage the options in programs supplied by KMC Controls in *Conquest compliant controllers* on page 469. In addition to configuring controllers, it also contains a selection of documents that can be used to plan and commission a job.

Using the applications is a three step process.

Selection This is the start of the process. By selecting from a list of features in the supplied application library, the possibilities are narrowed to the set of features that best fit the job requirements. The selections are saved in an application job file.

Configuration This application customizes and fine tunes the default settings for a chosen application. The settings are saved as templates in the selected application job.

Deployment For deployment, templates are matched with and then written to individual controllers.

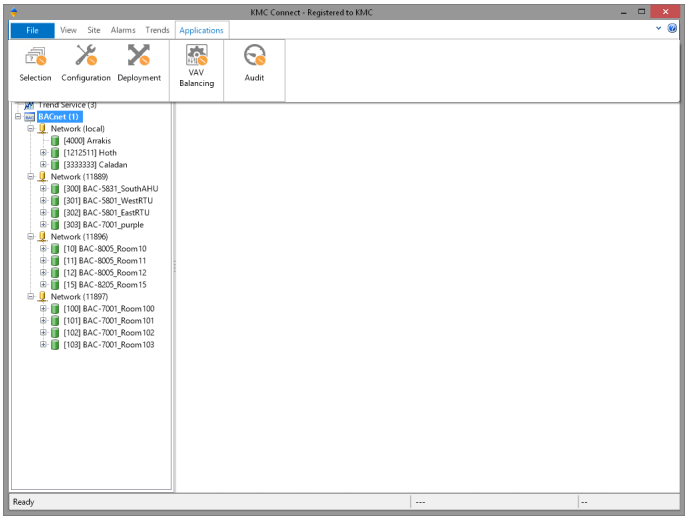
Details of the applications.

- The Selection and Configuration applications can be used without connecting to a site.
- For easiest configuration, plan the site with sequential device instance numbers for controllers with identical applications.
- More than one type of application template (VAV, FCU, RTU, etc.) can be developed for a job.

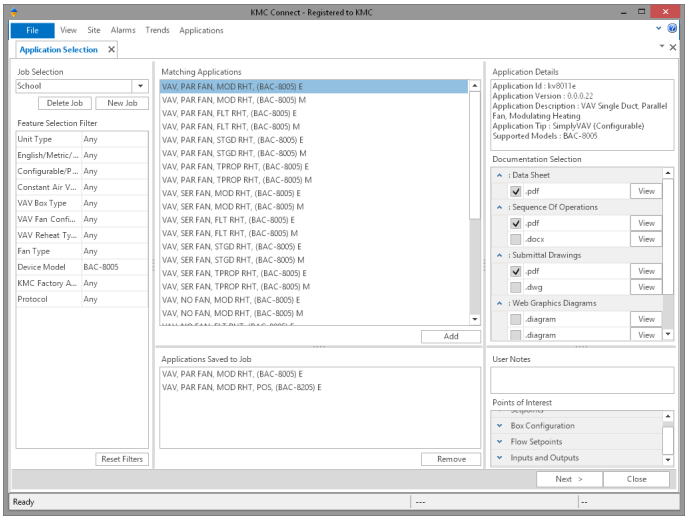
Step 1—Selection

The Selection page sets up application jobs.

1 On the **Applications** tab click **Selection**.



2 Under **Job Selection**, choose an existing job or click **New Job** and enter a name for the new job.



3 Under **Feature Selection Filter**, select the features for the job.

As features are selected, the features that are not applicable are removed.

Example: Choosing ***VAV Single Duct*** from ***Unit Type*** will remove all features that do not apply to VAV single duct applications.

4 Under **Documentation Selection**, select or clear the check boxes next to each type of document available for the job.

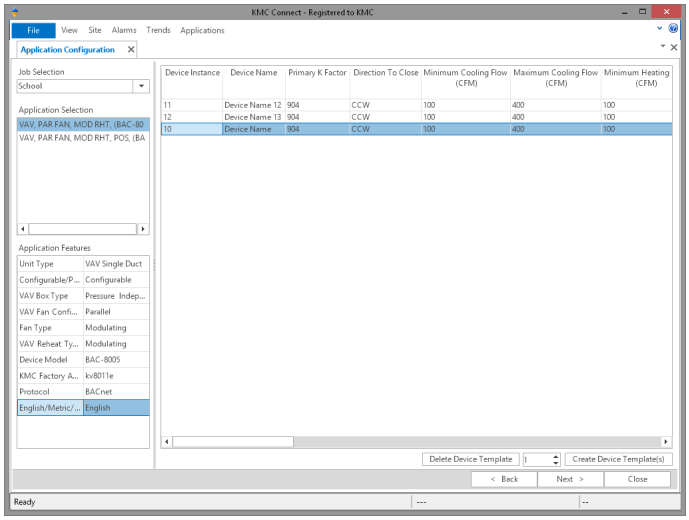
Note: Documents must be selected before the next step. Changing document selection after the next step does not add documents to the job.

5 Under **Matching Applications**, select the model or models that are the best fit for the application. Click **Add** to save them to the job. More than one type of application can be added to each job.

6 Click **Next** to advance to the Application Configuration tab.

Step 2—Configuration

This step sets the values for the actual points of interest in the application. Setpoints, fan operation, and VAV terminal unit parameters are all examples of points of interest.



1. Select a job and application from the Job and Application Selection drop down boxes. The application is added to the template list.
2. Make any required changes to the template such as changing a default setpoint.
3. Make a clone of the template for each target controller.

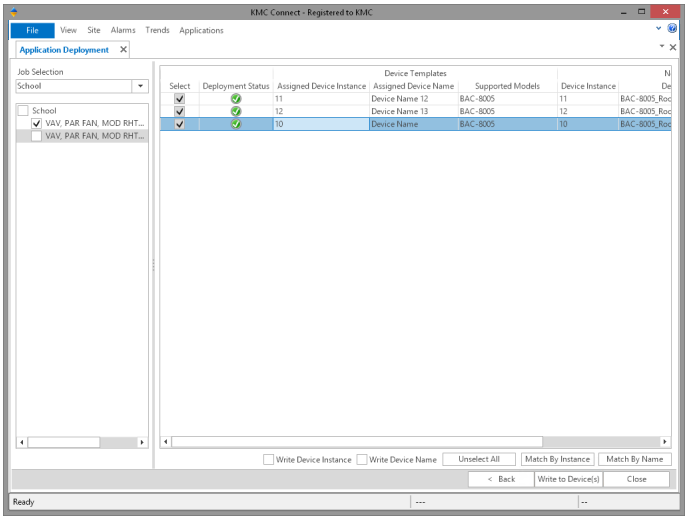
   Example: A job may use the same model VAV controller but, the controllers may be mounted on VAV units with different K-factors. Make a duplicate template for each size of VAV terminal unit and enter the correct K-factor for each template.

   Note: The device instance for each clone is automatically incremented. For example, if you start with device instance 10 and make nine clones, the clones will be numbered 11-19.
4. When finished, click **Next**. The templates are stored in the job file and the application advances to Deployment.

Step 3—Deployment

This step matches device templates to actual controllers in the Network Manager list.

1. Select a job and application from the Job and Application Selection drop down boxes. The templates for the application are added to the Device Template list
2. The application attempts to match the templates to the actual controllers in the Network Manager list. The match is either by device instance or device name.



3 When finished, click **Write to Devices**. The application begins the process of deploying the templates matched to each controller. This may take several minutes.

Section 36: Audit application

The Audit application is a tool to detect changes to the configuration of controllers on the BAS.

Use the Audit application to establish a baseline of performance and then, in the future, compare current settings to the baseline.

Using the Audit application is a two part process.

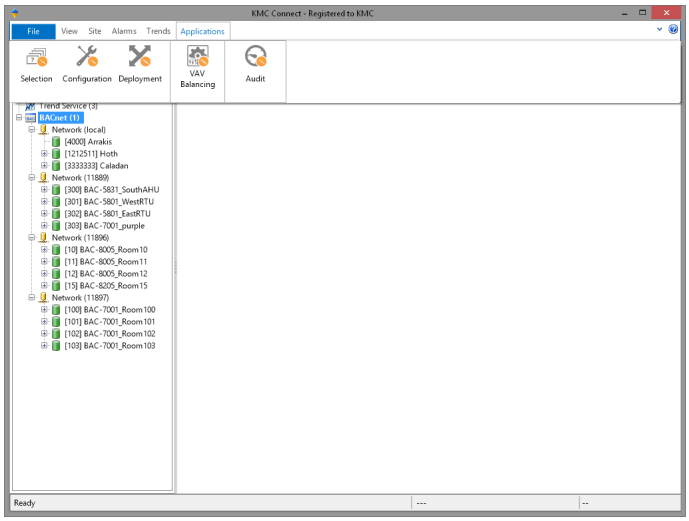
- After controllers are set up and commissioned, the values of the points-of-interest are saved as a baseline.
- To verify performance, baseline values are compared to current values.

For a list of controllers compatible with the application, see *Conquest compliant controllers* on page 469.

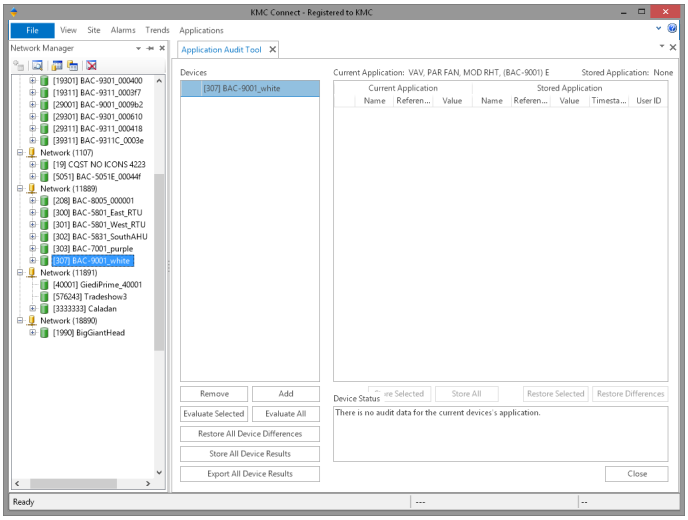
Establishing the baseline

Before establishing an audit baseline, set up the controllers. Once the baseline is established, it can be used at anytime to detect changes to the system.

1 On the **Applications** tab, click **Audit**.



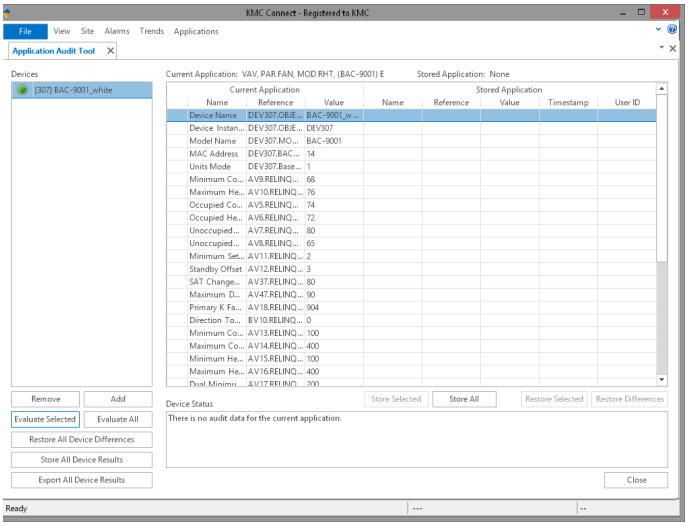
2 Open the Network Manager list and drag a device, network with devices, or the BACnet service icon to the Devices area. Only devices that are Conquest compliant are available for audit. All others are marked with yellow triangle ⚠.



3 Do one of the following:

- Click **Evaluate All** devices.
- In the devices list, right-click a device name and choose **Evaluate** from the shortcut menu.

The Audit application reads the points-of-interest and then lists them in the Current Application column.



4 Save the baseline by clicking **Store All** or **Store Selected**. This adds the values to a stored baseline for future reference.

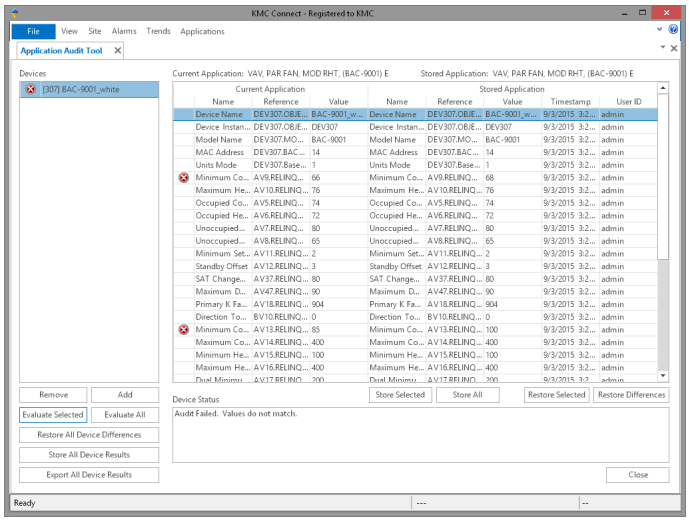
Comparing controllers to a baseline

For controllers with an established baseline, the values in the controllers can be evaluated at any time against the baseline.

1 On the **Applications** tab, click **Audit**.

2 Open the Network Manager list and locate the objects to evaluate against the stored baseline. Drag the devices, network icon or the BACnet service icon to the Devices

area.



3 Click **Evaluate** or **Evaluate All**.

4 Devices and object that do not match the values in the baseline are flagged with the changed icon .

- To use the current values in the controller as a new baseline, click **Store Selected** or **Store All**.
- To restore the device to the original configuration, click **Restore All Device Differences**.

Section 37: VAV Balancing application

Topics in this section are for control technicians or engineers who will be balancing the airflow in VAV controllers.

The VAV Balancing application has three parts.

- *The VAV Balancing sequence* on page 433 is a tool for control technicians to balance the airflow in a VAV unit.
- *Commanding VAV airflow* on page 437 to monitor airflow through a VAV unit without starting the balancing sequence in a controller.
- *Commanding VAV airflow* on page 437 to temporarily set a specific airflow.

For a list of controllers compatible with the VAV application, see the topic *Conquest compliant controllers* on page 469.

The VAV Balancing sequence

Balancing airflow is the process of calibrating the internal airflow sensor of a VAV controller to a known standard. In the field, airflow is measured with an airflow hood or other measuring instrument and then compared to the controller's internal airflow measurements.

When the balancing sequence starts, all other functions of the controller are locked out.

The airflow balancing procedure described in this section requires the following items:

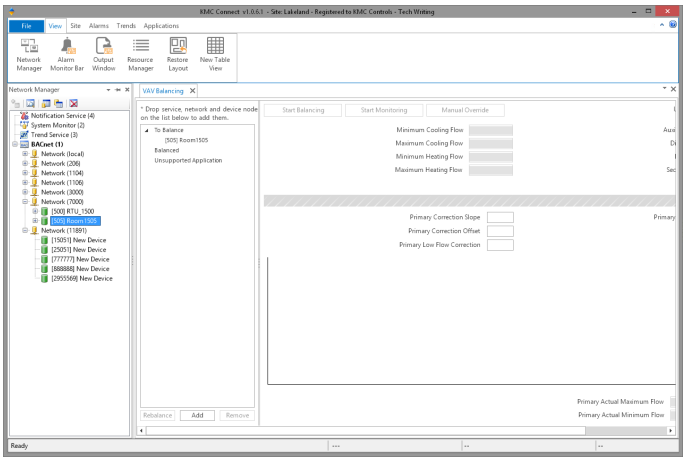
- A flow hood or other accurate method to measure airflow.
- The engineering design specifications for the VAV airflow setpoints.

For a list of controllers compatible with the wizard, see *Conquest compliant controllers* on page 469.

Note: The procedures in this section are for pressure independent systems only.

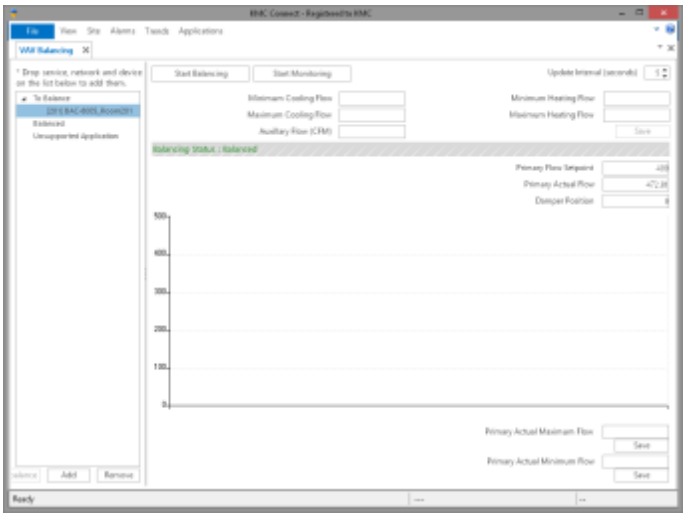
Tip: Once balancing has started, all steps must be finished in order. If the procedure is interrupted, unpredictable VAV operation will result.

1 On the **Applications** tab, click **VAV Balancing**.



2 Open the Network Manager list and drag a device, a network with devices, or the BACnet service icon to the **To Balance** area.

3 Select an individual device and then click **Start Balancing**. The application commands the VAV controller to position the damper for maximum airflow.

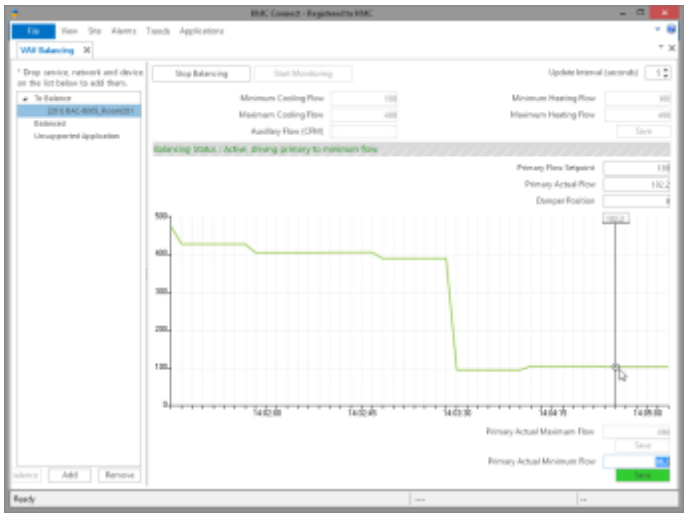


4 Monitor the airflow graph and the value in Primary Actual Airflow until the airflow stabilizes at or near the maximum airflow value.

5 At the VAV terminal unit outlet, measure the airflow with an airflow hood or other accurate instrument. Make note of the value.

6 Enter the actual measured airflow value in the **Primary Actual Maximum Flow** text box and then click **Save**. The controller immediately begins to position the damper for minimum air flow.

7 Again, monitor the airflow graph and the value in Primary Actual Flow and wait for the airflow to stabilize at or near the minimum airflow value.



8 At the VAV terminal unit outlet, measure the airflow with an airflow hood or other accurate instrument. Make note of the value.

9 Enter the actual measured airflow value in the **Primary Actual Minimum Flow** text box and then click **Save**. The program calculates new airflow constants and returns the VAV controller to normal operation.

Note: For dual-duct VAV systems, both the primary and secondary airflows are displayed by the application. Both airflows must be balanced before the controller is considered balanced.

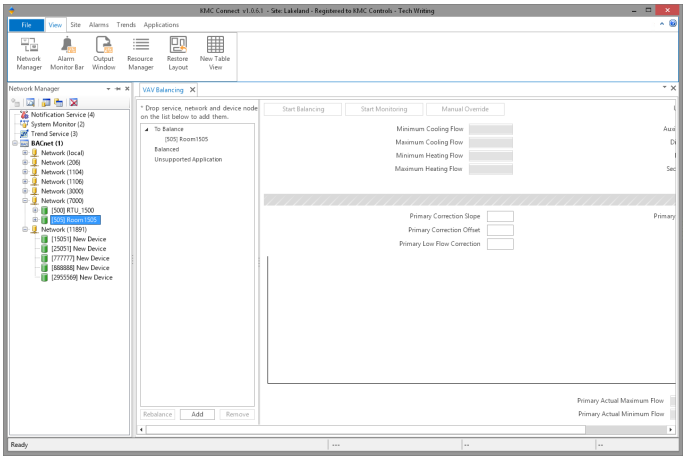
VAV flow monitoring

Using the flow monitoring function in the VAV application is a way to monitor airflow through a VAV unit without starting the balancing sequence in a controller.

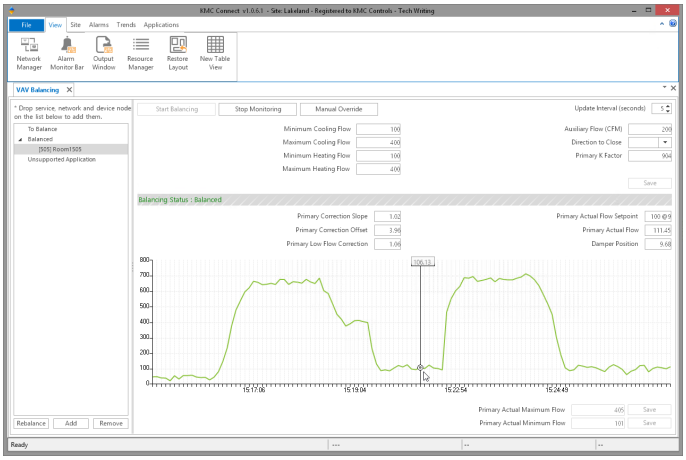
To balance the airflow for a VAV unit, see the topic *The VAV Balancing sequence* on page 433.

To monitor airflow through a VAV unit, do the following.

1. On the **Applications** tab, click **VAV Balancing**.



2. Open the Network Manager list and drag a device, a network with devices, or the BACnet service icon to the **To Balance** area.
3. Select an individual device and then click **Start Monitoring**. The application begins graphing the airflow through the VAV unit.

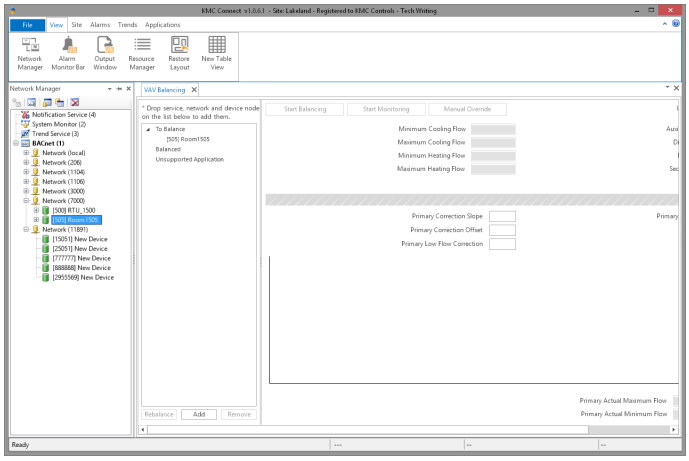


Commanding VAV airflow

Use the VAV airflow manual override function to command airflow to a specific airflow value. When a controller is in manual override, the calculated airflow setpoint is replaced with a temporary fixed setpoint. This function is used to make adjustments or measurements that require a steady, specific airflow.

To command a specific airflow through a VAV unit, do the following.

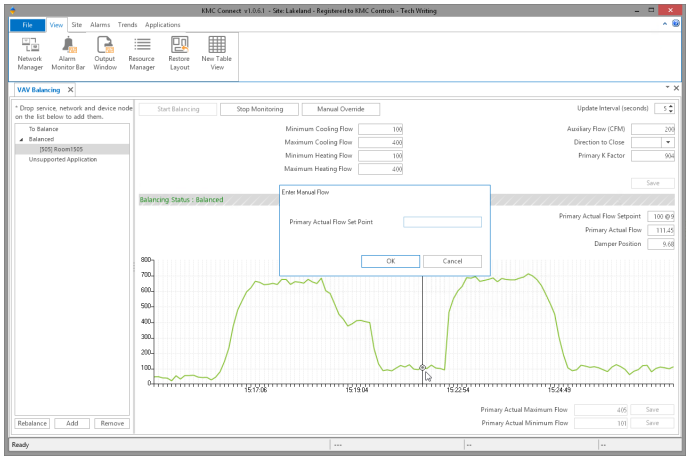
1 On the **Applications** tab, click **VAV Balancing**.



2 Open the Network Manager list and drag a device, a network with devices, or the BACnet service icon to the **To Balance** area.

3 Select an individual device and then click **Manual Override**.

4 Enter an airflow setpoint and then click **OK**.



Section 38: Custom Applications Wizard

This section describes how to build unique applications with the Custom Application Wizard to use with the Selection, Configuration, and Deployment applications.

Use the Custom Applications Wizard to save the customized setup of a controller that contains a standard KMC application. It can also be used to define points of interest for the Audit application for older KMC BACnet controllers. Custom applications are saved in the Resource Manger Custom Application Library folder.

Conquest and SimplyVAV controllers—For Conquest and SimplyVAV controllers use the Custom Applications Wizard to do the following:

- Build a unique application based on a standard application.
- Define a custom set of points-of-interest for auditing.
- Save all modifications, object assignments, Control Basic, NetSensor configure, etc. in `.bnd` file that can be deployed to other controllers.

Non-Conquest and third-party controllers—Define a set of points-of-interest for the Audit application.

Before starting the wizard, complete the configuration of the controller that will be the template for the custom application.

Setting up the application

The start of building a custom application sets up the application name, features, and supported models.

1. Open the Network Manager list and drag a device with the custom application to the orange workspace area.
2. Enter the following items.
   - **Application Name**—Enter a unique name for the custom application. This name will appear in the Matching Applications list of the Selection Tool. This will also be the name of the folder in the Custom Applications library.
   - **Application ID**—Enter a unique identification for the application that is stored as a property in the device object.
   - **Library Name**—Choose an existing custom library or click **New** to add a new library. New libraries are added to the Custom Application Library folder in the Resource Manager.
   - **Application ID Reference** —A reference ID for the application that is stored as a property in the device object. This is for information only and cannot be changed
   - **User Notes**—Add any pertinent information about the custom application.
   - **Application Description**—Describe the custom application.

3. Choose a **Unit type** or click **New** to create a new unit type. The new unit type must be unique. It cannot duplicate an existing KMC or custom name.
4. Check the **Include VAV Balancing** and **Include VAV Configuration** check boxes if these features are required.
5. Add new features and values.
   - The feature name is added to the Feature Selection Filter list in the Selection tool.
   - The value is used to select a specific feature.
6. Select, add, or delete a supported model. The model of the template device cannot be deleted.
7. To add documents to the custom application library, select the **Include Documents** check box. Document selection will take place after points selection.
8. When ready, click **Next** to advance to points selection.

Points selection

This part selects points that will be used by the Selection and Audit tools.

1. Select points to include in the custom application.
   - **Configuration**—These are points that are visible and can be changed in the Configuration tool.
   - **RD (Relinquish Default)**—Select to make the Relinquish Default property part of the Audit tool process.
   - **PV (Present Value)**—Select to make the Present Value property part of the Audit tool process.
   - **Points of Interest**—Adds points to the application for TotalControl graphics.
   - **Graphical points**—Adds points to the application for Niagara graphics.
2. To add objects not in the points list, click **Add** and then choose an object and property to add to the list.
3. Click **Next** to advance to document selection.
   - If the **Include Documents** check box was selected in the first screen, the wizard advances to Document Selection.
   - If **Include Documents** check box was not selected, the wizard saves the custom application in the Custom Affliction Library folder in the Resource Manager.

Document Selection (optional)

If the **Include Documents** check box was selected in the first screen, you can select documents to add to the custom application. The documents are copied from their original location and then stored with the job in the Custom Affliction Library folder in the Resource Manager.

1. Choose any documents to include with the custom application.
2. Click **Next** to save the custom application in the Custom Affliction Library folder in the Resource Manager.

Section 39: Input object wizards

Topics in this section cover the BACnet input object wizards.

The BACnet input wizards are a group of the most common properties that must be set up to configure a controller input.

To open an input wizard, do the following:

1. In the Network Manager list, locate the controller and then open the Inputs folder.
2. Locate an analog or binary input icon.
3. Right-click the icon and choose **Analog Input Wizard** or **Binary Input Wizard** from the shortcut menu.

Each of the wizards is explained in one of the following topics.

- *Analog input object wizard* on page 441
- *Binary input object wizard* on page 445

Analog input object wizard

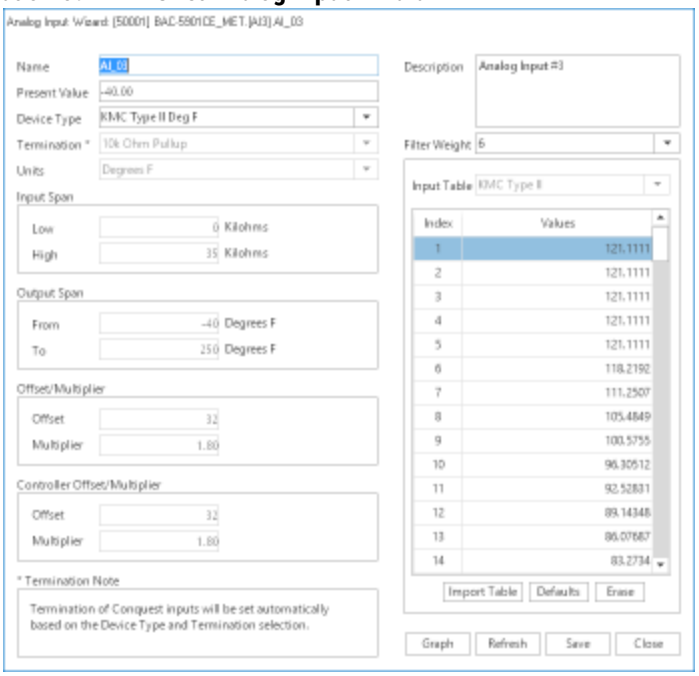
Use the Analog Input Wizard to match the BACnet Analog Input object and Input device table properties to the input device connected to the controller. These properties are fully described in the topics *Input object–analog* on page 340 and *Input tables* on page 413.

To open an Analog Input Wizard, do the following:

1. In the Network Manager list, locate the controller and then open the Inputs folder.
2. Locate an analog input icon.
3. Right-click the icon and choose **Analog Input Wizard** from the shortcut menu.

Additional information is in the Application Guide AG160516A, ***KMC Input Wizards***.

Illustration 39–1 BACnet Analog Input Wizard



The properties in the wizard are those properties required to match the signal from the input device to the input of the controller.

Standard device types

Choosing an item from the Device Type list automatically configures most of the input object properties.

To configure a standard device type, do the following:

1. In the **Device Type** list, select the entry that matches the type of device connected to the controller. Passive temperature sensor inputs (thermistors) are listed as KMC Type II or Type III in degrees Centigrade or Fahrenheit.
2. For non Conquest controllers, set the correct input termination as noted in **Termination** and **Termination Note**. The exact method of termination will be described in the instructions for the controller.
3. Enter a name and description for the input.
4. If required, change the **Filter Weight**.
5. Click **Save** when finished.

Active Custom device types

For active input devices, devices with a voltage or current output, that are not in the Device Type list, set up a custom device type. Setting up a custom device requires information supplied by the manufacturer of the device. This information is either a voltage or current range and a monitored range. For example a CO2 sensor output signal may be specified as 500–2000 PPM and an output current of 4–20 mA.

1. In **Device Type**, choose **Custom**.

2 From **Termination**, choose the type of voltage or current signal connected to the controller. This setting must cover the range of the signal from the active device connected to the input.

- If the controller is a Conquest series controller, the termination is set by the wizard.
- If the controller is not a Conquest series controller, set the termination as indicated in Termination Note. The exact method of termination will depend on the model and manufacture of the controller.

3 In **Input Span**, enter the **Low** and **High** range of the input signal as specified by the device connected to the controller.

4 In **Output Span**, enter the **Low** and **High** monitored range of the connected device.

5 Typically, the output from an active device is linear and does not require a table to correct nonlinear signals.

- If a table is not required, in **Input Table** select **None**.
- If a table is required, choose a table from **Input Table**. Either directly enter values or import the values by clicking **Import Table**. See *Input tables* on page 413 for details on building and importing tables.

6 Select **Units** to match the type of input signal.

7 Enter a name and description for the input.

8 If required change the **Filter Weight**.

9 Click **Save** when finished.

Passive Custom device types

For passive input devices (thermistors, potentiometers, RTC, etc.) that are not in the Device Type list, set up a custom device type. A passive device is powered from a pull-up resistor that is typically switched with a jumper or DIP switch at the input. The exact method will depend on the model and manufacturer of the controller.

1 In **Device Type**, choose **Custom**.

2 From **Termination**, choose either 1K Ohm or 10K Ohm pullup.

- The 1K Ohm pullup resistor is for the RTC platinum sensor and is not supported in all controllers.
- If the controller is a Conquest series controller, the termination is set by the wizard.
- If the controller is not a Conquest series controller, set the termination as indicated in **Termination Note**. The exact method of termination will depend on the model and manufacture of the controller.

3 In **Input Span**, enter the **Low** and **High** range of the resistance of the device as specified by the device connected to the controller.

4 In **Output Span**, enter the **Low** and **High** monitored range of the connected device.

5 Typically, passive devices have some points of nonliterary that are corrected with an input table.

- If a table is required, choose a table from **Input Table**. Either directly enter values or import the values by clicking **Import Table**. See *Input tables* on page 413 for details on building and importing tables.
- If a table is not required, in **Input Table** select **None**.

6 Select **Units** to match the type of input signal.

7 Enter a name and description for the input.

8 If required, change the **Filter Weight**.

9 Click **Save** when finished.

Related topics

- *Binary input object wizard* on page 445
- *Input object–analog* on page 340
- *Input tables* on page 413

Binary input object wizard

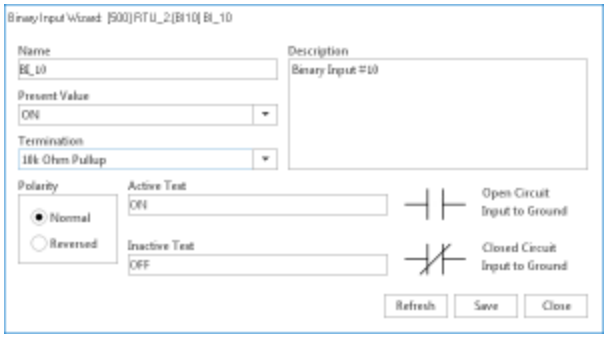
Use the Binary Input wizard to match the BACnet Binary Input object properties to the input device connected to the controller. These properties are fully described in the topic*Input object–binary* on page 345.

To open a Binary Input wizard, do the following:

1. In the Network Manager list, locate the controller and then open the Inputs folder.
2. Locate the binary input icon.
3. Right-click the icon and choose **Binary Input Wizard** from the shortcut menu.

Additional information is in Application Guide AG160516A,*KMC Input Wizards*.

Illustration 40–1 BACnet Binary Input Wizard



The properties in the wizard are those properties required to match the signal from the input device to the input of the controller. The typical application is a switch contact that is powered through a pull-up resistor in the controller. The Present Value property of the object will depend on the setting of the Polarity property and the position of the switch. This is interaction is shown in the following table.

Table 40-1 Input object polarity relationships

Passive input with pull-up	Polarity	Present Value	Text example
⊣⊢	Normal	Active	Off, Stop
⊣/⊢	Normal	Inactive	On, Start
⊣⊢	Reverse	Inactive	On, Start
⊣/⊢	Reverse	Active	Off, Stop

To configure a Binary Input object, do the following:

1. Set the termination.
   - For Conquest controllers, choose **10k Ohm Pullup**. This is the default setting and will automatically be set by the wizard.
   - If the controller is not a Conquest series controller, terminate the input with a 10k Ohm pullup resistor. The exact method of termination will depend on the model and manufacture of the controller.
2. Choose the polarity. See the table *Input object polarity relationships*.
3. Enter descriptive names **Active** and **Inactive Text**.
4. Enter a name and description for the input.
5. Click **Save** when finished.

See also the related topics *Analog input object wizard* on page 441 and *Input object–binary* on page 345.

Appendices

Appendix A: Configuring the Cimetrics BACstac driver

The Cimetrics BACstac driver connects the computer running KMC Connect to a BACnet internetwork. Topics in this section cover the configuration of BACstac for BACnet IP or BACnet Ethernet networks.

Computers running KMC Connect require the Cimetrics BACstac driver to connect to a BACnet internetwork. The following topics cover configuration, version verification, and removal of the BACstac driver.

- Prerequisite system information and requirements are covered in the topic *Before you begin* on page 449.
- Configuring the driver for the correct BACnet protocol
  - *Configuring BACstac for BACnet IP* on page 451
  - *Configure BACstac for Ethernet 8802.3* on page 456.
- Removal procedures are covered in the topic *Remove previous versions of the BACstac driver* on page 457.

Before you begin

The Cimetrics BACstac driver is installed when KMC Connect is installed. To configure the BACstac driver for connection to a BACnet internetwork you will need the following information.

- For BACnet IP networks—a local BACnet network number and UDP port number for the local network.
- For foreign device registration—the IP address and UDP port number of a remote BBMD.
- For BACnet Ethernet 8802.3—the local BACnet Ethernet network number.

Network number Network numbers are assigned by BACnet routers or devices with a routing function such as a KMC Controls BAC-16161 Building Controller. The network number in BACstac must match the network number set by the router.

Routing protocol The routing protocol is established by a router or a device with a routing function. By using Cimetrics BACstac as a driver, KMC Connect supports the following BACnet routing protocols.

- BACnet IP (Internet protocol)
- BACnet IP foreign device
- BACnet IP BACnet Broadcast Management Device (BBMD)
- BACnet Ethernet-8802.3

The routing protocol must be set to the protocol that matches the routing protocol set my the router.

UDP Port The UDP Port number is assigned by a BACnet router devices with a routing function. All BACnet IP networks will have a UDP Port associated with it. The network number in BACstac must match the port number set by the router.

Related topics

- *Configuring BACstac for BACnet IP* on page 451
- *Configure BACstac for Ethernet 8802.3* on page 456
- *Remove previous versions of the BACstac driver* on page 457

Configuring BACstac for BACnet IP

The BACstac driver is automatically installed when KMC Connect is installed. However, the driver must be configured to match the type of BACnet network to which the computer is connected.

Caution

If the computer running BACstac includes a firewall, the port selected in **UDP Port** must be opened in the firewall. If the port is not opened, BACnet devices cannot be seen in the Network Manager list and data cannot be transferred from the device to the computer running KMC Connect.

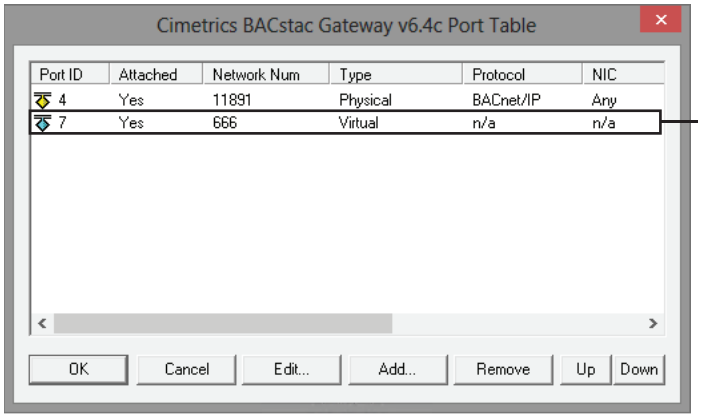
Configure the computer as a BACnet IP device

To configure the computer as a BACnet IP device, the computer must meet one of the following conditions:

- It must be connected to the same IP subnet as one of the BACnet IP networks.
- It must be registered as a foreign device to a BACnet Broadcast Management Device (BBMD).
- It must be configured as a BBMD that is part of the BACnet internetwork.

All three types of IP routing protocols require the entries under the **Port** tab. To configure the computer as an IP device, do the following.

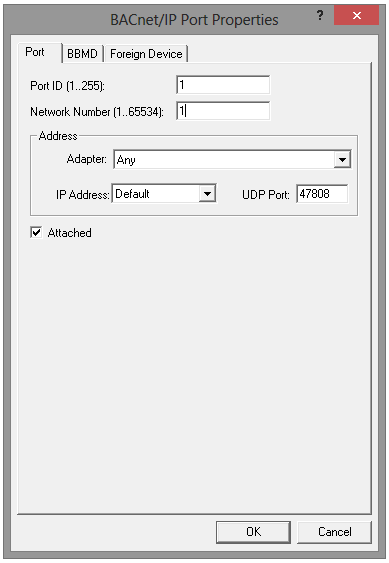
1. Choose **Start**, **Cimetrics**, and then **Configure BACstac**. The Port Table dialog opens.
2. If there are any unneeded entries in the port table, select them and then click **Remove**.



3. Click **Add** and then choose **BACnet/IP**.

4 Enter the configuration for the following in the BACnet/IP dialog:

- Port ID
- Network number
- IP Address
- UDP Port



5 If required, perform either of the following additional configurations:

- *Register as a foreign device* on page 453
- *Configure BACstac as a BBMD* on page 454

6 Click **OK** to finish and save the configuration.

Port ID Enter any non-zero number.

Network number Enter the BACnet IP network number from the BACnet system plans. The network number must match the network number for the BACnet IP network to which the computer is connected. BACnet IP network numbers are configured in routers or devices with routing functions.

IP Address This lists the IP addresses of the Network Interface Cards (NIC) installed in the computer. Select the address for the NIC in use.

Subnet Mask The IP subnet to which the computer hosting the PDS is connected.

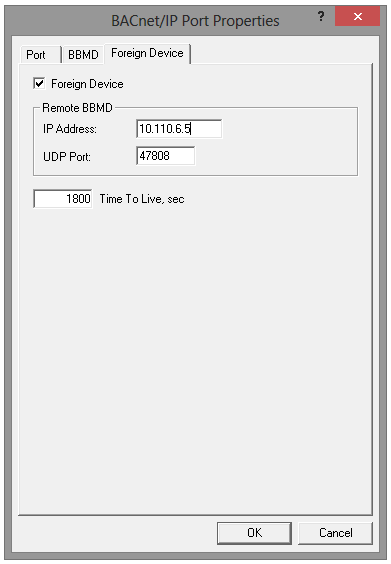
UDP Port Enter the UDP port number from the BACnet system plans. The UDP port number must match the port number for the BACnet IP network to which the computer connected. UDP port numbers are typically 47808 to 47817 (0xBAC0 to 0xBAC9 in hexadecimal notation).

Register as a foreign device

The computer can register as a foreign device to a BACnet Broadcast Management device (BBMD). Foreign device registration to a BBMD is a technique for crossing IP subnet boundaries with BACnet broadcast messages.

To configure the KMC Connect computer as a foreign device, do the following:

1. *Configure the computer as a BACnet IP device* on page 451
2. Click the **Foreign Device** tab.



3. Enter the configuration for the following properties:
   - IP Address of the remote BBMD
   - UDP Port of the remote BBMD
   - A value for Time-to-Live
4. Close all dialog boxes when finished.

IP Address Enter the IP address of the remote BBMD. If network address translation (NAT) is used between the PDS computer and the BBMD, contact the network system administrator for the correct public IP address.

UDP Port Enter the UDP Port of the BBMD to which BACstac is registering as a foreign device.

Time-to-Live Sets the interval at which the driver sends a registration message to the BBMD with which it is registered.

If the BBMD does not receive a registration message within the Time-To-Live period plus 30 seconds, the BBMD will remove the device from its foreign device table and will not send broadcast messages to the device.

- The valid time range is 1 to 65535 seconds.
- If the entry is zero (0), the registration is forever.
- The default value is 1880 seconds.

Configure BACstac as a BBMD

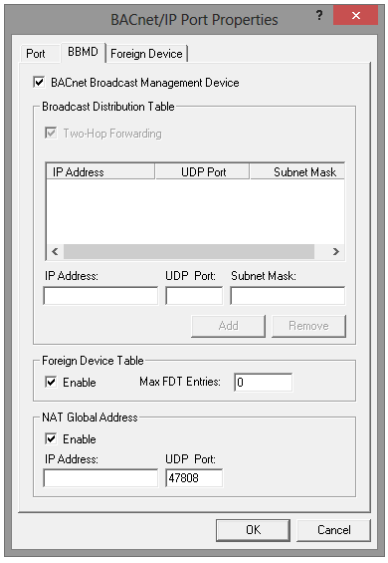
When configuring the driver as a BBMD, observe the following rules.

- Configure only one BBMD for a single IP subnetwork.
- Assign the same BACnet network number to all BBMDs on the internetwork.
- The BBMD can accept registration from foreign devices or perform BBMD-to-BBMD routing.

Note: To use BACstac as a permanent BBMD, the computer must remain connected to the internetwork and running at all times. Typically, a BBMD is not used on a computer that is used only as a service tool.

To configure the PDS computer as a BBMD, do the following:

1. Start with the procedure *Configure the computer as a BACnet IP device* on page 451.
2. Click the **BBMD** tab.



3. Select the **BACnet Broadcast Management Device** check box.
4. Enter the configuration for the following:
   - Broadcast Distribution Table
   - Two-Hop Forwarding
   - Foreign Device Table
   - Max FDT Entries
5. Close all dialog boxes when finished.

Broadcast Distribution Table The broadcast distribution table is a list of all BBMDs in the internetwork. The table includes the IP address, UDP port number and IP subnet mask of each BBMD. Use **Add** or **Remove** to edit the table.

Two-Hop Forwarding Always select the **Two-Hop Forwarding** check box.

Foreign Device Table Select the *Enable* check box to enable foreign devices to register with the BBMD. The maximum number of devices is limited by the value in **Max FDT Entries**.

Max FDT Entries Sets the maximum number of foreign devices that can register at one time. The value for **MAX FDT Entries** is 1 to 128.

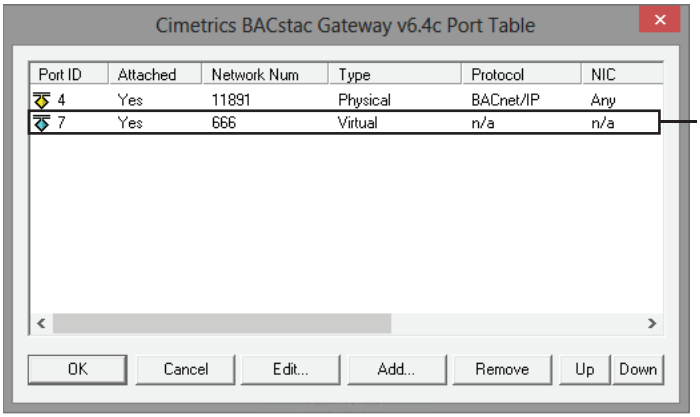
NAT Global Address When using network address translation and port address translation (NAT and PAT), enter the public IP and port number. NAT and PAT are used when the BBMD needs to be accessible from the Internet. The IT department provides the IP address and UDP Port number.

Configure BACstac for Ethernet 8802.3

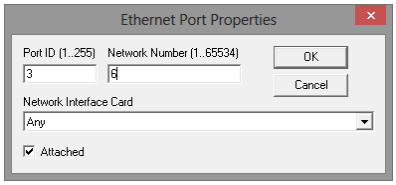
The BACstac driver is automatically installed when KMC Connect is installed.. However, the drive must be configured to match the type of BACnet network to which the computer is connected.

1 Choose **Start**, **Cimetrics**, and then **Configure BACstac**. The **Cimetrics Port Table dialog** opens.

2 If there are any unneeded entries in the port table, select them and then click **Remove**.



3 Choose **Add** and then select **Ethernet**.



4 In the Ethernet Port Properties dialog, enter the configuration for **Port ID**, **Network Number** and **Attached**.

5 Close all dialog boxes when finished.

Port ID Enter any non-zero number.

Network number Enter the local BACnet 8802.3 network number from the BACnet system plans. The network number must match the network number for the BACnet Ethernet network to which the computer is connected.

Attached Always select.

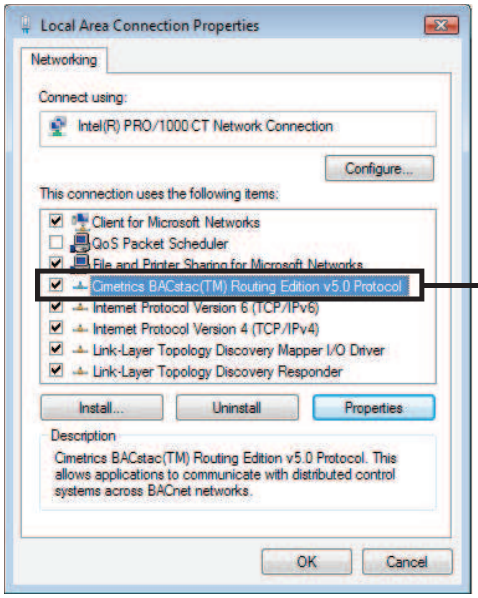
Remove previous versions of the BACstac driver

KMC Connect may require an old version of the Cimetrics BACstac driver to be removed before a new version is installed. This is usually required only on computers that were running programs other than KMC Connect or if a problem occurs during installation.

Note: This procedure applies only to BACstac versions earlier than 6.0. Version 6.0 and later are removed from the Windows Control Panel program.

To remove an old version of BACstac, do the following:

1. Choose **Start**, **Settings**, **Network Connections** and then **Local Area Connections**. The Local Area Network Dialog opens.



2. Click **Properties**.
3. Scroll through the list of connection items.
4. If you find *any* version of **BACstac (TM) Routing Edition**, select it and then click **Uninstall**.

Appendix B: Versions of Control Basic

Next Generation and Generation 5 Control Basic are advanced versions of Control Basic that are supported in newer BACnet controllers. The following information explains the variations of Control Basic and the differences among Standard, Next Generation, and Generation 5 Control Basic.

With the 2015 release of KMC Connect and the Conquest series of controllers, there are now three versions of Control Basic for BACnet controllers.

Standard Control Basic This is the version of Control Basic that was part of the release of the original KMC BAnet controllers. It is very similar to the version of Control Basic found in the KMD series of controllers. Controllers with Standard Control Basic are programmed with little change. However, when existing programs are loaded from a controller you will see some changes to the keywords and references to remote points.

Next Generation Control Basic With the release of BACstage 2.3 (June 2008) and TotalControl Design Studio 1.5, KMC Controls added Next Generation Control Basic. The new version includes additional commands and functions, new syntax, labels, and the elimination of line numbers.

Generation 5 Control Basic This version of Control Basic includes all of the enhancements of Next Generation Control Basic and adds several new functions and commands.

Review the following topics to become familiar with the new features of Control Basic:

- *Control Basic versions in controllers* on page 460
- *Changes to IF THEN* on page 462
- *Deprecated keywords* on page 461
- *File compatibility* on page 462
- *New keywords* on page 463
- *Line numbers and labels* on page 467
- *Local variables* on page 468
- *References to objects in remote devices* on page 468

Control Basic versions in controllers

The following table lists the versions of Control Basic that are supported in the BACnet controllers from KMC Controls.

Table B-1 Versions of Control Basic

	Control Basic Version		
Model number	**Standard**	**Next Gen**	**Gen 5**
BAC-A1616BC		■	
BAC-5801 BAC-5802	■		
BAC-5831	■		
BAC-5841 BAC-5842	■		
BAC-7001 BAC-7051	■		
BAC-7003 BAC-7053	■		
BAC-7301 BAC-7301C	■		
BAC-7302 BAC-7302C	■		
BAC-7303 BAC-7303C	■		
BAC-7401 BAC-7401C	■		
BAC-5900			■
BAC-9000			■
BAC-9100			■
FlexStat BAC-10000 BAC-11000 BAC-120000 BAC-130000 BAC-140000		■	

Deprecated keywords

Keywords in the table *Control Basic deprecated keywords* on page 461 changed in BACstage version 2.3. Only the keywords are changed; the functions and commands they represent remain the same.

- When writing programs for controllers with Standard Control Basic (see *Control Basic versions in controllers* on page 460), BACstage will accept and compile ***either*** the deprecated keywords or the replacement versions of the keywords. For example either DEW-POINT or DEWPOINT may be used when writing a program.
- When BACstage or TotalControl retrieves a Control Basic program from a controller and decompiles it, the deprecated keywords are replaced with the new keywords. For example DEW-POINT becomes DEWPOINT and TIME-ON becomes TIMEON.
- For controllers with Next Generation or Generation 5 Control Basic, BACstage or TotalControl will not accept or compile the deprecated keywords.

Table B-2 Control Basic deprecated keywords

Deprecated keyword	Replacement keyword
COS-1	ARCCOS
DEW-POINT	DEWPOINT
DEW-POINT-SI	DEWPOINTSI
ENTHALPY-SI	ENTHALPYSI
LN-1	INVLN
MODEL-NUMBER	MODELNUMBER
NETSENSOR-STATUS	NETSENSORSTATUS
ON-ERROR	ONERROR
OUTPUT-OVERRIDE	OUTPUTOVERRIDE
PANEL-ADDRESS	PANELADDRESS
SCHED-ON	SCHEDON
SCHED-OFF	SCHEDOFF
SENSOR-OFF	SENSOROFF
SENSOR-ON	SENSORON
SIN-1	ARCSIN
TAN-1	ARCTAN
TIME-ON	TIMEON
TIME-OFF	TIMEOFF

Changes to IF THEN

Both Next Generation and Generation 5 Control Basic support block and nested IF THEN statements.

```
Locals ChilledWaterSetpoint
AV24 = ChilledWaterSetpoint
IF BV258 THEN
    ChilledWaterSetpoint=52
ELSE
    Chilledwatersetpoint=48
ENDIF


t=TIME
IF t> 00:07:00 THEN
    IF t < 00:09:00 THEN
        START BO1
    ENDIF
ENDIF
```

File compatibility

Not all versions of BACstage and TotalControl are compatible with the versions of Control Basic.

TotalControl and KMC Connect compatibility

All versions of KMC Connect and versions of TotalControl released after June 2015 are compatible with all versions of Control Basic.

BACstage file compatibility

When saving and opening files with versions of BACstage other than BACstage 2.3, be aware of the following compatibility issues.

Note: TotalControl Design Studio and KMC Connect will not directly import BACstage `.bas` or `.ng` Control Basic files. However, programs can be pasted into a Control Basic editor window and then modified.

- No version of BACstage is compatible with the Generation 5 keywords.
- Programs saved in a controller with BACstage 2.3 can be loaded from a controller with earlier versions of BACstage. BACstage 2.2 and earlier will list the programs using the deprecated keywords and original syntax for remote points.
- If a `.BAS` file includes any of the new or deprecated keywords, BACstage versions earlier than 2.3 will open but not compile the program. The new keywords and syntax must be changed to the original format.

- When transferring a .BAS file to a controller with Next Generation basic, line numbers will have to be deleted. Labels will have to be added to lines that are referenced by a GOSUB or GOTO control statement. See *Line numbers and labels* on page 467.
- Files created with ***Backup Device*** (.BAC files) in BACstage 2.3 are backwards compatible with BACstage 2.2 and earlier.

New keywords

This topic includes only a brief description of the keywords not in the original version of Control Basic. Details for each keyword are included in help for each program that uses Control Basic.

Next Generation

The following keywords are added to Next Generation Control Basic and are also recognized by Generation 5 Control Basic.

- ALIAS
- BIND
- CONST
- FLUSH
- HALT
- ISNAN
- LOCALS
- NAN

ALIAS

Declares a local variable and dynamically binds the value of a property to the variable. It also sets two intervals at which Control Basic will read from or write to the property bound to the variable.

Syntax: *ALIAS(device, object, property, local, read interval, write interval)*

See *ALIAS* on page 182 for a more detailed description.

BIND

Binds a device instance to a physical network address. This is typically used to bind an MS/TP slave to a master device. BIND is required in only one program within the device.

Syntax: *BIND (device, network, mac, option)*

See the keyword *BIND* on page 185 for a more detailed description.

CONST

Use to declare one or more variables and assign to them a fixed value. Do not use with variables that change with subsequent steps in the program.

Syntax: *CONST, variable[, variable, ...]*

See *CONST* on page 186 for a more detailed description.

FLUSH

When a FLUSH statement runs, Control Basic immediately reads from or writes to the property bound to the local variable declared by ALIAS.

Syntax: *Flush (LocalAlias1)*

See the topics *FLUSH* on page 194 and *ALIAS* on page 182 for more detailed descriptions.

HALT

Stops the program from running and sets the ***Program State*** property in the program object to ***Halted***. The string ***Message*** is displayed in the property ***Description of Halt***.

Syntax: *HALT "Message"*

See the keyword *HALT* on page 198 for a more detailed description.

ISNAN

ISNAN tests the value of ***expression*** to determine if it is a valid number. If the value of ***expression*** is equal to ***NAN*** (Not A Number), then ISNAN returns ***true***. A typical use of ISNAN is to test the present value property of an object in a remote device.

Syntax: *ISNAN(_expression_)*

See *ISNAN* on page 203 for a more detailed description.

LOCALS

Use to declare local variables or arrays of data type REAL. A local variable may be used only within the program in which it is declared.

Syntax: *LOCALS variable[, variable, ...]*

Syntax: *LOCALS arrayName[size]*

See the keyword *LOCALS* on page 204 for a more detailed description.

NAN

Use NAN to set a variable or property to a ***Not A Number*** constant or to test if the variable or property is equal to ***Not A Number***. NAN can be used in both Standard and Next Generation Control Basic.

See *NAN* on page 207 for a more detailed description.

Generation 5

The following keywords are now recognized by Generation 5 Control Basic.

- COV
- ENDFUNC
- ENDPROC

- FRETURN
- FUNCTION
- INTEGER
- NETPOINT
- NETPOINTCACHE
- PRETURN
- PROCEDURE
- REAL
- STRING
- TOSTRING

COV

This command sets up an object and property for a COV (Change of Value) subscription. The object must be in a device that supports COV subscriptions. It applies to all read accesses in a program regardless of location.

Syntax: *COV(device.object.property, increment , confirmed, lifetime)*

See the keyword *COV* on page 188 for details and examples.

ENDFUNC

Marks the end of a user defined Control Basic function.

See the keyword *ENDFUNC* on page 192 and for details and the topic *User defined procedures and functions* on page 176 for an overview of functions.

ENDPROC

Marks the end of a user defined Control Basic procedure.

See the keyword *ENDPROC* on page 192 for details and the topic *User defined procedures and functions* on page 176 for an overview of procedures.

FRETURN

Use FRETURN to designate the value that is returned by the function. A function can include multiple FRETURN statements but only the first one executed designates the returned a value.

See the keyword *FRETURN* on page 195 and for details and the topic *User defined procedures and functions* on page 176 for an overview of functions.

FUNCTION

This keyword marks the beginning of a user defined function. A function is a self-contained block of statements that return a single value.

See the keyword *FUNCTION* on page 195for an example and the topic *User defined procedures and functions* on page 176 for an overview of functions.

INTEGER

Use the command INTEGER to declare a local variable or array of data type *INTEGER*.

Syntax: *INTEGER variable[, variable, ...]*

Syntax: *INTEGER arrayName[size]*

See the keyword *INTEGER* on page 201 for details and an example. For a description of data types, see the topic, *Data types* on page 179.

NETPOINT

Use the NETPOINT function to set the interval for reading from and writing to BACnet off-panel objects.

Syntax: *NETPOINT(device.object.property, read interval, write interval)*

See the keyword *NETPOINT* on page 208 for details and an example.

NETPOINTCACHE

Use NETPOINTCACHE to enable and disable the NETPOINT and COV functions.

Syntax: *NETPOINTCACHE(expression)*

See the keyword *NETPOINTCACHE* on page 209 for details and an example.

PRETURN

Use PRETURN to return from a procedure before reaching ENDPROC. PRETURN is optional.

See the keywords *PRETURN* on page 215 and *PROCEDURE* on page 215 for details and examples.

PROCEDURE

This keyword marks the beginning of a user defined procedure. A procedure is a self-contained block of Control Basic statements that perform a task.

Syntax: *PROCEDURE name([argument, argument, argument...])*

See the keyword *PROCEDURE* on page 215 for an example and the topic *User defined procedures and functions* on page 176 for an overview of procedures.

REAL

Use the command REAL to declare a local variable or array of data type *REAL*. For a description of data types, see the topic, *Data types* on page 179.

Syntax: *REAL variable[, variable, ...]*

See the keyword *REAL* on page 217 for details and an example.

STRING

Use the command STRING to declare a local variable or array of data type *STRING*. For a description of data types, see the topic, *Data types* on page 179.

Syntax: *STRING variable[, variable, ...]*

See the keyword *STRING* on page 226 for details and an example.

TOSTRING

Use TOSTRING to convert a numerical value to a string.

Syntax: *TOSTRING(argument1, argument2)*

See the keyword *TOSTRING* on page 231 for details and an example.

Line numbers and labels

When using Next Generation or Generation 5 Control Basic, the line numbers found in Standard Control Basic are not used. For branching commands that require a destination, such as GOSUB or GOTO, a label is used instead.

Line numbers

Line numbers are not used in Next Generation or Generation 5 Control Basic programs. However, a line number is displayed in the Control Basic editors for BACstage, TotalControl Design Studio, and KMC Connect. The line numbers displayed are only for identifying problems when a program is compiled.

Line numbers continue to be used in controllers with Standard Control Basic.

- In BACstage, enter line numbers as the programs are written.
- In TotalControl Design Studio and KMC Connect, the line numbers are generated automatically.

Labels

In both Next Generation and Generation 5 Control Basic, labels are used instead of line numbers when program flow is redirected with any of the following statements.

- GOSUB
- GOTO
- ONERROR
- ON GOSUB
- ON GOTO

In the following program example, ***CoolMode*** and ***HeatMode*** are examples of destinations for a program redirection.

```
    IF T > 55 THEN GOTO CoolMode
    IF T <= 55 THEN GOTO HeatMode
    END
  CoolMode:
    REM Cooling sequence runs here
    END
  HeatMode:
```

```
REM Heating sequence runs here
END
```

Declare labels by typing a name followed immediately by a colon (:).

- A label can be any combination of letters (A-Z or a-z), numbers (0-9) or the underscore (_).
- Labels are not case sensitive.
- Labels are unique to the program in which they are declared.
- Labels cannot be a Control Basic keyword.

Local variables

The single-letter local variables a-z and A-Z may still be used without program modification. In addition to single letters, more descriptive variables may be used by declaring a variables with the following statements.

- *INTEGER* on page 201
- *LOCALS* on page 204
- *REAL* on page 217
- *STRING* on page 226

References to objects in remote devices

When referring to an object in a remote device, the device name or instance is now separated from the object reference by a period(.). In previous versions, the instance and name were separated with a dash (-).

```
A = 1214.AI1
```

In BACstage the name of the device or object can be used in place of an instance number.

```
A = MechanicalRoom.TempMechRoom
A = 1214.TempMechRoom
A = MechanicalRoom.AI1
```

Place a WAIT statement after the expression that reads the point. Place both the WAIT statement and the expression for the remote point at the end of the program.

```
REM * POINTS TRANSFERRED FROM DEVICE 1213
REM * OUTSIDE AIR TEMPERATURE *
AV24 = 1213.AI7
WAIT 0:05:00
END
```

Appendix C: Conquest compliant controllers

The following KMC Connect applications support controllers listed in this section.

- *Application selection, configuration, and deployment* on page 425
- *Audit application* on page 429
- *VAV Balancing application* on page 433

BAC-4001
BAC-4002
BAC-4003
BAC-4007
BAC-4008

BAC-4021
BAC-4022
BAC-4023
BAC-4027
BAC-4028

BAC-4201
BAC-4202
BAC-4203
BAC-4207
BAC-4208

BAC-4221
BAC-4222
BAC-4223
BAC-4227
BAC-4228

BAC-8001
BAC-8005
BAC-8007
BAC-8205

BAC-5901

BAC-9001
BAC-9021
BAC-9301
BAC-9311

This appendix lists controllers that are compliant and compatible with applications in KMC Connect.

Appendix D: Glossary

This glossary lists the more common terms you may encounter when designing and installing a KMC Connect system.

alarms
Audible or visual messages indicating a value is out of range or an abnormal condition is present. KMC digital controllers generate alarm messages.

analog
Analog describes any fluctuating, evolving, or continually changing process. Examples of analog units are temperatures, setpoints, percent, volts and amperes.

APDU
Application Layer Protocol Data Unit. An APDU is the significant data in a network packet.

ASHRAE
The American Society of Heating, Refrigerating and Air-Conditioning Engineers. Founded in 1894, it is an international organization of 55,000 persons with the mission of advancing heating, ventilation, air conditioning and refrigeration.

ASP.NET
Software that provides the services necessary for developers to build web applications.

BACnet
Building Automation Control Network. A data communications protocol for building automation systems. Developed and maintained by ASHRAE, it is an American National Standards Institute standard (ASHRAE/ANSI 135-1995). BACnet defines how information is packaged for transportation between building automation system (BAS) vendors.

BACnet broadcast
A message that is intended to be received by a group of devices on an internetwork. There are three types of BACnet broadcasts:

- Global Broadcast - All devices in the internetwork get the message.
- Remote Broadcast - All the devices in a remote network get the message.
- Local Broadcast - All the device in the local network get the message.

BACnet broadcast management device (BBMD)
A special type of routing device that is used in BACnet/IP networks to distribute broadcast messages across multiple IP subnetworks.

baud
Pronounced ***bawd***, it is a common reference to the speed at which a modem or other serial device can transmit data. In KMC networks the speed at which a Tier 2 or BACnet MS/TP

network operates is referenced in baud.

The term is named after J.M.E. Baudot, the inventor of the Baudot telegraph code.

BACnet device
Any device, real or virtual, that supports digital communication using the BACnet protocol. Examples of devices are operator terminals, routers, unitary controllers, etc.

client
In networked systems, an application or device acting as a requestor or consumer of data. A client requests a *server* device for data resident in the server.

conformance class
Conformance classes describe the capabilities of a BACnet device for communicating data and interoperating with other BACnet devices. A device's protocol implementation conformance statement (PICS) details its conformance class.

Control Basic
A program embedded in KMC controllers that interprets a set of instructions. Control Basic programs are either written by the installer or supplied the manufacturer with the controller.

controller
A Direct Digital Controller (DDC).

database
A collection of information organized into related tables of data and definitions of data objects. The data within a database can be accessed and manipulated with a computer program. KMC Connect uses SQL (Structured Query Language) as the language for its database.

DCOM
Distributed Component Object Model (DCOM) is a set of Microsoft program interfaces in which client program objects can request services from server program objects on other computers in a network.

device instance
A number that uniquely identifies the device on the internetwork. The device instance number is determined by the BACnet system designer. Valid instance number's range from 0 to 4,194,303 and are assigned to the controller during configuration. It is by reference to the device instance number that data is exchanged between BACnet devices.

digital
Any electronic technology that generates, stores, and processes data in terms of two states: positive and non-positive. Positive is expressed or represented by the number 1 and non-positive by the number 0. Valid units for digital points in KMC controllers include Off/On, Closed/Open, 1 / 0, and Unoccupied/Occupied.

direct digital control
A microprocessor based device or network of devices to control a system or process. The DDC device operates digitally and converts digital information into analog signals which

control HVAC equipment such as hot water valves, heat pumps or air control devices.

directly connected network

A BACnet network that is accessible from a router without messages being relayed through an intervening router. A PTP connection is to a directly connected network if the PTP connection is currently active and no intervening router is used.

end of line

A set of switches or jumpers that indicates the controller is the last physical panel at the end of a network cable. This is not the same as the software ***Last Panel*** setting for KMD series controllers.

enthalpy

Enthalpy is a measure of the heat content within a given sample of air and is expressed in BTUs per pound or as joules per kilogram of dry air. It is used to determine the amount of outside air to add for best economy.

Ethernet

Ethernet is a widely-installed *local area network* (LAN) technology specified by the IEEE standard, IEEE 802.3. Original versions of the Ethernet LAN used coaxial cables and were referred to as thicknet or thinnet. Newer versions (10baseT and 100baseT) connect with unshielded twisted pairs of wires in a cable. Ethernet operates also on fiber optics and as a wireless LAN.

firewall

A security mechanism, or combination of mechanisms, designed to prevent unauthorized or unwanted communications between sections of a computer network. Firewalls are usually both software and hardware based.

flash memory

A special type of EEPROM (electrically erasable programmable read-only memory) that can be erased and reprogrammed in blocks instead of one byte at a time. Flash memory gets its name because the microchip is organized so that a section of memory cells are erased in a single action or “flash.” Flash memory is a non-volatile memory device that retains its data after the power is removed.

frame

See *packet* on page 477.

frameset

A set of image files each of which represents an increment of motion or position.

- When the image files are displayed in rapid succession, the frameset displays an object in motion.
- When the image files are displayed in rapid succession, the frameset displays an object in motion.

gateway

A device that connects two or more different communication protocols so that information can be passed from devices on one network to the other. Gateways are similar to human language translators. A BACnet gateway uses BACnet as a common language on one side and some non-BACnet (usually proprietary) communication scheme on the other side.

Graphics Designer

A comprehensive graphics package within KMC Connect. Within the Graphics Designer system designers can create logical views of the control system on graphics pages. Graphics pages typically include floor plans, air handlers, chiller plants, boiler, central plants, and utility layouts.

half-router

In BACnet, a device that can participate as one partner in a point-to-point (PTP) connection. Two half-routers form an active PTP connection and act as a single router. See the topic *point-to-point* on page 477 for details about using half-routers to link BACnet networks.

hub

A common connection point for nodes on a network. Hubs connect segments of a LAN and contain multiple ports. When a packet (message) arrives at one port, it is copied to all other ports.

Internet Information Services (IIS)

Microsoft Internet Information Services (IIS) are a set of Internet-based services for servers using Microsoft Windows. It is an integrated set of services that enable the deployment of high-performance web sites, web applications, and web services. IIS was initially released as an additional set of Internet based services for Windows NT 3.51. Today, it is shipped with both Windows Server 2003 and Windows XP Professional.

I-Am service

The I-Am service is used to respond to Who-Is service requests. However, the I-Am service request may be issued at any time. It does not need to be preceded by the receipt of a Who-Is service request. A device may be programmed to broadcast an I-Am service request when it powers up. The network address is derived either from the MAC address associated with the I-Am service request, if the device issuing the request is on the local network, or from the NPCI if the device is on a remote network. See *Who-Is service* on page 481.

instance

See *device instance* on page 472.

internetwork

A BACnet internetwork can be as simple as a single network but is usually two or more BACnet networks connected by routers. The BACnet protocol permits up to 65,534 interconnected networks in an internetwork. Internetworks may contain similar or dissimilar physical types such an Ethernet LAN and several MS/TP LANs or several MS/TP LANs. A BACnet internetwork permits only one message path between any two devices.

IP address

Short for Internet Protocol address, it is the address of a computer or other network device on a network using the IP protocol. The number *10.1.1.2* is an example of a typical IP address. The IP address is usually assigned by the network administrator.

KMD

An abbreviation for KMDigital, a KMC proprietary line of DDC controllers. The controllers are either Tier 1 or Tier 2 controllers. Tier 1 controllers connect directly to a LAN through Ethernet; Tier 2 controllers connect to each other and Tier 1 controllers on a EIA-485 network. KMC Connect connects only to Tier 1 controllers; Tier 2 controllers are accessible to KMC Connect through the Tier 1 controllers.

last panel

The highest numbered panel on the network. A controller assigned as the last panel will pass the token back to the lowest numbered panel when it is done talking on the network. Last Panel is not the same as end-of-line termination.

local network

The network to which a BACnet device is directly connected.

local area network

A collection of interconnected equipment that can share data, applications, and resources. It may include computers, printers, data storage devices and industrial controllers and machines. A LAN device can send and receive signals from all other devices in the network. Networks use protocols, or rules, to exchange information through a single shared connection. These protocols prevent collisions of data caused by simultaneous transmission between two or more computers.

The physical connection between LAN devices can be a coaxial cable, pairs of copper wires, or optical fiber. Wireless connections also can be made using infrared or radio-frequency transmissions.

MAC address

The MAC address uniquely identifies a device on its network. Each network type–Ethernet 8802-3, IP or MS/TP–has its own MAC addressing scheme.

main Panel

See *Tier 1 controller* on page 480.

master and slave devices

MS/TP devices come in two varieties:

- Slave devices are suited for the lowest-cost implementations but they lack the capability to initiate requests; they can only reply to messages from other devices.
- Master devices are able to initiate requests, but they must also be able to negotiate for a time slot in which to make their requests. This adds some processing and memory requirements to the Master device which can result in higher cost than the slave.

Microsoft.NET Framework

The Microsoft .NET Framework is a component of the Microsoft Windows operating system. It provides a large body of pre-coded solutions to common program requirements, and manages the execution of programs written specifically for the framework. The framework is intended to make it easier to develop computer applications and to reduce the vulnerability of applications and computers to security threats.

Modbus

A protocol developed to establish master-slave/client-server communication between industrial devices. It is an open and widely used network protocol in the industrial manufacturing environment. Modbus devices communicate over a serial network in a master/slave (request/response) type relationship using one of two transmission modes: ASCII (American Standard Code for Information Interchange) mode or RTU (Remote Terminal Unit) mode.

MS/TP

The MS/TP (master slave/token passing) protocol is unique to BACnet and is implemented using the EIA–485 signaling standard. This is a shielded twisted-pair LAN operating at speeds from 9600 baud to 76,800 baud.

Network Manager

Network Manager is the user interface for navigating through the control system. It displays a hierarchical view of the physical layout of the control system including networks, devices, controllers, objects, and properties.

network segment

A logical or physical subdivision of network.

Ethernet segments In Ethernet, bridges, hubs, switches, and repeaters can couple multiple physical network segments into one logical network segment.

BACnet network segments An electrically separate section of a network.

node

A device—such as a computer or a controller—on a network that is capable of communicating with other network devices.

network numbers

A number from 1 to 65,534 that identifies specific BACnet network. It is assigned by the BACnet system designer at the time a router is initialized for network operation.

node

A device such as a computer or a controller on a network that is capable of communicating with other network devices.

object

Objects are the means by which a BACnet device represents information that can be observed or changed. The object may represent a physical point such as an input or output or a logical grouping of data such as a PID loop, schedule or variable. Objects have a set of properties and a group of functions that can be applied to them.

The BACnet standard defines a standard set of objects, these include analog and binary inputs, outputs, and values; control loops; and schedules.

OPC

OPC is the original name for ***OLE for Process Control***, an open standards specification developed in 1996 by an industrial automation industry task force. The standard specifies the communication of real-time plant data among control devices from different manufacturers. OPC servers provide a method for many different software packages to access data from a process control device, such as a PLC or DCS.

packet

A packet (or frame) is piece of a message transmitted over a packet-switching network. One of the key features of a packet is that it contains the destination address in addition to the data.

PAD router

A BACnet IP PAD is a special type of router that connects two or more BACnet network segments that are separated by at least one IP-only router. The PAD router monitors network traffic for BACnet messages addressed to the other subnet and repackages the message that can pass through IP routers, in effect forming a “tunnel” between the two network segments. A companion PAD router unpacks and retransmits the message on the remote BACnet network.

peer-to-peer communications

The controllers in a network hierarchy which are considered to be at the same level. There are no host or master controllers. Each controller is at the same level as its peers.

PICS

See *Protocol Implementation Conformance Statement* on page 478

PID controller

A Proportional Integral Derivative loop is an algorithm built into each KMC controller that calculates a value between 0 and 100 percent. The output of the loop can then be used to control the position of an actuator. The output value is based on the sensed value and the required setpoint.

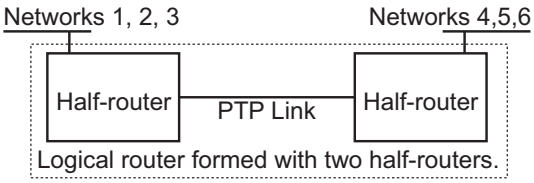
point

Any hardware or software object configured in a KMC KMD series digital controller. A point can be an input, output, variable, schedule, log or a PID controller loop.

point-to-point

In BACnet, it is a method of data transmission to provide serial communications between two BACnet devices. Typically used for remote and roaming access to BACnet systems, dial-up communications over modems or a portable computer connection to a controller, for example. PTP is based on the EIA-232 direct point-to-point connection or via dial-up telephone modems. The point-to-point link in *BACnet half-routers* on page 478 connects networks 1, 2 and 3 to networks 4,5 and 6 as if they were on the same LAN.

Illustration D–1 BACnet half-routers



BACnet PTP provides for internetworked communications over modems and voice grade phone lines. PTP accommodates modem protocols V.32bis and V.42 and also supports direct cable connections using the EIA-232 signaling standard.

port

An interface on a computer, either physical or logical, to which you can connect a device. Examples of physical ports are connections for disk drives, display screens, keyboards, networks, etc. Ports may also be logical connection on networks. For example, port 80 is used for HTTP traffic.

priority array

BACnet devices use the priority array property to control ***Present Value*** in certain objects. For KMC Controls BACnet devices this property is part of both analog and binary output and value objects.

properties

Properties are used to read information from objects or write information to objects. Each BACnet object is provided with a standard set of properties that describes the object and its current status. Certain properties of an object may be required, while others may be optional.

protocol

A definition or rules of communication for a computer network. A formal set of conventions governing the format and relative timing of message exchange between two communications terminals.

Protocol Implementation Conformance Statement

A Protocol Implementation Conformance Statement (PICS) is a document that specifies exactly which portions of the BACnet Standard a device actually implements. Manufacturer's issue a PIC statement for each BACnet device.

recipient

An individual operator that is designated to receive alarms from a KMC Connect system.

remote network

Pertaining to devices or nodes on a network different than the referenced device.

repeater

A network device used to regenerate analog or digital signals distorted by transmission loss. A repeater cannot do the intelligent routing performed by bridges and routers.

routers, BACnet
BACnet routers connect different types of BACnet networks. The KMC Controls BAC–5050 BACnet router links BACnet 8802-3, BACnet IP and MS/TP networks. It can also be configured as a *PAD router*.

server
In networked computer systems, an application or device acting as a provider of data, responding to a request from a *client*. The term server can refer to hardware optimized to perform server functions or software such as an SQL server.

services
BACnet services control the transfer of information between BACnet devices. Examples of services include scheduled commands and alarms between BACnet devices. BACnet defines 26 standard services. Some services read or write properties of objects in the receiving device. Others convey notification of alarms or other special events, others read and write files, and so on. The services provided by a BACnet device are generally described by the device's PIC statement.

setpoint
The desired temperature of a space.

scan
The process Control Basic reads the state of points or properties. The period of time required for the processor to perform all of its instructions and programs.

Site Explorer
Site Explorer is the user interface to manage graphic pages and connections to a KMC Connect managed building automation systems.

slave device
See *master and slave devices* on page 475.

SQL
Abbreviation for structured query language and pronounced either see-kwell or as separate letters. SQL is an industry standard query language for requesting and manipulating information from a database.

subnet
A subdivision of an IP network. Each subnet has its own unique network ID.

subnet panel
See *Tier 2 controller* on page 480.

subnet mask
Short for subnetwork mask, a subnet mask is method of dividing a network of IP addresses into groups. It enables the recipient of IP packets to distinguish the network ID and host ID portions of the IP address. A common example of a subnet mask used is 255.255.255.0. Subnet masks are assigned by the network administrator.

switch
A special type of hub that forwards packets to the appropriate port based on the packet's address. Switching hubs improve performance over conventional hubs. A switch may also be referred to as a switching hub.

system
All of the components that combine to control a building. This may include controllers, repeaters, modems, and computers.

system monitor engine
One of the KMC Connect building services. The system monitor engine (SME) performs data management among the other components of TotalControl.

Tier 1 controller
An upper level controller in the KMC digital network hierarchy. The Tier 1 controller can have one or more Tier 2 networks connected to it. A Tier 1 controller may also have ports for connection to computers, a network, buffered modem communications, a dedicated system printer and OEM equipment interfacing.

The Tier 1 controller has many other advantages for large networks such as large areas of memory for system groups, Control Basic programs, schedules, and passwords. Tier 1 controllers have additional commands and statements available to allow complex programs to be easily written.

Also referred to as a LAN controller, MultiNet® or main panel.

Tier 2 controller
A KMC controller with built-in peer-to-peer, EIA–485 network communications. Also referred to as a Subnet controller or sub-panel.

TCP/IP
An abbreviation for ***Transmission Control Protocol*** and ***Internet Protocol***. TCP/IP is two separate protocols that are used together.

- The Internet Protocol standard defines how packets of information are sent out over networks. IP has a packet-addressing method that lets any computer on the Internet forward a packet to another computer that is a step (or more) closer to the packet's recipient.
- The Transmission Control Protocol ensures the reliability of data transmission across Internet connected networks. TCP checks packets for errors and submits request for re-transmissions if errors are found; it also will return the multiple packets of a message into a proper, original sequence when the message reaches its destination.

token
A special network message that circulates around a token ring network. Only the device that has the token can transmit data on the token ring network. BACnet MS/TP and the KMD protocal are both token passing protocols.

trend logs

Trend logs are a record of the value of a point or property as measured by a building automation device or controller. KMC Connect supports three types of trend logs.

- Controller trend logs are sets of historical data collected and stored only within the memory of an individual controller.
- Database trend logs are controller trend logs that KMC Connect has retrieved from a controller and then stored in the site database.
- A PC trend log is a set of data resulting from regularly sampling the value of a property or point and then storing the results in the site database.

tunnel router

See *PAD router* on page 477.

UDP/IP

An abbreviation for ***User Datagram Protocol*** and ***Internet Protocol***, a connectionless protocol that, like TCP, runs on top of IP networks. Unlike TCP/IP, UDP/IP provides very few error recovery services, offering instead a direct way to send and receive datagrams over an IP network. It's used primarily for broadcasting messages over a network.

variables

Virtual points in the KMD protocal that can represent temperatures, setpoints, offsets, multipliers, or digital values.

universal serial bus

An external bus standard that supports data transfer rates of 12 Mbps. A single USB port can be used to connect up to 127 peripheral devices such as mice, modems and keyboards.

want-point

Points that are being requested from the network. A list of want-points is created in each panel when a point in another panel is referenced in a Control Basic program, system group, or trend log.

Who-Is service

A BACnet device sends The Who-Is service message to determine the device object identifier and network addresses of all devices on the network, or to determine the network address of a specific device whose device object identifier is known, but whose address is not. See *I-Am service* on page 474.

wide area network (WAN)

A geographically widespread network, usually comprising one or more LANs. WANs are often connected through public networks, leased lines, and satellite or microwave communications.

XML

The Extensible Markup Language (XML) is a standard for marking up documents and data. Its primary purpose is to facilitate the sharing of data among diverse systems and programs that are often operating on different computer platforms. Its flexible syntax allows you to use

it to describe virtually any kind of information from a simple recipe to a complex database. An XML document effectively structures and labels the information it contains so a computer program can extract, sort, filter, and arrange the information in highly flexible ways.

Index

A

about
alarms 103
BACnet algorithmic 113
BACnet intrinsic 108
BACnet services 291
change of value notifications 119
Code Editor 151
Control Basic 165
KMC Connect 29
Network Manager 37
objects 291
Output Window 43
properties 291
Resource Manager 45
schedules 145
table view 55
ABS 181
abs block 247
ACC 175
access door object 316
accumulator object 336
action in PID loops
BACnet loop object 361
add
inputs and outputs to expansion panels 302
objects to device 302
user to site
in Security Manager 71
users to a group
in Security Manager 73
add block 247
address
IP 313
AI 175
alarms 471
Alarm Monitor bar 131
audible 34
BACnet
algorithmic 113
intrinsic 108
configuration wizard 106
configuring BACnet 105
disable by network 134
email server 141
filtering 133
filtering by user group 139
GEST 323
muting 34
operator notifications 136
overview 103
printer 141
summary in BACnet device 130
wizards 106
algorithmic alarms
about 113
configuring 114
ALIAS 182
analog 471
BACnet input objects 340
BACnet output object 366
BACnet value objects 392
AND 184
and block 248
annual schedules
See BACnet calendar object 321
antiCycle block 248
AO 175
APDU 471
applicable controllers
block programming 233
application
audit 429
commanding VAV airflow 437
configuration 425
custom application wizard 439
deployment 425
selection 425
VAV balancing 433
VAV flow monitoring 436
ARCCOS 184
arccos block 249
arccosine for KMD contollers 187
archiving
trend logs 100
ARCSIN 184
arcsin block 249
arcsine for KMD controllers 224
ARCTAN 184
arctan block 250
arctangent for KMD controllers 227
arithmetic operators 172
arrays
declaring with
INTEGER 201
LOCALS 204
REAL 217
STRING 226

programming with 178
ASHRAE 471
ASP.NET 471
audit
application 429
log 75
Auto-Hide 32
automatic
blanking NetSensor for BACnet 404
MAC addressing 308
automatic log off 34
AV 175
average
BACnet inputs 340
averaging objects 320
AVG 184
avg block 250

B

backup
BACnet backup 309
devices from Network Manager 309
entire site 63
files for devices and controllers 51
with .bnd file 309
BACnet 471
about services 291
backup device 309
broadcast 471
broadcast management device 451, 471
controllers 419
device or controller 472
Ethernet 456
foreign device 451
half-router 474
IP 451
network number 451
objects 315
overview 291
restore device 309
BACstac driver
configuring 449
BACstage
importing and exporting files 310
balancing application 433
baud 471
in devices 323
BBMD configuration 313
BBMD, see BACnet broadcast management device 471
BI 175
binary
BACnet input objects 345
BACnet output objects 370
BACnet value objects 396
BIND 185
block programming 233
applicable controllers 233
importing 245
limitations 245
local variables 239
macros 241
reference to blocks 247
routines 240
staring the editor 234
subroutines 240
unsupported keywords 245
blocks
abs 247
add 247
and 248
antiCycle 248
arccos 249
arcsin 249
arctan 250
avg 250
call 240
clear 250
close 251
compare 251
compareEqual 252
compareGreater 252
compareGreaterEqual 253
compareLess 253
compareLessEqual 254
compareNotEqual 254
cos 255
counter 256
dec 257
delayOnBreak 257
delayOnMake 256
dewpoint 257
dewpointsi 258
disable 258
divide 259
dom 260-261
enable 261
end 262
enthalpy 262
enthalpysi 262
equalToDiff 263
for 263

Get 238
greaterThanDiff 264
if 265
ifMinus 265
ifPlus 266
inc 266
int 267
integerDivide 267
interval 267
invln 268
lessThanDiff 268
limiter 269
ln 269
lsel 264, 269
macro 241
max 270
min 270
minMaxAvg 271
mod 271
modelnumber 273
month 272
multiply 272
nan 273
netsensorstatus 272
oneShot 274
open 274
or 275
outputoverride 275
paneladdress 275
pi 276
power 276
powerloss 276
program start 277
rem 277
reset 277
rlq 278
rnd 278
scans 279
schedoff 279
schedon 280
select 280
sensoroff 281
sensoron 282
Set 238
sin 283
slope 283
sqr 284
start 284
subtract 285
switch 285
tan 286
tbl 286
time 286
xor 287
year 288
bnd backup files 51
BO 175
bookmarks 158
boolean logic 172
buttons
NetSensor-BAC controllers 404
BV 175

C

cached values
COV client 120
calendar object 321
Calendar object 148
calibration
BACnet analog input objects 340
BACnet custom device types 340
NetSensor-BAC controllers 404
call block 240
Cbasic table 414
change of value 119
about 119
COV client 119
COV server 119
subscribe configuration pages to object 121
subscribe configuration pages to property 122
subscribe to object 119
subscribe to property 119
subscribe to trend logs 124
subscription list 119
unsubscribing 123
CLEAR 185
clear block 250
clearing
Control Basic programs 152
Output Window messages 44
client 472
CLOSE 186
close block 251
Code Editor
about 151
changing text font and color 162
options 162
programming 151
select all 157
cold start 306
color
in Code Editor 162
command object 322

compareEqual block 252
compareGreater block 252
compareGreaterEqual block 253
compareLess block 253
compareLessEqual block 254
compareNotEqual block 254
comparison operators 171
compile
  Control Basic programs 152
  options 152
computer
  names 18
  requirements and specifications 16
configuration application 425
configuration files, scanning 308
configuring
  alarms
    in BACnet devices 105
    operator notifications 136
  BACnet objects 300
  BACnet PDS 311
  BACstac for Ethernet 456
  BACstac for IP, BBMD and foreign device 451
  building services 40
  Cimetrics BACnet driver 449
  COV notifications
    for configuration pages 121-122
    for trend logs 124
  Design Studio workspace 30
  devices 300
  NetSensors 403
  notification service 141
  trend logs
    BACnet controller 92
    BACnet device 94
    BACnet PC 96
conformance class 472
CONST 186
constants in Control Basic 186
Control Basic 165, 169, 204, 472
  BACnet tables 414
  editing 152
  expressions 166
  find and replace 158
  functions 166
  Generation 5 459
  keywords 181
  labels 167
  line numbers 167
  mnemonics 174
  Next Generation 459
  scans 165
  statements 166
  undo 157
  variables 173
  writing offline 154
Control Basic,version in controllers 460
Control Basic;procedures and functions 176
controller 472
controllers 469
converting input and output objects 303
copy
  Code Editor 157
  device configuration 300
COS 187
COS-1 187
cos block 255
counter block 256
COV 188
COV See change of value 119
create
  BACnet objects 302
custom
  BACnet device type 340
custom application wizard 439
custom table view 59
cut in Code Editor 157

D

database 472
day
  of month 190
  of week 191
  of year 191
daylight saving time
  in BACnet devices 323
DCOM 472
DDC, See direct digital control 472
DEC 189
dec block 257
declaring
  arrays
    INTEGER 201
    LOCALS 204
    REAL 217
    STRING 226
  constants with CONST 186
  variables
    INTEGER 201
    LOCALS 204
    REAL 217
    STRING 226

default values
PC trend log interval 41
delayOnBreak block 257
delayOnMake block 256
delete
inputs and outputs from expansion panels 302
objects from device 302
user from Site in Security Manager 72
deployment application 425
deprecated keywords 461
derivative in PID loops
BACnet controllers 363
device
BACnet object 323
instance 323
device instance 472
device tables 413
DEW-POINT 189
DEWPOINT 189
dewpoint block 257
DEWPOINTSI 189-190
dewpointsi block 258
digital 472
direct digital control 472
directly connected network 473
DISABLE 190
disable block 258
discovering devices 39
discovering devices, See Find Devices 298
divide block 259
docked panes 30
DOM 190
dom block 260
DOW 191
dow block 260
DOY 191
doy block 261
DST
in BACnet devices 323
dynamic access 178

E

editing Control Basic 152
ELSE 198
email
alarm notification 136
backup files 63
server configuration 141
ENABLE 191
enable block 261
END 192
end block 262
end of line 473
ENDFUNC 192
ENDIF 198
ENDPROC 192
engineering units
BACnet analog 417
BACnet binary 417
enthalpy 473
ENTHALPY 193
ENTHALPY-SI 193
enthalpy block 262
ENTHALPYSI 193
enthalpysi block 262
equalToDiff block 263
Ethernet 473
connection in devices 313
Ethernet in BACstac 456
event
acknowledgment 365
enrollment object 329
enrollment summary 124
processing 365
summary in BACnet device 130
event log objects 333
example programs 181
exception schedules 379
expansion modules
add inputs and outputs 302
expiration of passwords 68
exporting
archived trend logs 100
BACstage .BAC files 310
from Network Manager 309
trend logs
from trend viewer 84
expressions in Control Basic 166

F

file objects 336
filter weight in BACnet inputs 340
filtering
alarms 133
alarms by user group 139
BACnet analog inputs 340
Output Window messages 44
find
devices 298
text in Control Basic 158

firewall 473
firmware updates 77
flash memory 473
FLASH over the network 77
floating panes 32
FLUSH 194
for block 263
for the workspace 34
FOR TO NEXT 194
foreign device configuration 313
frame 473
frameset 473
FRETURN 195
FUNCTION 195
functions and procedures 176
functions in Control Basic 166

G

Gated Event State Transitions 323
gateway 474
Generation 5 Control Basic 459
GEST 323
get block 238
GOSUB 196
GOTO 197
goto in Code Editor 157
Graphics Designer 474
greaterThanDiff block 264
group
  BACnet group object 336
  user groups in Security Manager 73
Groups tab 68

H

half-router 474
halt 152
HALT 198
heartbeat 311
help examples 181
hibernation mode 18
hidden panes 31
hierarchy of operators 170
holiday schedule
  BACnet controllers 321
HSEL 198
hsel block 264
hub 474
humidity programing
  NetSensor-BAC controllers 404

I

I-Am service 474
icons
  BACnet devices 298
  NetSensor 409
  Network Manager 38
  Output Window 43
IF- THEN 200
if block 265
IF THEN (ELSE) 198
IF THEN ELSE 200
IF+ THEN 199
ifMinus block 265
ifPlus block 266
IIS 474
importing
  BACstage .BAC files 310
  in Network Manager 309
importing block programming 245
INC 200
inc block 266
input objects
  accumulator 336
  analog 340
  BACnet pulse 336
  binary 345
  changing type 303
  creating and deleting 302
  multistate 348
input tables 413
installing
  computers 18
  KMC Connect 19
instance 472
INT 202
int block 267
INTEGER 201
integerDivide block 267
integral in PID loops
  BACnet controllers 363
Intellisense filter 156
Internet Information Service 474
internetwork 474
INTERVAL 202
interval block 267
INVLN 202
invln block 268
IP address 313, 475
ISNAN 203

K

keyboard shortcuts
  Code Editor 157
keywords 181
  deprecated 461
  Intellisense filter 162
  prompting 156
KMC Connect 29
KMD
  backup files 51
  protocol 475

L

labels 167
LAN, See Local Area Network 475
last panel 475
leap year 206
lessThanDiff block 268
LET 203
life safety objects
  point 351
  zone 354
limitations of block programming 245
limited discovery 39
limiter block 269
line numbers 167
LN 204
LN-1 204
ln block 269
load control objects 358
Local Area Network 475
local network 475
local variables 173, 239
LOCALS 204
logarithm
  invln (inverse natural) block 268
  ln (natural) block 269
loop
  BACnet PID loop object 361
LOOP 175
lost password 68
LSEL 205
lsel block 269

M

MAC address 475
  in BACnet 308
  in devices 323
macro blocks
  antiCycle 248
  delayOnBreak 257
  delayOnMake 256
  equalToDiff 263
  greaterThanDiff 264
  limiter 269
  minMaxAvg 271
  reset 277
  reusing 241
  saving 241
  select 280
  slope 283
  switch 285
main panel, See Tier 1 controller 475
manual change
  analog value 392
  binary value 396, 399
  output 370
MAX 205
max block 270
messages in Output Window 43
Microsoft.Net 476
MIN 206
min block 270
minMaxAvg block 271
mnemonics 174
MOD 206
mod block 271
Modbus 476
MODELNUMBER 206
modelnumber block 273
MONTH 207
month block 272
motion sensing
  NetSensor-BAC controllers 404
MS/TP 476
MSI 175
MSO 175
MSV 175
multiply block 272
multistate objects
  input 348
  output 373
  value 399
mute alarms 34

N

names
  alarm recipient operators 136
  in Security Manager 68
names for computers 18

NAN 207
nan block 273
NEG_INFINITY 207
NETPOINT 208
NETPOINTCACHE 209
NetSensor
  Conquest 409
  STE-9000 409
NetSensor-BAC controllers
  auto-blanking 404
  buttons 404
  calibration 404
  humidity 404
  motion sensing 404
  programming 404
  setpoint 404
  temperature 404
  time 404
  verifying status 404
NETSENSOR-STATUS 209
NETSENSORSTATUS 209
netsensorstatus block 272
network
  directly connected 473
  FLASHing controllers 77
  network segment 476
  numbers 476
Network Manager 37, 476
  display settings 41
  icons 38
  opening 38
  options 41
  scanning for uninitialized files 308
network number 451
networks 42
new
  security group in Security Manager 73
Next Generation Control Basic 459
node 476
NOT 210
not block 273
notification class object 365
notifications
  BACnet algorithmic 113
  BACnet change of value 119
  BACnet intrinsic 108

O

object 476
objects
  about 291
  access door 316
  accumulator 336
  analog input 340
  analog output 366
  analog value 392
  averaging 320
  binary input 345
  binary output 370
  binary value 396
  calendar 321
  command 322
  creating and deleting 302
  device 323
  event enrollment 329
  event log 333
  file 336
  group 336
  life safety point 351
  life safety zone 354
  load control 358
  loop 361
  multistate input 348
  multistate output 373
  multistate value 399
  notification class 365
  program 377
  schedule 379
  structured view 383
  trend 384
  trend log multiple 388
off-panel points
  reading 173
  writing 173
offline programming 26
ON GOSUB 210
ON GOTO 211
ONERROR 212
oneShot block 274
OPC 477
OPEN 212
open block 274
opening
  Control Basic programs 152
  Network Manager 38
  Output Window 43
  Resource folder 52
  Resource Manager 47
operators
  arithmetic 172
  boolean 172
  comparison 171

precedence 170
relational 171
options 34
alarm popup 34
audible alarm 34
Code Editor 162
Network Manager 41
Output Window 44
OR 213
or block 275
OUTPUT-OVERRIDE 213
output objects
analog 366
binary 370
changing type 303
creating and deleting 302
multistate 373
Output Window 43
icons 43
OUTPUTOVERRIDE 213
outputoverride block 275

P

PAD router 477
PANEL-ADDRESS 214
PANELADDRESS 214
paneladdress block 275
panes
docked 30
floating 32
hidden 31
password 65
changing in Security Manager 72
expiration 68
NetSenSensor 409
to reinitialize BACnet device 306
paste
Code Editor 157
peer-to-peer 477
permissions
changing for a group 73
PI 214
pi block 276
PIC statement 478
PID controller loops 477
PID loops
BACnet PID loop object 361
pinned 32
point 477
point-to-point 477
ports 478
POS-INFINITY 214
POWER-LOSS 215
power block 276
power settings 18
POWERLOSS 215
powerloss block 276
precedence of operators 170
PRETURN 215
priority array 478
BACnet standard levels 292
priority for writing
macro block 241
set block 238
PROCEDURE 215
procedures and functions 176
process identifier 365
program start block 277
programming examples
NetSensor-BAC controllers 404
programming notation 169
properties 478
about 291
proportional in PID loops
BACnet controllers 363
protocol 478
Protocol Implementation Conformance Statement 478
PTP 477
pulse input
BACnet accumulator 336

R

radians 214
random numbers 219
READ-CONBIAS 216
READ-CONPROP 216
READ-CONRATE 216
READ-CONRESET 216
reading properties from other devices 173
REAL 217
redo in Code Editor 157
regenerate networks 304
regular expressions 158
reinitialize device 306
relational operators 171
REM 218
rem block 277
renumber Control Basic lines 157
repeater 478
replace
text in Control Basic 158

requirements
computers 16
reset block 277
Resource folder 52
Resource Manager 45
restart BACnet device 306
restart BACstac driver
from Network Manager 42
restoring
controller configuration 48
devices from Network Manager 309
site from backup file 64
RETURN 218
reusing macro blocks 241
RLQ 218
rlq block 278
RND 219
rnd block 278
router
BACnet 479
half-router in BACnet 474
routine start block 278
routines 240
run
Control Basic programs 152

S

saving
trend logs
from trend viewer 84
saving macro blocks 241
scanning for uninitialized files 308
scans 165
SCANS 219
scans block 279
SCHED 175
SCHED-OFF 220
SCHED-ON 221
SCHEDOFF 220
schedoff block 279
SCHEDON 221
schedon block 280
schedules
about 145
annual, See calendar object 321
BACnet schedule object 379
exception 379
holiday 321
object 379
override 379
SCHED-OFF 220
SCHEDON 221
security 65
adding a new user 71
groups 73
question 68
Security Manager 65
select all 157
select block 280
selection application 425
SENSOR-OFF 223
SENSOR-ON 224
SENSOROFF 222
sensoroff block 281
SENSORON 223
sensoron block 282
server 479
services 479
configuring 40
set block 238
setpoint 479
NetSensor-BAC controllers 404
shedding load with load control object 358
shortcut keys
Code Editor 157
Show By in Output Window 44
SIN 224
SIN-1 224
sin block 283
site
add new 23
back up 63
close 26
management tools 29
offline programming 26
open existing 25
restoring from backup file 64
Site Explorer 479
slave device 479
sleep mode 18
slope block 283
SME, see system monitor engine 480
Sort By in Output Window 44
sorting
Output Window messages 44
Output Window options 44
specifications
computers 16
SQL 479
SQR 225

sqr block 284
START 225
start block 284
startup 22
statements in Control Basic 166
STEP 194
stop
  disable 284
STOP 226
stop block 284
STRING 226
structured view objects 383
sub-net 479
subnet 479
Subnet 479
subnet mask 479
subscribe to object 119
subscribe to property 119
subscription list 119
subtract block 285
summary
  event enrollment 124
switch 480
switch block 285
switches, end of line 473
system monitor engine 480

T

table view
  about 55
  custom views 59
  group of objects 56
tables 413
  BACnet Control Basic 414
  BACnet device 413
  BACnet input 413
  Cbasic 414
  TBL in Control Basic 228
TAN 228
TAN-1 227
tan block 286
TBL 228
tbl block 286
temperature
  NetSensor-BAC controllers 404
termination, end of line switches 473
text
  font and color in Code Editor 162
Tier 1 Controller 480
Tier 2 Controller 480
time
  BACnet synchronization interval 311
  DST in BACnet devices 323
  NetSensor-BAC controllers 404
  setting in BACnet devices 307
  synchronization for BACnet 307
  UTC in BACnet systems 307
TIME 229
TIME-OFF 230
TIME-ON 231
time block 286
TIMEOFF 230
TIMEON 231
timeout 34
TL 175
token 480
toolbar
  Network Manager 38
tools
  site management 29
TOSTRING 231
trend logs 481
  archiving 100
  change of value 96
  configuration wizard 90
  configuring
    BACnet controller 92
    BACnet device trend logs 94
    BACnet PC trend logs 96
  creating with trend log multiple object 388
  creating with trend object 384
  trend log multiple object 388
  viewing
    BACnet controller trends 384
  viewing controller 87
  wizard 90
trend object 384
types 179

U

UDP/IP 481
undo 157
uninstall 20
units
  standard analog units 417
  standard binary units 417
universal serial bus 481
unsubscribing from COV subscriptions 123
updating firmware 77
USB 481
user groups in Security Manager 73

user name 65
User tab in Security Manager 68
UTC (Universal Coordinated Time) 307, 323

V

value objects
  analog 392
  binary 396
  multistate 399
variables 173, 481
  declaring with
    INTEGER 201
    LOCALS 204
    REAL 217
    STRING 226
variables, local 239
VAV
  balancing application 433
  commanding airflow 437
  flow monitoring 436
viewer 84
viewing
  COV subscriptions 124
  event enrollment summary 124
  groups of objects 55
  notification summary in BACnet 130
  trend logs
    controller logs 87

W

WAIT 232
WAN, See wide area network 481
want-point 481
warm start 306
weekly schedules
  in BACnet schedule objects 379
wide area network 481
within block 287
wizards
  alarm configuration 106
  analog input object 441
  Bacnet input objects 441
  binary input object 445
  custom application 439
  trend configuration 90
workspace in Design Studio 30
writing
  programs 166
  properties to other devices 173

X

XML 481
XOR 232
xor block 287

Y

YEAR 232
year block 288

Z

zero-length files 308
zoom in and out
  trend logs 87